# From Discourse Analysis to Groupware Design

A Dissertation

Presented to

The Faculty of the Graduate School of Arts and Sciences

Brandeis University

Michtom School of Computer Science

Richard Alterman, Advisor

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

Alexander C. Feinman

May, 2006

This dissertation, directed and approved by Alexander C. Feinman's committee, has been accepted and approved by the Graduate Faculty of Brandeis University in partial fulfillment of the requirements for the degree of:

**DOCTOR OF PHILOSOPHY**

Adam B. Jaffe, Dean of Arts and Sciences

Dissertation Committee:

Richard Alterman, Chair

Bradley Goodman

David Jacobson

James Pustejovsky

To Jennifer:

May you some day know me as other than a grad student.

# Acknowledgments

My deepest acknowledgments to my advisor Richard Alterman and my fellow graduate students Seth Landsman and Josh Introne for their contributions, past and present. My gratitude also goes to others who gave their direction and feedback in some degree during the course of this thesis: Alex Bradley, Jennifer Fitzsimmons-Gauger, Mike Head, Heather Quinn, Dan Ritter, David Wittenberg, my parents, and many others.

Thanks also to the Fall 2003 CoSci 111 data for their patience and generosity in providing data for my conclusions, and to the many test subjects for letting us record their fortes, foibles, failures, and triumphs.

# Abstract


## From Discourse Analysis to Groupware Design

A dissertation presented to the Faculty of
the Graduate School of Arts and Sciences of
Brandeis University, Waltham, Massachusetts

by Alexander C. Feinman

Online collaboration has become ubiquitous. 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. Analysis methods that help system designers understand the complex interaction of online participants in a joint activity are crucial for designing effective software tools. This thesis describes how the referential structure of participant discourse can be used to model the interaction and recommend new system designs.

The thesis details a methodology for systematically analyzing online interaction, centered around applying ethnographically-informed analysis techniques to the discourse generated by participants. The methods we propose use references in the discourse as a way to infer the flow of information between representations. Experimental evidence will be presented showing that the methods produce models that reflect the reality of an interaction and predict the effect that changes will have on an interaction. Evidence will also be presented demonstrating the ability of the methodology to be taught and to be applied to a wide variety of domains. By matching the types of information, and the characteristic patterns of information access within the system, to specific representation properties, it is possible to recommend new representations which will improve task performance and reduce user frustration and error rate.

# Contents

**Abstract** **vi**

**1 Introduction** **1**
1.1 Overview . . . 1
1.1.1 Motivation: designing groupware . . . 2
1.1.2 Design methodology . . . 3
1.2 Theoretical background . . . 6
1.3 Referential structure analysis . . . 9
1.3.1 Analyzing the dialogue . . . 11
1.3.2 Using the methods to inform redesign . . . 14
1.4 Overview . . . 14

**2 Related Work** **17**
2.1 Designing groupware . . . 17
2.1.1 Example: the burger joint . . . 18
2.2 The design process . . . 19
2.2.1 Prescriptive rulesets . . . 22
2.2.2 Design tradeoffs . . . 23
2.3 Modeling interaction . . . 24
2.3.1 Distributed Cognition . . . 25
2.3.2 Task Analysis . . . 28
2.3.3 Workflow Analysis . . . 32
2.3.4 Activity Theory . . . 33
2.4 Analyzing ethnographic data . . . 35
2.4.1 Discourse Analysis . . . 36
2.4.2 Conversation Analysis . . . 39
2.5 Predicting system impact . . . 41
2.5.1 Impact on interpersonal relations . . . 43
2.6 Conclusions . . . 45

**3 Redesigning Groupware** **47**
3.1 Chronology . . . 47
3.2 The VesselWorld testbed . . . 49

3.2.1 Collecting usage data in VesselWorld . . . . . 51
3.3 Recurrence analysis . . . . . 53
3.3.1 Recurrent patterns of coordination . . . . . 55
3.3.2 Recurrent errors in coordination . . . . . 56
3.3.3 Creation of secondary structure . . . . . 58
3.4 Redesigning VesselWorld: VW2 . . . . . 62
3.4.1 Shared Planning . . . . . 63
3.4.2 The Object List . . . . . 64
3.4.3 Strategy . . . . . 64
3.5 Testing the redesign: VW3 . . . . . 66
3.5.1 Anticipated results . . . . . 67
3.5.2 Unexpected results . . . . . 69
3.5.3 Summary . . . . . 70

**4 Referential Structure Analysis 71**
4.1 Method . . . . . 71
4.1.1 Extracting referential structure . . . . . 71
4.1.2 Referent types and tokens . . . . . 73
4.1.3 Sample analyses . . . . . 75
4.2 Examining the unexplained VW3 results . . . . . 81
4.2.1 Using referential structure analysis . . . . . 81
4.2.2 Explaining the VW3 results . . . . . 83
4.2.3 Representational choices in VesselWorld . . . . . 85
4.3 The "visible" VesselWorld experiment . . . . . 89
4.3.1 Modeling and predictions . . . . . 90
4.3.2 Experimental results . . . . . 91
4.3.3 Summary of VVW . . . . . 96
4.4 Visualizing experimental data . . . . . 96
4.4.1 Using visualization to explore the data . . . . . 97
4.5 Limitations of the method . . . . . 102
4.5.1 Collecting data . . . . . 102
4.5.2 Choosing referent categories . . . . . 103
4.5.3 Ambiguity . . . . . 104
4.5.4 Phantom references . . . . . 106
4.5.5 Lost and duplicated references . . . . . 109
4.5.6 Lack of communication . . . . . 110
4.5.7 Excessive diligence . . . . . 110
4.5.8 Hierarchies of knowledge . . . . . 112
4.5.9 Non-stationary work practice . . . . . 112
4.6 Summary . . . . . 113

**5 Experimental Verification 114**
5.1 Correctness . . . 115
5.1.1 Referential structure depends on referent type . . . 116
5.1.2 Representations affect referential structure . . . 117
5.1.3 Teachable . . . 118
5.1.4 Reproducible results . . . 120
5.2 Domain independence . . . 126
5.2.1 Using recurrence analysis for redesign . . . 127
5.2.2 Using referential structure analysis for redesign . . . 129
5.3 Summary . . . 130

**6 Generating Design Recommendations 132**
6.1 Representation . . . 132
6.1.1 Structured representations . . . 134
6.1.2 Representation properties . . . 136
6.2 Modeling information . . . 137
6.2.1 Information flows . . . 138
6.2.2 Information features . . . 140
6.2.3 Defining information features . . . 143
6.3 Recommending representations . . . 148
6.3.1 Representation properties . . . 150
6.3.2 Sample properties . . . 151
6.3.3 Selecting matching representations . . . 154
6.4 Summary . . . 156

**7 The Business Travel Experiment 157**
7.1 Experimental design . . . 158
7.1.1 Analyzing the base group data . . . 159
7.2 Drawing conclusions . . . 161
7.2.1 Designing matching representations . . . 164
7.2.2 Designing non-matching representations . . . 166
7.2.3 Predictions based on referential structure analysis . . . 168
7.3 Experimental results . . . 169
7.3.1 Verifying predictions . . . 171

**8 Conclusions 173**
8.1 Impact of the Methodology . . . 174
8.2 Future Directions . . . 175
8.2.1 Automatic Referential Structure Extraction . . . 175
8.2.2 Expansion of Information Flow Analysis . . . 175
8.2.3 Taxonomy of Representations . . . 176

**A Building blocks for representations 178**

**B VesselWorld Manual** **182**
B.1 Starting Up . . . 183
B.1.1 Logging in . . . 183
B.1.2 The Inhabitants of VesselWorld . . . 183
B.2 Information and Manipulation of VesselWorld . . . 187
B.2.1 Control Center . . . 187
B.2.2 Messages . . . 187
B.2.3 World View . . . 188
B.2.4 Markers . . . 189
B.2.5 Weather Display . . . 190
B.2.6 Information Window . . . 191
B.2.7 Legend . . . 192
B.2.8 Score . . . 192
B.3 Planning and the Planning Window . . . 193
B.3.1 Planning in the World View . . . 193
B.3.2 Planning Window . . . 195
B.4 Coordinating with other Captains . . . 196
B.4.1 Chat . . . 196
B.4.2 Object List . . . 197
B.4.3 Strategic Planning . . . 198

**C Lyze Manual** **202**
C.1 Iotas and the tagging scheme . . . 202
C.1.1 Iota types . . . 204
C.1.2 Grammar . . . 205
C.2 An Example . . . 206
C.3 Practical Issues . . . 208
C.4 Using the Tools . . . 209
C.4.1 The Line Graph . . . 210
C.4.2 Visualizations using a spreadsheet program . . . 211

**Bibliography** **212**

# List of Tables


1.1 Referential structure data for the GHT example. . . . . 13

3.1 Overview of VesselWorld experiments. . . . . 48
3.2 Comparison of VW-CR and VW-NO-CR groups. . . . . 68

4.1 Results from the analysis in Figure 4.2. . . . . 77
4.2 Referential structure data from the VW3 experiment. . . . . 82
4.3 Comparing Visible VesselWorld data to VW3 results. . . . . 93

5.1 Representation system has an effect on information access patterns for some referent types. . . . . 118
5.2 Unweighted kappa values for each group. . . . . 122
5.3 Fair to Good agreement on the "stomach" reference. . . . . 123
5.4 Fair to Poor agreement involving the referential phrase "read the left side". . . . . 123
5.5 Differing levels of detail meant that some analysts tagged different referents. . . . . 124
5.6 Methodological rationales used by student groups for redesign. . . . . 127
5.7 Students designed new representations based on finding new referent types. . . . . 129

7.1 Referential structure data from the Business Travel domain. . . . . 160
7.2 Number of concurrent referents for each type. . . . . 161
7.3 Mapping referent type properties to desired features of new representations. . . . . 163
7.4 Objective results comparing 'non-matching' system to 'matching' system. 169
7.5 Survey results from the initial BT study. . . . . 170

# List of Figures


1.1 Overall design process for the GROUP lab methodology. . . . . 4
1.2 Group Homework Tool interface . . . . . 9
1.3 Analyzing GHT data with referential structure analysis. . . . . 10

2.1 A functional perspective of activity in a burger joint. . . . . 19
2.2 Conversation for Step 1 of the burger example. . . . . 19
2.3 Expanded iterative-design cycle . . . . . 21
2.4 The burger joint, from a Distributed Cognition perspective. . . . . 26
2.5 A simplified hierarchical task analysis of the burger joint. . . . . 30
2.6 Assessing the burger joint from an Activity Theory perspective. . . . 34
2.7 A language/action perspective on discourse in the burger joint. . . . . 38
2.8 Conversation analysis for Step 1 of the burger example. . . . . 40

3.1 The VesselWorld interface. . . . . 50
3.2 The VesselWorld VCR, allowing replay of log files. . . . . 53
3.3 Many groups had difficulty coordinating plan submission. . . . . 56
3.4 Mistakes in recalling waste information lead to confusion. . . . . 57
3.5 Adjacency pairs in VesselWorld dialog. . . . . 59
3.6 Secondary structure created to help waste reporting. . . . . 60
3.7 Marker check reveals a discrepancy in the users' private representations. 61
3.8 The Shared Planning Window. . . . . 64
3.9 The Object List. . . . . 65
3.10 The Strategic Planning window. . . . . 65

4.1 Referential structure analysis the burger joint. . . . . 72
4.2 Applying referential structure analysis. . . . . 75
4.3 Analysis of data from the GHT experiment (see Figure 1.2). . . . . 78
4.4 Users spent much of their time negotiating plans for waste retrieval. . 80
4.5 Path of waste info in the VW-NO-CR system. . . . . 87
4.6 Path of waste info in the VW-CR representation system. . . . . 89
4.7 Path of waste info in the VVW-NO-CR system. . . . . 91
4.8 Information feature differences are made visible in a scatter plot. . . . 97

4.9 Scatter plot of location referents reveals clusters of deictic vs. definite references. . . . 98
4.10 Scatter plot of waste referents reveals short-lived wastes to be 'phantom' referents. . . . 99
4.11 Scatter plot of plan referents revealed no obvious structure. . . . 100
4.12 Histogram of plan referents vs. percent lifetime. . . . 101
4.13 A phantom referent is born, wreaks havoc, is tracked down, and finally expunged. . . . 108

6.1 Scales for observed information features. . . . 141
6.2 Design recommendations stemming from specific levels of individual information features. . . . 149

7.1 Itinerary produced by a test subject in the Business Travel study. . . . 159
7.2 The Task Table, designed to match observed work practice. . . . 165
7.3 The Plan Table, designed to conflict with observed work practice. . . . 167

B.1 The login window . . . 183
B.2 An inhabitant and its zone . . . 183
B.3 Cranes, equipment, and joined operation . . . 184
B.4 The tug . . . 185
B.5 A barrel of toxic waste . . . 185
B.6 A leaking barrel of toxic waste . . . 186
B.7 The large barge . . . 186
B.8 A small barge . . . 187
B.9 The control center . . . 188
B.10 The marker list . . . 189
B.11 Adding a marker . . . 190
B.12 The weather window . . . 190
B.13 The information window . . . 191
B.14 The legend window . . . 192
B.15 The score window . . . 193
B.16 The planning window . . . 195
B.17 The chat window . . . 197
B.18 The object list window . . . 197
B.19 The strategic planning window . . . 198

C.1 A view of the Lyze tool. . . . 210

# Chapter 1

# Introduction

## 1.1 Overview

This thesis demonstrates a technique for building a model of interaction by tracking the references that participants make and using them to infer features of information exchange within the system. The referential structure of discourse is used to model the interaction of participants in an ongoing, online task. This model gives an analyst insight into features of the information flows within the system, and provides the basis for recommending new representations that reduce the collaborative effort of participants.

This thesis details the theoretical underpinnings for the analysis technique, including applying the technique to a number of examples. It also presents experimental evidence demonstrating the validity of the approach. This technique has been utilized in the analysis and redesign of a number of groupware systems. It has also been used to predict the outcome of a design process. Finally, the technique was taught to students, who were able to use them to redesign groupware systems of their own devising.

### 1.1.1 Motivation: designing groupware

Computer-mediated collaboration has become ubiquitous. The domain of computer-supported cooperative work has a long and complex history [13]. Remotely-taught educational courses, collaboration with work divisions in remote locations, and coordinating military personnel distributed across a net-centric battlefield are all domains where same-time / different-place [36] interactions can become difficult. Participants in different locations have access to different physical environments; hence, the methods participants use for interacting are different than those used in face-to-face interaction. Procedures for referring to, pointing at, modifying, and reviewing objects, as well as gauging the focus and intent of other participants must necessarily be altered when participants interact online.

However, despite the best efforts of designers, groupware applications often end up interfering with or fundamentally altering the very work they are designed to support [43, 53, 98]. It is crucial, but difficult, to provide a system which matches the needs of the participants. One reason for this difficulty is that, as participants perform a joint activity, a work practice emerges which can be hard to foresee. As participants interact, they formulate procedures and methods for doing their work that generally diverge from the practice as preconceived by task designers.

Ideally, software would be built to match this practice [57, 105, 127]. However, it is usually difficult to predict what this work practice will be, and is likewise difficult to predict what impact a new groupware system will have on that practice. It therefore becomes necessary to perform an ethnographic analysis of work in practice which gathers interaction data of participants as they perform their tasks.

*Tools to understand this data have historically been either very abstract and hard to apply, or very specific and only usable on specific domains.* The goal of this thesis is to

provide techniques which model the interaction by analyzing information flow between representations, in a fashion that can be used to produce concrete recommendations for design. This work is part of a research project aimed at understanding and constructing collaborative software. Some other important threads of this research include: creating software that generates complete, replayable transcripts; building adaptive system components which simplify inference of user intent; and the primary work of this thesis, building analysis methods which generate models of an interaction.

The techniques discussed in this thesis combine concepts from distributed cognition, conversation analysis, and discourse analysis to create methods that model the situated activity of participants. By means of a rigorous examination of the structure and content of user discourse, an analyst can identify specific features of information flows within the system, and use this model to generate concrete recommendations for system redesign. These methods analyze transcripts of interaction to answer questions like: What types of information do participants exchange? How do participants access different types of information? What areas of the interaction are problematic? Where, and how, does the system need to be modified to reduce coordination effort?

### 1.1.2 Design methodology

Research to answer these questions, including the efforts outlined in this dissertation, was undertaken as a part of the GROUP lab under the direction of Dr. Richard Alterman. GROUP (the Group for Research of Online Usability Problems) was formed to explore a number of related concepts to achieve these goals. The lab's research examines problems in same-time / different-place coordination through construction of groupware and explores the theoretical background of joint sense-making.

As a part of its goal of inquiry into groupware design, the GROUP lab has established a standard method for designing new systems. This experimental methodology

combines software engineering with ethnographic analysis of usage data. The GROUP lab's methodological approach to analysis and design of a system to support an ongoing interaction is shown in a simplified fashion in Figure 1.1. As can be seen in the figure, the redesign process begins with a rapid assessment of the ongoing work practice. From this, a base system for supporting the interaction is created using building blocks made available by a groupware toolkit such as THYME. This system is not intended to solve coordination problems within the practice; rather, its purpose is to provide a minimally-intrusive means of collecting accurate ethnographic data about the work practice.

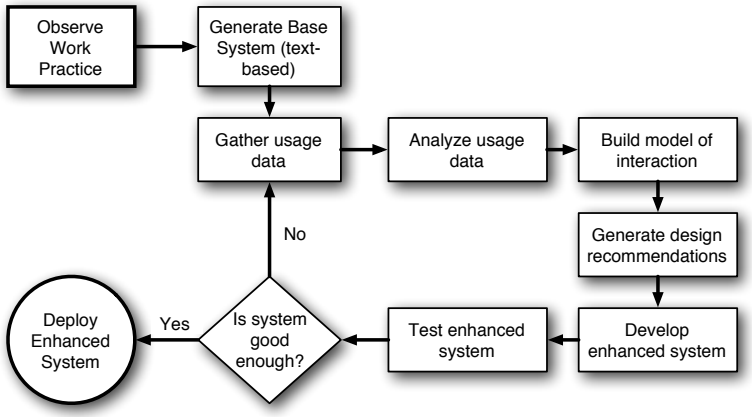


**Figure 1.1: Overall design process for the GROUP lab methodology.**

This basic system is then used to gather information via recording of interaction into complete, replayable transcripts of usage data. Our techniques for analysis are applied to the data, generating observational conclusions and recommendations for redesign. These recommendations are used to select new representations and integrate them into a useful, enhanced system which matches the observed work practice. This new system is then re-assessed, using the same usage data as before, and if determined to be sufficient (as appropriate to the work context), is deployed; otherwise, the redesign cycle can continue until a satisfactory system is created.

Research to date has focussed on three major areas:

**Constructing Component-oriented Groupware** The goal of this research track was to produce a toolkit for creating groupware rapidly and efficiently. This research was primarily the work of Seth Landsman [75, 76, 77]. The end product of this research was THYME, a framework of reconfigurable groupware components which can be combined into a working application. Thyme-generated groupware applications automatically generate complete, replayable transcripts of the users' online activity. The data is collected in a format that can be replayed by a VCR-like program; Thyme also provides the capability to semi-automatically create a replay application (SAGE) customized for the particular domain.

**Inferring User Intent via Structured Representations** This research, primarily the work of Joshua Introne [61, 62, 63], examines the way that structured representations can be used to infer user intentions. Because these representations are structured, giving computer-readable tags to information, it is easier for AI-type inference algorithms, such as Bayesian networks, to reason about user intent. This dual-purpose language of representations can help bridge the tool-agent gap and be used to create systems which provide more autonomous support for interaction. This work aids generation of an enhanced system by providing additional available capabilities for automating work processes based on inference of user intent.

**Analysis of Representational Work in Interaction** This track of research, which I will discuss in this dissertation, is aimed at understanding collaboration by analyzing the representation work of participants. Past work has built up a theoretical background based on distributed cognition of how participants structure their talk to coordinate their actions using *coordinating representations* [6], how these rep-

resentation affect the interaction of participants [5], and how to analyze this impact [39]. More recent work has looked at topics such as collaborative learning and the nature of intersubjectivity [4].

## 1.2 Theoretical background

Suchman and Trigg [127] stated that for design to be successful, it must be based on a close examination of the actual work environment:

> "Where technologies are designed at a distance from the situation of their use, as most are, there is an inevitable gap between scenarios of use and users' actual circumstances. ... What we see consistently is that the closeness of designers to those who use an artifact ... directly determines the artifact's appropriateness to its situation of use."

This perspective indicates the need for an analytic method to be able to examine an interaction *in situ* in order to accurately capture it. This requires a non-invasive, yet adequately perceptive, analytic technique. To accomplish this goal, our lab decided to focus on discourse, a feature of the interaction which is simultaneously one of the most revealing and one of the most accessible. To precisely model the discourse, I turned to three major bodies of work: conversation analysis, distributed cognition, and discourse analysis.

Conversation analysis [107, 108, 115] is a social theory that examines the structures that arise out of human interaction, notably conversational interactions. It posits that participants create order within their interaction so as to be able to work together; this order serves to orient the participants to their joint action, providing participants with an ability to better estimate the future actions of other participants. This order, once created, is amenable to being analyzed and modeled in formal terms.

Conversation analysis says that participants create social order in their conversational interaction. Hence, studying conversation is one way to understand how people are organizing their behavior. Conversation analysis provides a way to understand interaction, but does not provide a way to populate the content of the model for a particular interaction.

Distributed cognition [57, 58, 59, 60, 106] applies cognitive science analysis techniques to representation in a system of more than one individual. Hutchins [59] explains:

> "... the classical cognitive science approach can be applied with little modification to a unit of analysis that is larger than an individual person. ... we wish to characterize the behavioral properties of the unit of analysis in terms of the structure and processing of representations that are internal to the system. ... The analysis presented here shows that structure in the environment can provide much more than external memory."

This perspective allows the analyst to examine representations that are internal to the system, but external to the minds of the participants. The contents and form of these external representations are available for direct inspection, simplifying the analysis task. Hutchins and Klausen [60] write:

> "We can see that the information moved through the system as a sequence of representational states in representational media. From speech channels to internal memories, back to speech channels, to the physical setting of a device. Its representation in each medium is a transformation of the representation in other media."

That is, information can be viewed as flowing from one representation to another. We term this theoretical construct an *information flow* — a directed communica-

tion of information from one representation to another. These information flows are themselves mediated by the representations they occur in. Seen from this perspective, conversation is a representation for information whose contents are open to analysis. Analyzing the content of discourse can therefore reveal important details about the interaction.

To analyze the content of discourse, I used techniques from discourse analysis, specifically, coreference chaining. Lockman and Klappholz [82] presented a model of language understanding which expanded the traditional definition of "reference" to include connection by almost any semantic method; they call this "contextual reference". They state:

> "we are *defining* contextual reference to be the phenomenon whereby an (any sort of) item in the semantic representation of a text has the property that for the text to be fully interpreted the reader must recover some (any sort of) intended, but not explicitly stated, connection (relation) between that item and some other item in the *semantic representation of the text.* [original emphasis]"

This broad definition of reference gives a technique with which to track conversational items in the discourse by connecting successive references to an item. I have utilized this technique to draw conclusions about the way people talk about conversational items, by computing various statistics for these chains of reference, such as length and density of reference. This method produces a model of information flows within the system. I call this method referential structure analysis.

## 1.3 Referential structure analysis

*Referential structure analysis*, the analysis method presented in this thesis, is concerned with determining what types of information participants discuss, and how they communicate, refer to, and store task information. The method builds contextual coreference chains by tracking specific conversational items as participants refer to them.

To illustrate use of the method, we will present an example taken from one of the experimental domains we analyzed using these methods, the Group Homework Tool [78]. This project, the work of Dr. Tim Hickey and his students, studied the educational impact of having students program in pairs versus solo programming. I was brought in to help analyze results and assist in redesigning the system.

Two students are situated at two remote computers, and asked to complete a coding assignment together using a chat tool, a shared editor, and a pair of shared web browsers. Before and after the assignment, the students are individually tested on their programming skills; the results are then compared to examine knowledge transfer during pairs programming.

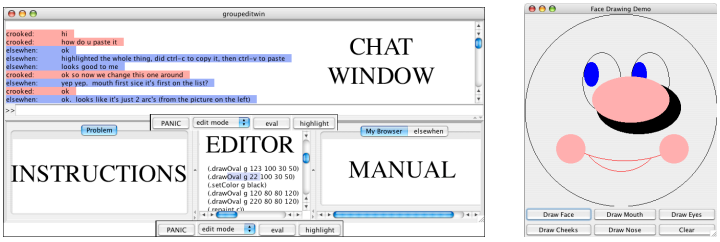


**Figure 1.2: Interface for the Group Homework Tool experiment; to the right, the face produced by the code.**

A screenshot of the Group Homework Tool is shown in Figure 1.2. In the center is a shared editor; above it is a chat window that the students can use to communicate. On either side are shared browsers to view the instructions and the reference manual

for the programming language. All communication and actions are recorded for later analysis. The assignment in this case was to draw a cartoon face, with various specified features such as a nose, mouth, and so forth. A sample result is shown on the right side of Figure 1.2.

Figure 1.3 shows a portion of the chat transcript generated by the pair of students that drew the face shown above. This segment starts near the beginning of the session, as the students discuss the instructions and manual. Note that all dialogue is copied exactly as it was typed, with elisions, misspellings, and the like retained intact; necessary interpolations of the dialogue are indicated with square brackets.

| | | Discourse | *mouth* | *plan* |
|---|---|---|---|---|
| 7 | B: | yep yep. mouth$_a$ first$_b$ sice it's$_c$ first on the list? | a, c | b |
| 8 | A: | ok$_a$ | | a |
| 9 | B: | ok$_a$. looks like it's$_b$ just 2 arc's (from the picture on the left) | b | a |
| 10 | B: | although I'm not sure what the parameters for .drawArc are..... | | |
| 11 | A: | how will we be abe to see if its$_a$ correct | a | |
| 12 | B: | well, [it]$_a$ doesn't have to be 100% correct i'm guessing..[it]$_b$ just has to look similar | a, b | |
| 13 | A: | ok | | |
| 14 | B: | so we just use the eval. button and pray it$_a$ looks ok :) | a | |

**Figure 1.3: Analyzing GHT data with referential structure analysis.**

In this example, a pair of students have completed their pre-test and are beginning their problem-solving session. The two students, who have not met before, are of disparate skill levels, and as a result user B ends up tutoring user A for most of the session. At the start of this excerpt, user B has copied and pasted some sample code from the instructions into the shared editor. The two users begin by discussing

what part of the multi-part assignment to do first, and then move on to discuss implementation details. The students are provided with some sample code, which they use as a starting point. The conversation quickly turns to a discussion of how to address the first portion of the assignment – using the `drawArc` function to draw a cartoon mouth, and using the 'eval' button to evaluate the code to fine-tune the appearance of the mouth.

### 1.3.1 Analyzing the dialogue

Referential structure analysis involves tagging references in the discourse and combining them into coreference chains. The aim of the method is to discover what sorts of things participants in a joint activity spend their time talking about, how much they talk about them, and for how long. For example, an analyst might investigate how frequently participants refer to some domain object, or how long they spend discussing a plan for action. To do this, the analyst tags the references made by participants and consolidates them according to the referent they refer to. Subsequent references are examined and either attached to existing referents, or treated as the first reference to a new referent.

As they are tagged, referents are assigned types. These types allow investigation of the differences between various sorts of information. Types can be domain-specific — in this case, "mouth" might be categorized as referring to a "coding task" — or domain-independent, including the "plan" to draw the mouth seen above. By aggregating statistics over all referents of a particular type, the analyst can analyze how these different types of information differ; in general, each type of information will have distinctive characteristics for how participants access it.

Figure 1.3 shows the results of tracking two referents through sample dialogue from the GHT domain. There are many possibilities for reference: code constructs,

elements of the instructions, plans for action, the shape of the desired output, division of labor, and so forth. For clarity, the figure shows only two referents from the dialogue, and marked each reference with a unique subscript; references are sorted into separate columns depending on which referent they point at.

The first referent examined is the "mouth" referent, referring to a portion of the face which the code is meant to produce. References to it are collected in the second column from the right. On line 7, B refers to it by name ("mouth"), followed by an anaphoric reference ("it's"). On line 9, B refers to the mouth — in this case, the picture of the mouth provided in the instructions. Discussion on lines 11 and 12 refers to the mouth a number of times, including by means of elided pronouns; and at the end of the dialog on line 14, B refers to the mouth once more. From this, it is clear that the mouth referent remains relevant for some time — in this segment, the first reference to it is on line 7, and the final reference is on line 14. (Conversation about it continues past this brief segment of dialogue; this analysis is restricted for didactic purposes.) For this segment of discourse, it has a lifetime of relevance to the participant of eight utterances (inclusive), and is referred to seven times in five separate utterances during that lifetime.

Participants also discuss a plan for drawing the mouth at the start of the extract. This "plan" referent is referred to differently than the "mouth" referent. References to this plan referent are noted in the right-most column of Figure 1.3. On line 7, B proposes the plan ("mouth first [...] ?"); on line 8, A accepts ("ok"), and on line 9, B acknowledges acceptance ("ok"). This constitutes the entire conversation about this plan. In contrast to the mouth referent, the plan for ordering tasks is relevant for three utterances (lines 7–9), but after this is never discussed again. Although the plans continues to be a topic for discussion — lines 10–14 are primarily concerned with how to carry out the plan — the participants have committed to it and do not

refer to the plan itself again. This particular plan referent has a short lifetime of relevance (three utterances), and is mentioned three times over that span.

Once a body of referent data has been generated, the analyst can start to draw conclusions about information types. At this stage the analyst can compute a number of measures based on the referent data, including:

- Frequency: what percent of all referents are of a particular type
- References: how many lines contained a reference to this referent
- Lifetime: lines, inclusive, between first and last reference to this referent
- Density: the ratio of references to lifetime; a measure of conversational importance

The results of computing these statistics for the limited example in Figure 1.3 is shown in Table 1.1. Obviously, at this small scale, the statistics are suspect; however, they do serve to illustrate the method. The contrast seen in this figure — a long lifetime versus a short one, higher density versus lower — is the sort of observation the method is designed to find. By examining these patterns of reference to referents, the analyst can gauge how participants are exchanging information, and determine appropriate representations for supporting this exchange.

| Type | Frequency | References | Lifetime | Density |
|---|---|---|---|---|
| Coding Task | 50% | 5 | 8 | 62.5% |
| Plan | 50% | 3 | 3 | 100% |

**Table 1.1: Referential structure data for the GHT example.**

### 1.3.2 Using the methods to inform redesign

The model generated by referential structure analysis can be used to provide redesign recommendations. Referential structure analysis reveals features of information flows within a distributed cognitive system. This thesis has identified a handful of information features which can be measured using referential structure analysis. By establishing values ("low", "high") for each of these features (e.g., "read frequency"), the analyst can build a model of each important information flow. The method then recommends representations whose properties match these information features. This matching process has been shown to produce systems which improve performance.

In the above example, the "mouth" referent, which could be given the type "coding task", has a long lifetime of relevance. Coupled with the repeated access to the information throughout its lifetime, this feature indicates the need for a representation which is *persistent* and *easily readable* and *modifiable*. For example, a on-screen list of current coding tasks which can be seen and edited by both users could serve to reduce collaborative effort. In contrast, the "plan" referent, and its putative brethren, have a relatively short lifetime of relevance, with a very high density during that time. This means that a persistent representation is unlikely to be necessary — participants will likely consider wasted any effort required to encode the plan information into this external representation, and will end up not using that representation. In this way, an analyst can use the model of each information flow to select appropriate representations.

## 1.4 Overview

This thesis presents an analysis method, referential structure analysis, which builds a model of interaction by examining the discourse generated as a part of an ongoing

interaction. This model identifies specific information features within the discourse, such as lifetime of relevance, for information items of a specific type. By connecting these information features to specific properties of a representation, (a long lifetime of relevance might imply a persistent representation), the analyst can use the model to select appropriate representations for supporting the interaction. In aggregate, these representations aid task performance, reducing the time and effort necessary to complete tasks and reducing overall error rate. The methodology represents a strong tool in the arsenal of system analysts and designers with which to create new, useful groupware systems.

This thesis presents experimental evidence for each step of the methodology. Subsequent chapters will follow the evolution of the methods from creation, through refinement, application, and finally confirmation. The thesis will continue with an examination of related work, including a summary of theoretical frameworks for analysis and practical methodologies for designing groupware in Chapter 2.

The second part of the thesis discusses the history of the creation and refinement of the analysis methods. Chapter 3 gives background information for the methods that lead up to the development of referential structure analysis, and demonstrates their use on some samples from VesselWorld. Chapter 4 explains the theoretical basis for referential structure analysis, demonstrates its use, and shows its utility by giving reasons for results previous methods were unable to explain. The method is also used in a predictive capacity, explaining an experiment testing intuitions about how system usage is altered by small changes in representation.

The next part of the thesis presents experimental evidence for the utility of this approach. Chapter 5 presents experimental evidence of the general utility of the method and its ability to be taught to other analysts. This chapter details the process of teaching the methods to a class of students, and demonstrates the ability of the class

to use the methods to generate and redesign software systems. These experiment also showed the ability of independent coders to use the methods to produce comparable results from identical transcripts.

The final segment of the thesis presents techniques for using referential structure to aid in groupware design. Chapter 6 discusses the technical details of how discourse features can be quantified and used to select representations. The Business Travel Experiment, covered in detail in Chapter 7, demonstrates the ability of the methods to predict use of representations. Finally, Chapter 8 concludes the thesis with a discussion of possibilities for future research building on these techniques.

# Chapter 2

# Related Work

## 2.1 Designing groupware

Computer-mediated collaboration, or computer-supported cooperative work (CSCW; [13]) has become ubiquitous. Remotely-taught educational courses, collaboration with work divisions in remote locations, and coordinating military personnel distributed across a net-centric battlefield are all domains where same-time / different-place [36] interactions can become difficult. Participants in different locations have access to different physical environments; hence, the methods participants use for interacting are different than those used in face-to-face interaction. Procedures for referring to, pointing at, modifying, and reviewing objects, as well as gauging the focus and intent of other participants must necessarily be altered when participants interact online. The effective procedures for the maintenance of common ground are significantly modified even when high-fidelity technology such as video conferencing is used [26, 27]. The goal of a groupware system is to support these altered procedures, and in doing so allow participants to perform their root task efficiently.

However, despite the best efforts of designers, groupware applications often end up

interfering with or fundamentally altering the very work they are designed to support [43, 53, 98]. It is crucial, but difficult, to provide a system which matches the needs of the participants. One reason for this difficulty is that, as participants perform a joint activity, a work practice emerges which can be hard to foresee. As participants interact, they formulate procedures and methods for doing their work that generally diverge from the practice as preconceived by task designers. Ideally, software would be built to match this practice [57, 105, 127]. However, it is usually difficult to predict what this work practice will be. It is likewise difficult to predict what impact a new groupware system will have on that practice. Predicting the impact of a new representation system requires a strong model of the existing interaction. For this thesis, I based my work on the model provided by Distributed Cognition. I used techniques from Discourse Analysis to garner evidence for this interaction model, and to populate it with data. Finally, I incorporated an understanding of the impact of representations based on work in representation theory to translate my model of interaction to redesign recommendations.

### 2.1.1 Example: the burger joint

It is helpful to use a concrete example to discuss the various ways of analyzing and modeling interaction. A graphical view of an example — the burger joint — is shown in Figure 2.1. The domain of the example, chosen for its familiarity, is the ubiquitous neighborhood restaurant, with a customer interacting with a cashier and a cook to order and receive a burger. For clarity I have chosen to focus on a single interaction within the larger activity.

In this functional view of the interaction, a customer talks with the cashier to order a burger (1); the order is received by a cashier, who communicates it to the chef (2); and the chef prepares the burger to order (3), plating it for the cashier (4),

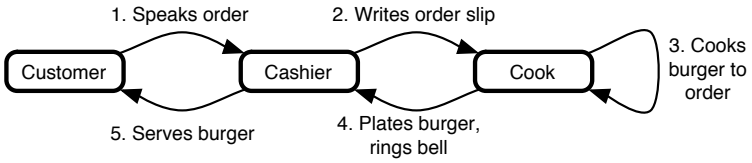


**Figure 2.1: A functional perspective of activity in a burger joint.**

who then serves it to the customer (5).

The burger joint is amenable to many levels of analysis, making it useful for explaining the utility of the various analytic techniques covered in this chapter. To provide more detail for the discourse-based analytic methods, a putative conversation for step (1) was created. This conversation is shown in Figure 2.2.

1. Cashier: Can I take your order?
2. Customer: Uh, yeah, I'll have a burger.
3. Cashier: What's on it?
4. Customer: Umm... extra mustard and pickles.
5. Cashier: Coming right up.

**Figure 2.2: Conversation for Step 1 of the burger example.**

## 2.2 The design process

In the past two decades, the focus of the design process has increasingly been on including the end-user in the design process, either as a source of inspiration, an occasional participant, as a test subject, or, as in this thesis, as a source of interaction data to drive design.

Originally, the design process was modeled as proceeding linearly from customer requirements through implementation to deployment. "Waterfall" design, a retronym created to distinguish this process from the newer "iterative" design process, is design viewed as a linear process from user requirements to design to testing to deployment. As this process requires designers to accurately extrapolate design criteria from users'

expressed requirements, and then create software that matches those requirements with little feedback from users themselves, the opportunity for misinterpretation and incorrect design is very high. Even extensive testing of software designed in this way — long usability tests, thorough Quality Assurance testing – generally can only serve to polish an imperfectly shaped stone. Despite advances in knowledge elicitation, it is very difficult and perhaps unrealistic to expect to gain a thorough understanding of a domain without watching users perform their tasks in their native environment.

"Iterative" design [49] starts again from user requirements, but attempts to include the user in the design process by repeatedly presenting prototype designs for approval and suggestion. This presentation is generally done as a demonstration, rather than as an actual deployment for producing work, and as a result feedback is of somewhat limited quality. In addition to simply increasing the amount of unfocused feedback hinted at in the previous paragraph, test users quickly gain a sense of entitlement over the design of the tool — when called upon to make suggestions, users expect to have those suggestions implemented, and can quickly become non-cooperative if their suggestions are apparently ignored. Finally, iterative design, by its nature, quickly becomes very expensive – effectively, it is repeated waterfall design, and each repeat multiplies the cost of development with successive decreases in the benefit.

Figure 2.3 shows the expanded view of iterative design used by our lab. The traditional iterative design 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 [97]. Hence, it is imperative to reduce the number of revisions necessary in constructing a successful product.

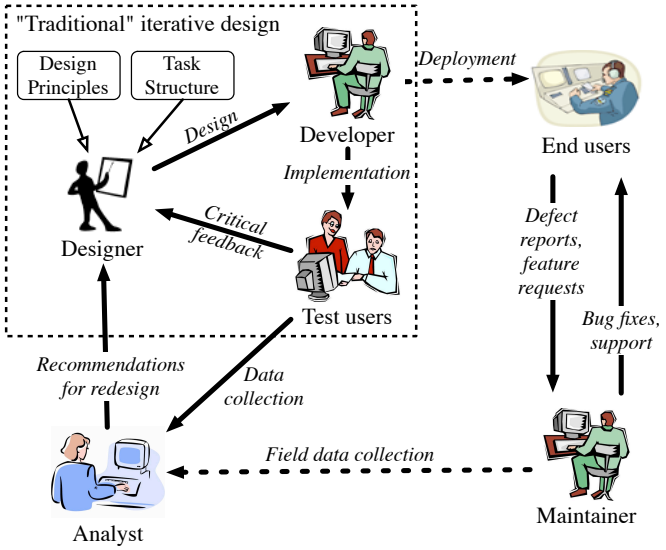


**Figure 2.3: Expanded iterative-design cycle**

This expanded view of iterative design has become more popular in recent years given the long lifetimes of software. In this view, deployed applications require constant updating and and maintenance by some maintainer. Frequent new releases containing bug fixes and enhancements are seen as one of the continuing costs of developing software. However, the opportunity for feedback in this expanded cycle is often overlooked. Reports from end users contain implicit information about use of the application that can be used to improve the application. This information can be used by an analyst to generate recommendations for redesign of future software version. These sources of information — usage data from both test users and end users — are an additional resource for analysts and system designers. However, like other ethnographic data, they require concrete methods to be used in generating redesign recommendations.

Participatory design [50, 88, 89, 121] attempts to address this by including users at more stages of the design process. By including users throughout the design process, instead of simply at milestones, the goal is to produce designs which have a higher

chance of being usable and useful. Again, the benefit to the designer of users involved in the design process quickly drops off: user biases such as familiarity with previous design iterations and feelings of entitlement toward old suggestions deteriorate the quality of feedback. Additionally, participatory design can be quite expensive, due to the necessity to both develop and to extract and implement user feedback continuously, with no guarantee of an increase in quality of results.

### 2.2.1 Prescriptive rulesets

At the other end of the spectrum is making use of the existing body of literature regarding proper design techniques. While this approach does not make use of user input directly, it can be very rapidly and inexpensively applied. As a result, making use of a prescriptive ruleset for system design is one of the most popular mechanisms for examining and redesigning an interaction.

There are a variety of prescriptive and proscriptive guidelines available. These have been produced over the years to share hard-won insights into system design: Shneiderman's basic rules for design [123], Cooper's landmark book on UI design [32], Norman's principles for design of interfaces [94], gatherings of design patterns for interaction (e.g., [18]), 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. [9] for software developers. Modern website design, still a work in progress, has been strongly influenced by the work of Nielsen [93], among many others; Nielsen's tips give specific interface guidelines ("Put navigation links at the top of the page", "highlight keywords", and so forth) which encode lessons learned from user feedback, successful designs of other websites, and psychological underpinnings. Shneiderman's "Eight Golden Rules of Interface Design" [123] and Cooper's case-by-case description of use of interface widgets [32] give a solid foundation for proper design. They help a designer enhance the user experience by encoding and highlighting many common mistakes, and by providing time-honored solutions for common problems.

However, prescriptive 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. Prescriptive rulesets are more valuable to novice designers than to expert ones; however, novice designers have a tendency to apply the rules they are given overzealously, impairing good design. Many frameworks tend to be proscriptive, rather than prescriptive — that is, they say what not to do, but leave the designer with less feedback about how to resolve problems. Finally, being abstract, they cannot speak to specific problems in a particular interaction, only to classes of problems that occur in general.

### 2.2.2 Design tradeoffs

As with all design, a compromise between thoroughness of domain investigation and efficiency of analysis had to be struck. One approach was what were termed "discount" usability measures [91, 92]. These measures were chosen to be less expensive in terms of user involvement, analyst effort, time, and money than full-fledged field trials and iterative design. These approaches aim to model the user's interaction with the software in some way, and then verify that model via a low-cost (small-scale, short-

term) usability study, or through pointed interviews (as opposed to broader-spectrum user interviews to define a task domain). Methods such as employing simple paper-and-pencil prototyping, small focus groups, and rapid iteration of usability methods can reduce the cost to evaluate designs. These approaches demonstrated a 'sweet spot' in the number of usability trials and design cycles where further inspection yields diminishing results.

It is worth noting that participatory design methods generally include observing users outside of their work routine in a controlled setting, and incorporate direct user suggestion and reaction into design. My methods use a different source of data, namely, the communications of the users as they interact in their native setting. As a result, the two methods can be used in a complementary fashion: referential structure analysis can be used to gain an understanding of the domain in conjunction with user interviews and domain task analysis; the direction given by RSA can be used to design prototype representations, which can then be refined via user-centric design methodologies.

## 2.3 Modeling interaction

Design needs to be based on a model of user activity and interaction to be successful. Approaches to modeling interaction vary primarily in the main focus of their analysis and in their scope. Distributed Cognition, which this thesis is based on, builds a model by examining the representations of information within a system. Task Analysis methods break interaction down into individual tasks, then build up a model of the interaction from these atomic pieces. Workflow Analysis centers around an understanding of task objects which are successively transformed by workers. Finally, Activity Theory examines activity-level behavior and couples it to social issues such

as division of labor and conflicts of interest to achieve a high-level model of an ongoing situation. This section will discuss each of these major approaches in turn, examining what each has to offer with regards to modeling online interaction.

### 2.3.1 Distributed Cognition

Distributed Cognition [42, 57, 59, 95, 106] provides a way to model interaction by examining the impact of representations into a system of behavior. Distributed cognition views representations both internal to participants, and externally realized in artifacts, as important pieces of the working system. 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. One of the key concepts is that the unit of analysis as human actors plus the external representations and tools around them allows a more thorough view of the way humans distribute work and knowledge into their environments [1, 96, 99]. This distribution of cognition into the environment allows a enhancement of a worker's abilities beyond what they would able to do, cognitively, with only the resources available internally. The working system can therefore be treated as a collection of representations plus the flows of information between representations [60].

Our methods utilize this view to ground discussion of information flows and the procedures and representations for supporting them. 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. In an ongoing interaction, these procedures may be embodied into artifacts which encode historical solutions to recurring problems [31].

Artifacts that mediate interaction in this way can have a great impact on how

efficient, effective, and pleasant that interaction is, by altering the opportunities for interaction between participants, and the qualities of that contact. Distributed Cognition therefore 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 [58, 60], air traffic control [54], navigation of a ship [59], and so forth.

In the 'burger joint' example from Figure 2.1, a number of representations are identifiable. These representations and the information flows between them can be seen in Figure 2.4. First, the customer begins with a partially or fully realized internal representation of what sort of burger he wants (1). He uses natural language, an unstructured medium, to communicate this information to the cashier (2). The cashier reinterprets this language internally (3), and then transcribes the information onto the order slip (4); the order slip is an external, shared representation with a specialized language. She hands this slip to the cook, who reads and reinterprets the slip internally (5), and formulates a plan for action based on his understanding of this information. Note that this is a somewhat idealized view of the process; there is opportunity for back-channel feedback, clarifications, and other such complications within the process. In practice this diagram would be generated based on careful observation and investigation of the actual work practice.

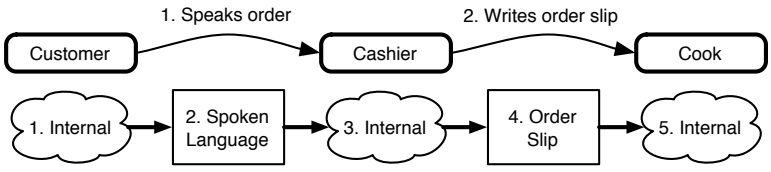


**Figure 2.4: The burger joint, from a Distributed Cognition perspective.**

In the methods presented in this thesis, the information flows (the conversation between customer and cashier, and the communication between cashier and cook

as mediated by the order slip) are examined carefully to determine the important features of each. A designer can then construct representations which support this communication by identifying representations which match the information features. For example, after examining the information flow between cashier and cook regarding burger condiments, a system designer might construct a more structured representation for this information, such as the order screens used in fast food. The methods presented in this thesis give a step-by-step approach to generating such redesign recommendations, a feature missing from 'pure' distributed cognition.

### Using Distributed Cognition: the Resources model

Wright et al. [40, 154, 153] applied distributed cognition to human-computer interaction, creating the concept of an "information resource". Information resources are specific types of information which are central to an interaction between people. They also identify a limited number of resource types, and demonstrated how they could be represented differently in an interface with resulting effect on performance.

The information structures identified by the Resources model include such things as expressions regarding plans, goals, affordances, history, relations between action and effect, and state; these types can be described independently of how they might be represented in an information representation internal or external to a participant. This allows for the concept of alternate representations of types of information, a concept which is also used in this thesis.

To test the impact of providing representations for information types, the researchers provided users with alternate representations of information for a specific task [29], in a fashion similar to, e.g., work by Chuah et al. [25]. The researchers noted the average user work (task time, and execution steps) required to perform a task given each representation. Representations were deemed well-suited to a par-

ticular type of information if they caused a reduction in the overall task time or execution complexity. Furthermore, the impact of the representations on the work was explained by noting how the specific information structures were represented by the interface, and noting whether the affordances of the representations subjectively matched the task performance.

The Resource model would analyze the burger joint example in terms of its "information resources". For example, information comprising the burger order, passing from cashier to cook, could be considered an information resource. Alternate representations could be conceived for this information resource: the order slip, versus a verbal ordering mechanism or a more structured mechanism such as the order screens used in fast food. This method would give a way to test the efficacy of these alternate representations.

While it laid important theoretical background which is relevant for this thesis, the Resources Model provides an analyst with only second-order tools for assessing the impact of a representation change. Only *ex post facto* analysis of the impact of introducing a new representation on a system was expected, with no inferential power about how to assess the work users were doing in a principled fashion so as to select appropriate representations. Additionally, because of the lack of an ethnographic step in gathering knowledge about information structures for a task as actually executed in situ, it is impossible to use these techniques alone to design systems that adequately support an emergent work practice. Nevertheless, this approach represents a significant influence on the methods presented in this thesis.

### 2.3.2 Task Analysis

*Task analysis* (see, e.g., [70]) can also be used to build a model of interaction. While I did not make use of task analysis in the methods presented in this thesis, this

important analytic perspective can be used in conjunction with referential structure analysis to build a more complete model of the interaction and guide interface design. Task analysis is a key step in building effective software systems, but works best when used in conjunction with an analysis of the underlying concerns and motives of participants. While task analysis often is not based on ethnographic data, impairing its applicability, this is an expansion of the method that is used in methods such as Groupware Task Analysis, discussed below.

Traditional 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 as 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 discover difficulties such as missing steps, mistimed steps, bottlenecks, and redundant work. From this model, a new design for the task can be constructed. *Cognitive Task Analysis* (CTA; e.g., [120]), a descendant of pure task analysis, adds a personal-cognition aspect to task analysis. It has also been applied to designing human computer interaction [22]. This method reveals to a certain extent 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.

A simplified hierarchical task analysis of the burger joint is shown in Figure 2.5. Here, the customer's perspective of the interaction is shown as a high-level plan for action. Breaking down the interaction in this top-down fashion gives an analyst a framework within which to place observations. Creating the framework can reveal hidden preconditions and in general requires an analyst to think carefully about the structure of the activity. However, on its own, task analysis does not capture the

richness of the information passing within an interaction. There are a number of extensions to task analysis which utilize this general perspective as the basis for a more complete analysis method; these are detailed below. For example, at the next level of analysis, Cognitive Task Analysis would include an understanding of the cognitive issues surrounding each task, based on interviews with task performers.

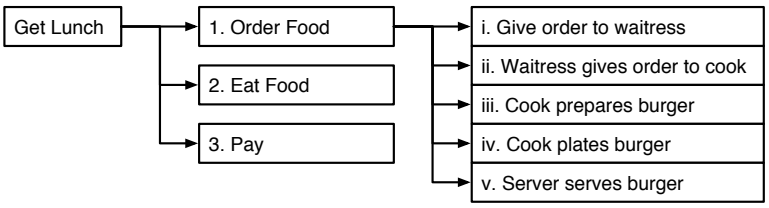


**Figure 2.5: A simplified hierarchical task analysis of the burger joint.**

*Time/space analysis*, based on research dating back to the beginning of the 20th century [133], examined the tasks of workers in an attempt to make factories more efficient. By observing the work in practice that the laborers performed, and assessing the physical actions necessary to perform each task, 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. The worker repacked 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 reduce the workload of the high-skill bricklayer positions, Taylor's method was able to improve the overall speed of production at little incremental cost.

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. In the burger joint domain, fast-food restaurants have optimized every step of the process, from honing the ordering script to replacing the order slip with a computerized readout to structuring the recipes for constructing properly designed burgers out of carefully measured ingredients.

*GOMS* [66, 65] (Goals, Operators, Methods, and Selection rules) combines tasks analysis with second-by-second timing of computer interface actions and cognitive operators based on models of the brain such as the Model Human Processor [21]. 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 descendants allow an analyst to compute how long a typical, error-free operation will take to perform. Because a system designer can use it to create concrete and accurate measurements for task completion times, GOMS 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 [65]; development on GOMS-based analysis software continues to the current day [67, 109]. GOMS-based techniques (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. As such, they provide an effective method for testing representation designs suggested by referential structure analysis.

*Groupware Task Analysis* [138, 139, 142] combines the rigor of task analysis with the thoroughness of ethnography. This combination of methods is used to produce a composite approach which promises to construct a model of interaction which more closely resembles the actual interaction of real-world participants. By combining knowledge elicitation about tasks and responsibilities with actual observation of the

work-practice of participants, this technique endeavors to overcome difficulties with traditional task analysis, such as the inability or unwillingness of participants to accurately describe their tasks and responsibilities. Likewise, by including the structural components of hierarchical task analysis, groupware task analysis provides some additional ability for inference about process and task design.

*Cognitive work analysis* [102] extends the basic framework of task analysis with 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, such as construction of a book-lending system for a public library [143].

### 2.3.3 Workflow Analysis

*Workflow analysis* [86, 119, 16, 137] is an interaction modeling approach which focuses on the construction and alteration of a work product, whether that be material or conceptual. It de-emphasizes considerations such as the specific representation of task information in favor of tracking inputs and outputs to each step in a process. Workflow analysis focuses on the "job", the product of the work that participants perform. It examines the processes that affect that job, the inputs and outputs to each step that transforms the 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. Analysts can then make recommendations about how to redesign the activity to reduce inefficiencies and speed production.

Due to its strong structural element, workflow analysis is well-suited to automation. Technologies such as Petri nets [100] have been used to strengthen the structural component to allow automated analysis [135]. There are a variety of commercial software packages available that allow a company to model workflows with the goal of improving business processes. Some work has been done to construct a universal workflow vocabulary and identify workflow patterns [136] that allow an analyst to easily chunk situations into manageable entities. Dynamic workflow management [74, 68] allows richer variation in task execution parameters, to more closely match the emergent behavior of real-world systems. Workflow-based software systems have also been coupled with predicate logic systems to create simulation environments, which allow experimental exploration of an activity space.

### 2.3.4 Activity Theory

*Activity Theory* [69, 73, 79] examines the social issues in an objective-oriented interaction as an explicit, first-class part of the analysis. By examining the activities performed by participants, and noting how they are affected by interrelations with other participants and with the surrounding social situation it is possible to build a more complete model of work in a cultural context. Activity Theory expands the basic subject-tool-object mediation triangle [144] to include a cultural-historical perspective [31, 37], a perspective which includes worker, task, and community in the scope of analysis. In this view, artifacts mediate the worker's interaction with the task, and procedures (for example, division of labor) mediate the worker's relation to

the community at large.

Bødker [17] 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, help design physical interfaces for shared environments [41]. Examining the use of artifacts in context [11] allowed these researchers to draw conclusions about how system design affects the entire socially distributed activity system. Investigating issues of organizational memory revealed the use of organization-level artifacts for storing business logic in a fashion that can persist beyond the employment of any single employee [12].

In Figure 2.6 I have applied the lens of activity theory to the "burger joint". Here there are a number of actors — cashier, cook, customer — whose interaction is mediated by tools (such as the script for ordering and the ordering slip itself), rules for the interrelation between actors, and an implicit division of labor amongst the participants for achieving a transformation of an object (in this case the hungry customer). The objective, or outcome, of the activity is the serving of the customer. These pieces of the system serve to organize observations by the analyst, and may highlight inherent conflicts within the system.

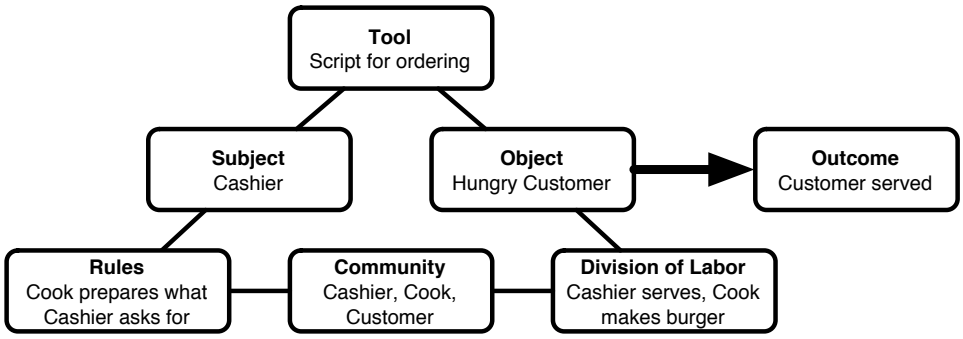


**Figure 2.6: Assessing the burger joint from an Activity Theory perspective.**

## 2.4 Analyzing ethnographic data

Models of interaction must be based on input. There are a variety of options: *a priori* understanding of the interaction held by the designers; design requirements garnered from interviews, suggestions, and surveys from customers; task requirements; and so forth. Within the CHI community, it is generally accepted that data generated by ethnographic analysis of a work domain produces results which are superior to ones based on an *a priori* analysis of the task domain or idealized models of how work ought to be done. However, existing methods for transforming ethnographic analysis into activity design tend either to be very abstract and hard to apply, or very specific and only applicable to a restricted subset of domains.

I have chosen to expand one form of ethnographic analysis, discourse analysis, and couple it with a theoretical model of interaction, distributed cognition, to create a methodology that bridges the gap between analysis and design. By using ethnographic data — namely, the discourse of participants — to populate and provide evidence for the model outlined by distributed cognition, the methods provide analysts with a tool for transforming observation to recommendation.

The discourse of participants is a rich source of ethnographic data. As they perform their tasks, participants must necessarily generate communication to stay coordinated during a task. Over time, participants will generally seek to simplify their interaction by establishing systematic procedures for performing actions and for communicating. Conversation analysis, a social theory that examines the structures inherent in human interaction, posits that this structure, generated to simplify interaction, can be examined to learn about that interaction. As I am seeking to design systems which support interaction, we can use these insights from conversation analysis to work backwards from output (generation of conversation) to generator (model

of interaction).

However, making use of these observations requires a model of interaction. I chose the distributed cognition model of interaction because it examines both internal and external representations of information. Representations that are external, "outside the head" of the participants are therefore available for direct inspection, simplifying analysis. The external nature of these representations allows an analyst to observe the contents of information in these representations directly, and ascertain the nature of its presentation.

### 2.4.1 Discourse Analysis

Discourse analysis [10, 52, 122, 126] is a method of understanding interaction by focusing on the content of the discourse (spoken, textual, and otherwise) that participants create as a part of their interaction. This analytic approach seeks to understand interaction by examining the meaning and purpose of utterances within a discourse, as well as the topics, focus, and conversational centers of a discourse [38, 51, 132].

For this thesis, I made use of theoretical ideas from this tradition, coupled with analytic techniques for examining the minutiae of the discourse, to garner data about information flows within a system. Lockman and Klappholz [82, 83, 140] discussed a way of organizing reference in discourse. This view treats any connection between conversational objects that is necessary for constructing a coherent explanation of the discourse as a form of reference. As I shall show in this thesis, by connecting references to common objects into co-reference chains in this fashion an analyst can determine the important conversational objects, methods of reference, and relations between objects within the discourse.

### The Language/action Perspective

In one area of research within Discourse Analysis, Searle examined the purpose of specific utterances within a conversation [122]. This work posited that the illocutionary purpose behind a utterance is highly context-dependent; a restauranteur saying, "it's cold in here" when standing in a walk-in freezer has a completely different purpose behind their statement than a grandmother sitting in a drafty living room saying the same words. This perspective of codifying the underlying meaning of utterances was useful for determining the topics of conversation within a discourse in referential structure analysis.

This work is closely related to research examining the establishment and transition between topics in discourse. Grosz [52, 51] established a three-part view of discourse: linguistic structure, which addresses the sequence of utterances; intentional structure, which examines the intentions of participants as revealed through conversation; and an attentional structure, which is a dynamic examination of the current focus of the conversation. These three inter-related models can be used to model interaction via examination of the conversation; I make use of concepts from these models in referential structure analysis, where the analyst tracks the topics and attention of participants by examining discourse referents. Winograd and Flores [151, 152] created a perspective for modeling interaction of language as a generator of action. Their theory of language use in interaction states that utterances act as transition steps within a state diagram of interaction.

Concepts from these approaches are used in Figure 2.7 to analyze the "burger joint". This figure shows how the conversation from Figure 2.2 can be viewed as a set of transitions between states in the ordering process. From an idle state, the cashier's opening statement transitions the interaction to what can be termed the

"ordering" state. The customer presents an order (this could be considered a sub-state transition within the ordering state); the cashier requests clarification, which pushes the conversation into a "clarification" state, and once the clarification is received and understood the conversation returns to the ordering state. Finally, the cashier explicitly signals the transition to the "waiting for fulfillment" state with her final utterance. This perspective is useful for modeling an ongoing interaction; I make use of it in referential structure analysis to establish the pattern of information use over time.

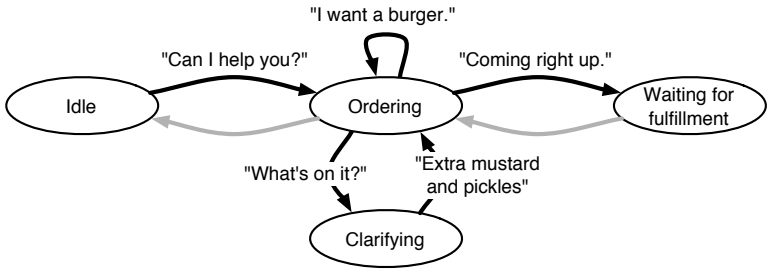


**Figure 2.7: A language/action perspective on discourse in the burger joint.**

Discourse analysis has also been used to model interaction and provide information used to feed into an intelligent agent. By tracking identifiable features of discourse, such as speaker choice, breakdowns in coordination can be identified. Work by Goodman et al. has used this technique to enable a conversational agent to highlight departures from a standard model of interaction, and if necessary to intervene to correct these departures [48]. Another potential level of discourse analysis is to track topic or word use within a discourse so as to better understand the intent of speakers. This level of detail in analysis of participant discourse has been used to gain insight into the meaning of a conversation, for purposes such as, e.g., language understanding by intelligent agents [104, 125]. This sort of referential chaining work is also the basis for referential structure analysis.

### 2.4.2 Conversation Analysis

*Conversation analysis* [34, 112, 113, 115] is an alternate form of ethnographic analysis that examines the social structure that emerges during interaction. It investigates the procedural infrastructure of interaction by examining structures that emerge in conversation: features such as 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.

Nuances such as word choice, sentence structure, the turn-taking of the participants, and other such features provide an analyst with insight into the structure participants create to organize their behavior, and the problems that the participants may be having in achieving their goals. As Sacks, Schegloff, and Jefferson [108] indicate, conversation is a party-administered, locally-managed behavior. That is, within conversation, a set of local conventions emerge which participants use to determine next speaker, transition between speakers, negotiate the ending of a conversation, and for other such recurring problems of coordination.

Conversation analysis has been put to use examining a wide variety of situations. Early work looked at telephone conversations [116]; more recently, it has also been applied to a variety of other domains, such as examining the conversations of doctors with their psychotic patients [85]. Characteristics such as turn-taking, speaker choice and speech act type have been used to identify breakdowns in coordination by highlighting departures from a standard model of interaction [48]. Examination of the duration and type of conversational utterances has also been used as a way of determining the impact of alternate representations on conversation [71].

In the 'burger joint' example from Figure 2.1, conversation analysis provides a way to understand the interaction through the conversation between cashier and customer. A possible conversation for step one of the interaction appears in Figure 2.8. Note that as this is not an authentic transcription, its utility for revealing the true utility of conversation analysis is somewhat limited; however, it will serve for explanatory purposes.

| | | |
|---|---|---|
| 1. Cashier: | Can I take your order? | *self-selects speaker* |
| 2. Customer: | Uh, yeah, I'll have a burger. | *response* |
| 3. Cashier: | What's on it? | *forces clarification* |
| 4. Customer: | Umm... extra mustard and pickles. | *clarifies* |
| 5. Cashier: | Coming right up. | *ends topic* |

**Figure 2.8: Conversation analysis for Step 1 of the burger example.**

This sample reveals a lot about the server/customer relationship that is an integral part of the conversation. The cashier self-selects for first conversational turn with a formulaic greeting, which sets the frame for the interaction and prompts the customer to reply [114]. The customer formulates an under-specified request; to avoid future repair, the cashier (following a script honed through years of cultural history) asks for clarification before continuing. Finally, after the clarification is received, the cashier explicitly ends the conversational unit, allowing transition to the next phase of interaction.

Conversation analysis would note the strong structure present in the conversation and conclude that this is a well-established interaction paradigm. From this perspective it is clear that the cashier has a goal in mind in the conversation, and uses speaker selection, prompting, and a known script to simplify this task. This perspective might lead a system designer to create software that supports this interaction by mirroring the strong customer/server relation; modern ordering systems do just this, by encoding the script and turn-taking into automated systems such as touch-screen

ordering.

## 2.5 Predicting system impact

The third piece necessary for generating redesign recommendations is the ability to predict the impact of alterations to the interaction system. Many researchers have studied the impact of representation on human understanding and interaction. The work in this thesis builds on research of Hutchins (op. cit.), Zhang and Norman [157], Chuah et al. [25], Wickens [150], Schmidt [117, 118], Suthers [35, 130, 129] and many others. These observational studies of representation use provide crucial information for a designer, and are necessary background when designing a new representation system. This section will discuss related work regarding the impact of introducing new representational media can have on three levels: personal, interpersonal, and societal. Understanding this impact is an important part of developing a new system, and as a result, this body of literature provides an important complement to the analysis and design methods outlined in this thesis.

There is a rich and growing literature addressing the moment-by-moment impact of introducing representations into a task. This work has primarily arisen from two traditions: ergonomics and psychology. These methods are therefore strongly observation-based, requiring formal experimentation in a controlled setting to accurately measure the impact on user performance of a particular representation; they are therefore best suited for use as a part of a larger methodology for redesign. This use will be demonstrated in Chapter 6.

Zhang [155] examined a simple problem-solving task – the game of tic-tac-toe – to establish the impact of presenting information in different ways. By altering the visible effects of actions, they were able to reduce the user work required to envision

future possibilities and trivialize planning in a trivial domain. Work by Chuah et al. [25] confirmed these results in more complex domains.

Literature in critical applications like Air Traffic Control have long examined the precise impact of information presentation. Handbooks [150] provide specific guidelines about what sort of representations are suitable for certain types of information within a limited domain. These efforts, while valuable, represent another instantiation of the prescriptive rulesets described above, and share their problems. However, as with those rulesets, this research can provide invaluable detail as a part of a larger method.

Research into affective computing [101] has, among other research threads, examined whether presenting information in an emotional fashion can alter a human's engagement. Studies with 'cute' robots [20] and emotion-expressing interface agents demonstrate that representations of such information can have legitimate uses.

Work by Suthers and collaborators [130, 129, 131] has examined how providing different representations for a task can affect how students learn information. Ainsworth [2, 141] has examined the effect of providing multiple, redundant representations to learners, demonstrating the improvements to be had by presenting information in alternate forms. As shown in this work, multiple external representations can be used to explore a task and its attendant information in a fashion unavailable to users limited to simple representations such as a chat window. The research in this thesis has itself been applied to examine this problem in the Group Homework Tool experiments [56, 78].

Schmidt, in conjunction with a number of collaborators [117, 118], has examined the use of coordinative artifacts in a work-practice situation. For example, a long-term study showed how architects within a firm made use of a large number of auxiliary representations for task information to structure their activities. Work by Kraut and

collaborators [47, 71] has examined how providing alternate representations can alter the dialogue of participants in a joint activity. These psychologically-based studies have collected good data about the reductions in user effort caused by introduction of such alternative representational media as shared views and graphical displays.

### 2.5.1 Impact on interpersonal relations

A subset of this work examines the impact of representational media on an interaction from a psychological, sociological, or anthropological perspective. The set of communicative methods available to participants in an online community can dramatically affect not only their perceptions of the task, but of one another as human beings and of their engagement as part of a larger online community or as members of society. This impact must be accounted for when redesigning the representation system of a groupware system. Although this thesis work does not directly utilize this research, it is important to keep such matters in mind during the design process.

One thread of research in this field examines how the different quality of mediated interaction affects interpersonal relationships. Research on this topic dates back at least to introduction of the telephone; the social psychology of long-distance relationships and telecommunications in general has been examined for decades [124]. Some analysts have examined the way that online communication, simultaneously highly personal and strangely impersonal, impacts how relationships arise and evolve [145, 146, 147]. This research indicates that the paucity of available context given the restricted representational system available to participants means that participants end up augmenting this communication with stereotypical attribution of characteristics [64].

Research over the last decade has also been concerned with the social impact of the rapid and accelerating adoption of internet-mediated technologies such as

email, instant messaging, and persistent online communities such as online forums and massively-multiplayer online games.

Email has changed how business gets done, and how family and friends stay in touch. Research has examined the impact of email on business [87, 148]. Online chat using technologies such as IRC, Instant Messenger, or ICQ has altered how friends stay in peripheral context. Studies of the interaction of participants in a closed work-group via instant messaging revealed problems of topic drift and awareness [90]. Multi-user variants have been proposed which address some of these concerns [19].

Persistent online communities such as those centered around mailing lists or online fora are excellent domains for applying our methods. Examining the discourse provides a good way to measure how the introduction of a new representation — say, a mailing list — changes the way a large group of people can interact. Work has been done on how to create persistent systems that users will adopt and find useful [55]. Other researchers have studied how these social technologies can paradoxically reduce social involvement by disengaging participants from the social process [72].

A prominent example of persistent communities are MUDs (alternately, multi-user dungeons, multi-user domains, and a number of other suggested expansions), originally created in 1979 by Roy Trubshaw and Richard Bartle [14]. These early computer programs allowed users of remotely located systems to interact via a chat-like interface with an embedded rule system. Two simple programmer-imposed artifacts — rules for interacting with other users and objects in the system, and the adoption of a false persona — moved the interaction in MUDs away from the simple 'chat room', party-line atmosphere of previous technology such as IRC. Anthropological investigation of MUDs revealed strong differences in how users used and reacted to the systems. Two landmark papers examined the engagement of users in an online character and the way in which a community responded to virtual violence [33], and

the differing goals of four identified types ("suits") of players [15]. These studies showed the impact that providing an extra layer of representation had on the way that users interacted.

Massively-multiuser online systems, such as MMORPGs, are the evolutionary successors to MUDs. These persistent online communities involve thousands of people, though a typical interaction between participants usually occurs on the scale of dozens. Studies of MMORPGs have examined the increasingly strong ties between these virtual economies and real-world economies, as well as conducting long-term studies about the economies and cultural impact of these online worlds [23, 24]. These sociological structures are, at their essence, secondary structure created by users in their societal discourse to attempt to structure their behavior, and as such are open to analysis using some of the same methods presented in this thesis (notably, recurrence analysis).

## 2.6 Conclusions

There are many frameworks for analyzing and building models of interaction. There are many methods for guiding redesign. However, there are few platforms which provide a connection between the two. The analytic platforms are generally quite useful for their descriptive power, enabling an analyst to understand important facets of the interaction they are observing. They also can be quite useful for their rhetorical power, giving a researcher better ability to describe what they see and frame it in terms other researchers can understand. However, in general, they are lacking in inferential power; most do not provide the analyst with the ability to foresee the effect of changes to the interaction. Also, many have little or no applicability to redesign; while they are able to describe an interaction, and potentially highlight

or detail problem areas within the coordination, these theoretical frameworks are generally unable to provide specific directions for redesign which will improve these difficulties.

As shall be shown in later chapters, the methods described in this thesis give the analyst tools with which to make specific design recommendations based on observation of participant discourse. The methods also give ways to predict the impact of the addition of representations to an interaction. However, it is important to emphasize that the methods presented are meant to be used in conjunction with, rather than as a replacement for, other analytic and design methodologies. Based on a strong foundation of principles from distributed cognition, and utilizing techniques from ethnomethodology and discourse analysis, these methods contribute a new tool which can help system analysts and designers produce concrete, quantitative conclusions about the information representation needs of an ongoing interaction.

# Chapter 3

# Redesigning Groupware

## 3.1 Chronology

The GROUP lab is concerned with building a model of interaction that can be used to create successful software systems. To address these issues, our research group constructed a component-based groupware toolkit, Thyme, to allow system designers to quickly and easily generate transcribable groupware systems. Systems generated using Thyme will automatically generated logs of interaction, based on the actions of participants [75, 76, 77]. This interaction data can then be summarized and replayed using Sage, a data playback program, and analyzed with a variety of tools for performing both qualitative and quantitative analyses of the data.

The primary system created to study issues in same-time/different-place coordination is the VesselWorld groupware system. The VesselWorld (VW) project was intended to provide a multi-user platform where the interaction between participants could be recorded and observed. The GROUP lab began construction of VesselWorld in 1997, with various versions released over the next four years. Primary construction was led by Seth Landsman, with significant additional development from the author

and Josh Introne. Over its lifetime, VesselWorld was used in five main experiments generating a total of about 300 hours of interaction data, four of which will be explored in this chapter. (The fifth is the work of Joshua Introne; details can be found in [61]) These experiments are summarized in Table 3.1.

| | Groups × Hours of Data | Results |
|---|---|---|
| VW1 | $2 \times 10$, $1 \times 60$ | Debugged the VW tool and domain |
| VW2 | $6 \times 20$ | Produced data for recurrence analysis |
| VW3 | $3 \times 12$ VW-NO-CR<br>$3 \times 12$ VW-CR | Recurrence analysis showed improvements in CR version |
| VVW | $2 \times 12$ VVW-NO-CR<br>$2 \times 12$ VVW-CR | Compared with VW3; demonstrated impact of representation system |

**Table 3.1: Overview of VesselWorld experiments.**

The first experiment conducted using the VesselWorld system (VW1) aided in debugging the VesselWorld experimental platform, and can be considered part of the development process. These experiments also provided experience gathering and analyzing interaction data.

The next experiment (VW2), conducted in conjunction with the Naval Undersea Warfare Center and in collaboration with Susan Kirschenbaum, produced evidence on the use of the base VesselWorld system, as well as additional experience in gathering ethnographic data. This data was analyzed using Recurrence Analysis, an analysis technique which will be discussed below. Conclusions from this analysis led to creation of a new system (VW-CR), which included three new coordinating representations.

VW3, the third experiment, was a larger-scale experiment which compared a basic version (VW-NO-CR) with a version (VW-CR) that included the three new coordinating representations. This experiment showed that the coordinating representations improved the speed and error rate of participants; these quantitative results will be explored in this chapter. The experiment also raised some questions about system

design that led to the development of Referential Structure Analysis, which was able to successfully explain these previously inexplicable results.

Finally, the 'Visible' VesselWorld (VVW) experiment demonstrated the ability of referential structure analysis to predict the impact of representation system changes. This experiment will be used in the next chapter to demonstrate the ability of the method to explain the impact of representational changes on how participants exchange information.

## 3.2 The VesselWorld testbed

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.

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 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 interface of the basic VesselWorld system is shown in Figure 3.1. 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.

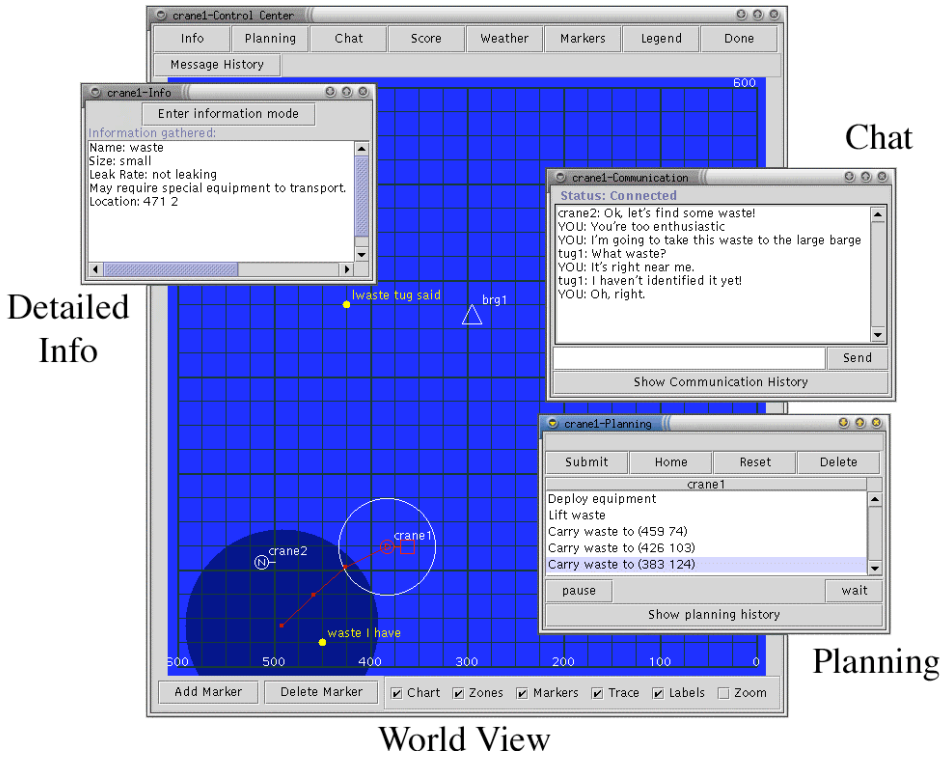


**Figure 3.1: The VesselWorld interface.**

Users navigate the harbor and formulate short-term plans for movement and clearing waste via the Plan Window. Plans are created by clicking within the main window; for example, clicking on the icon for a waste barrel will (in the case of the cranes) create a plan step to lift that barrel; clicking on a barge will either load a carried

waste barrel onto that barge, or attempt to unload a waste barrel from the barge (if any). Once the user generates a plan, he can submit the first step of that plan to the system via the Submit button in the Plan window. The user must then wait for the other two users to submit their plans; the system then calculates the results of the three submitted plans, alters the state of the world accordingly, and returns control to the users. If the users have made an error — for example, if a crane attempts to lift a waste barrel without deploying the correct equipment, it will cause the waste to begin to start leaking. Leaks require the tug to seal up that waste or risk further contamination of the harbor.

For each experiment, the three participants are trained in use of the system and then asked to solve a series of waste retrieval scenarios. Each session generally requires somewhere between thirty minutes and 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 in all groups.

A complete manual for the VesselWorld system can be found in Appendix B.

### 3.2.1 Collecting usage data in VesselWorld

To investigate the interaction procedures of participants we needed to understand how the work that participants were doing correlated to their interaction. All that is visible in the system is the interaction of the user with the interface, which because of the experimental setup included all conversation and other coordination with other users. As in any joint activity, participants were engaged in many simultaneous levels of work: at the most immediate level, work performed to complete the actual task of clearing the harbor; on top of this, work necessary to be able to perform that

work, in this case interface actions necessary to manage the system, such as window rearrangement and scrolling; and finally the work needed to maintain coordination with other agents. Our research focused on the interaction data; that is, the team work that participants performed to stay coordinated with each other. By examining this interaction data we were able to design an improved version of VesselWorld, and explore the process of analyzing and designing groupware systems tailored to a particular interaction.

As users operate the VesselWorld interface, they automatically generate both syntactic level interface events ('user typed control-w', 'user clicked mouse at 500,350') and semantic level system events ('crane1 closed info window', 'crane2 requested info for the point 500,350'). Both types of events are logged to disk automatically by the system for later inspection. By collecting semantic level events, VesselWorld allows the analyst to draw aggregate conclusions about the actions of participants. For example, it is straightforward to determine the percentage of waste barrels that participants got detailed information on, or to see how many markers users generated.

Initially these logs were inspected by hand or using basic text processing scripts. For example, a Perl script was written which extracted each line of chat and each window operation (moving, opening, closing), and presented them in a summarized, time-stamped format. This level of inspection was sufficient for analysts to investigate basic issues users were having with the system, and approximated the level of detail a transcription would provide. However, doing this proved cumbersome, and after some false starts, we created a playback application for VesselWorld, the "VCR", which allowed a level of detail similar to but richer than that available via videotape. Shown in Figure 3.2, the VCR allows an analyst to step forward through the log file in a graphical fashion. The controls are set up like a traditional VCR, with play, fast-forward, and so forth. Additionally, the VCR allows searching through the log

file for events of specific types, which is useful for refining study of the interaction. For example, the analyst can set the VCR to step forward to each new submission of plans.

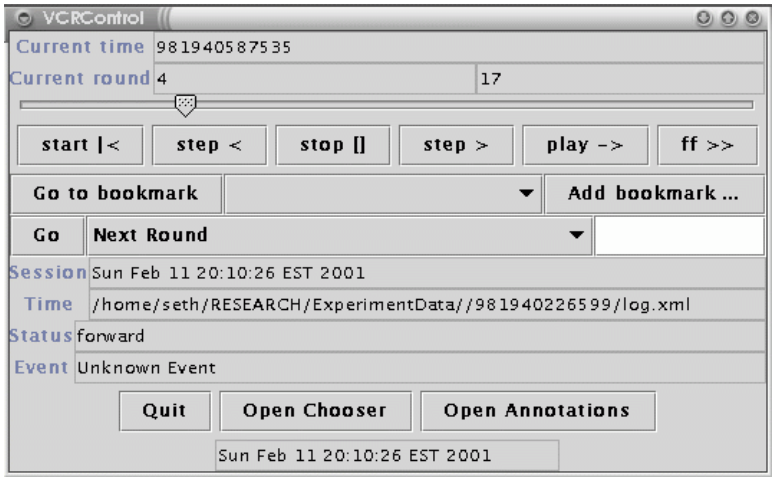


**Figure 3.2: The VesselWorld VCR, allowing replay of log files.**

The VCR interface also shows information about the composition of the log file: date and time of log creation, length in minutes, length in events, and current position within the file. A bookmarking function allows an analyst to return to places of interest within the log file. The VCR was used to analyze data from the experiments, allowing the analyst to get an overview of what all three users were doing at any moment in time. This functionality was put to good use during the application of recurrence analysis to VesselWorld usage data.

## 3.3 Recurrence analysis

Recurrence analysis is an ethnographic analysis technique developed by the GROUP lab which laid the groundwork for referential structure analysis [5, 6, 7]. It provided the backdrop against which referential structure analysis was created, and is included here for its historical significance. It is based on extensive observation of interaction

data. Recurrence analysis posits the existence of three indicators which suggest that incorporating an alternate representation might reduce participant effort:

1. recurrent conversation about activity
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. In the first case, participants find themselves discussing the same coordination situation over and over again. As we have previously discussed, participants generally only discuss an activity when they find it troublesome or exceptional. Obviously, a performance improvement can be achieved by supporting these recurring yet exceptional situations in a manner that smoothes the flow of work.

In the second case, the participants are committing the same sort of error repeatedly. This could be for a variety of reasons, from interaction design issues to user error to a misunderstood system metaphor. But whatever the root cause, it is worthy of further examination.

In the third case, the participants themselves have identified a difficult interaction situation, and have created a framework in their discourse to ameliorate the problem. Like an oyster making a pearl to surround an irritating particle, the participants have constructed some sort of secondary structure in the dialog in an attempt to smooth over the rough area of coordination. This is strong evidence for a need to improve the flow of work at this point.

This section will examine each of these indicators in order, and illustrate via examples how their occurrence reflects incompatibility between the available representation and the necessary coordination of the participants. The examples are taken from the

VW3 VesselWorld experiment.

### 3.3.1 Recurrent patterns of coordination

In situations characterized by the first indicator, participants find themselves repeatedly discussing a particular sort of interaction. The fact that participants must chat about their situation to accomplish their task indicates that the coordination tools (other than chat) they have are insufficient alone to organize their task. They must talk in order to coordinate. In most cases, chat is an inefficient method of coordinating action: though the flexibility of natural language means that it is usable in almost all circumstances, unstructured discourse acts are generally more error-prone and require more cognitive effort to produce and understand than communication via purpose-built conversational procedures or external artifacts.

One recurring situation in the VesselWorld data involved confusion about who had submitted their proposed plan to the system. Because all three users need to formulate and then submit a plan to the system to continue to the next step of execution, VesselWorld requires users who had submitted a plan to wait until the other users had also submitted. In this wait state, users can communicate and manipulate their interface freely, but cannot formulate new plans. Information about who had and had not submitted their plans is not easily available, and depending on how long it took to finish gathering information, making decisions, and formulate a plan required, a particular user might be delayed in submitting a plan long enough for the other users to get anxious. In most groups there was dialog similar to the three separate segments shown in Figure 3.3; these segments are pulled from different groups, but show similar frustration at the lack of feedback from other users regarding plan submission.

Differences in computer proficiency increased the severity of this problem; more proficient users were simply faster at clicking around and generating plans, but then

crane2: did you submit?

crane2: hurry it up and submit
crane2: before i die of old age

crane2: still waiting for someone to submit...
crane1: ive been submitted for quite a while

**Figure 3.3: Many groups had difficulty coordinating plan submission.**

had to wait impatiently while the less proficient users did the same. This further exacerbated the problem because the slower user had to take additional actions (i.e., chat) to broadcast his state to assuage the faster users.

A possible solution would be to provide an alternate mechanism for communicating the plan state (submitted/not submitted) of each user, either manually or automatically. For example, the plan window might have a yellow light/green light indicator that shows other users when a fellow user has submitted a plan. Even if the user is required to manually operate this signaling device, by crafting it to match the type of information being sent — in this case a two-valued toggle button for a boolean value — a designer can reduce the effort to communicate this information. Additionally, by selecting a representation that matches user expectations about such information — yellow meaning 'wait' (or 'slow') and green meaning 'go' in American culture — the representation would reduce cognitive effort to comprehend the information.

### 3.3.2 Recurrent errors in coordination

The second indicator, recurrent errors, points to areas where participants are having acute difficulty in maintaining coordination. These are situations where participants commit (or nearly commit, for more complex domains) the same error repeatedly. 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.

Analysis revealed 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. Discrepancies 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 3.4.

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 3.4: 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 the discrepancy between crane1's private representation and tug1's private representation. 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. As will be demonstrated in later sections, providing a shared, authoritative representation for such information greatly reduced incidence of this type of error.

### 3.3.3 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 such structure evolved, 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. The next few sections present three forms of secondary structure created by participants in the VesselWorld discourse, ranging from the simple to the complex.

#### Adjacency pairs

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 3.5. 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 [116] appeared.

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

**Figure 3.5: 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.

### Waste jargon

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 barrels could be discovered by any user, but each barrel required a particular set of actions involving one, two, or all three users

to handle successfully. For this reason, successful 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.

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 it was imperative for participants to share information about newly-discovered wastes. Such reporting became stylized in all groups as they developed conventions to simplify the task. A typical example is shown in Figure 3.6.

tug1: Weve got two wastes in my range: medium Dredge !leak at 373 301
tug1: Net medium !leaking 407 376
crane2: I have 3 M? !l@269,247—m? !l 296,305—m? !l 373,301
⋮
crane2: ohh was my compression of the waste info understandable
crane1: yeah
crane2: M? !l@546,295
crane1: large !l 389 781
tug1: 296 305 Net !l med

**Figure 3.6: Secondary structure created to help waste reporting.**

Most of the communication users generate during the early part of the session consists of waste reports using jargon similar to that seen in this figure; for example, "`m? !l 373,301`" indicated a medium-sized waste barrel, equipment unknown, not leaking, at the location `(373,301)`. 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. Conventions were unsuccessfully communicated, or were not followed exactly; complex formats required extra work on the part of the author, which was a cost some

participants chose not to invest. In many cases, the agreed-upon representation contained more information than was needed, and so was not used for every barrel of waste. These issues created problems for the participants, leading to new sorts of errors.

### Marker check

Another example of secondary structure involved the Marker Check (shown in Figure 3.7), 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 3.7: Marker check reveals a discrepancy in the users' private representations.**

However, producing and using this structure proved quite time-consuming for participants, 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. Later, this thesis will present experimental evidence that introducing well-chosen alternative representations significantly improves performance.

## 3.4 Redesigning VesselWorld: VW2

Recurrence analysis was used to perform an investigative analysis of the participant dialog from a VesselWorld pilot study, in conjunction with other knowledge elicitation techniques. Subjects were interviewed after they used the system, and asked to talk about issues with the interface and system execution. This was helpful for guiding our focus in the primary phase of the investigation, during which we looked at the actual interaction as recorded in detailed log files. Using the built-in playback features of THYME, we were able to replay the pilot sessions and examine the step-by-step interactions of the users as they performed their tasks. Our goals were to understand the problems that participants were having with the interaction, and to determine ways to improve the system's efficiency.

Using recurrence analysis we analyzed the VW2 data. This included about 60 hours of usable interaction data. The analysis revealed three notable areas of difficulty:

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 significant level of coordination work required for participants to complete their task. These areas also were the source of the majority of errors. To address these difficulties we redesigned the representation system of VesselWorld. In conjunction with user interviews and exploration of the log files, we used the analysis of the usage data to inform design of new representations for VesselWorld users. In the end, we produced one coordinating representation for each of the problem areas listed above, introducing the three new coordinating representations to create the VW-CR system.

The redesigned version included three new representations created to ameliorate the coordination problems participants were encountering: the Object List window, the Shared Planning window, and the Strategic Planning window. We will examine each of these new representations in turn.

### 3.4.1 Shared Planning

The Shared Planning coordinating representation was created to provide awareness of other users' plans. During planning in the base system, users had to chat constantly to maintain awareness of the planned actions of other users. This led to errors in both action choice — for example, one user might plan to load a waste barrel on a barge just as another user was moving that barge out of range — and in action timing, most noticeable in joint lifts and loads. By modifying the Planning Window to reveal the plans of all users, we were able to simply and effectively reduce both types of problems. The Shared Planning window, shown in Figure 3.8, was a natural extension of the basic system.

Use of the Shared Planning window was identical to use of the normal Planning Window, as detailed above, except that the user could also see the plans and currently selected plan step of other users.

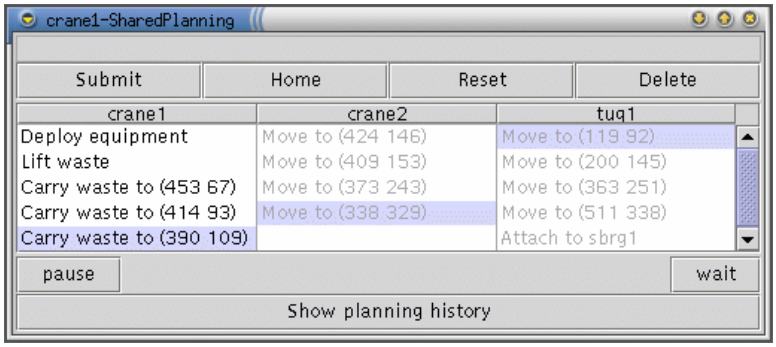


**Figure 3.8: The Shared Planning Window.**

### 3.4.2 The Object List

The Object List is shown in Figure 3.9. 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 [84] we added a free-form notes area for users to note information about a waste that did not fit into the structure presented.

### 3.4.3 Strategy

The final representation we created was the Strategic Planning Window, or Strategy Window for short. This representation was intended to reduce collaborative effort in

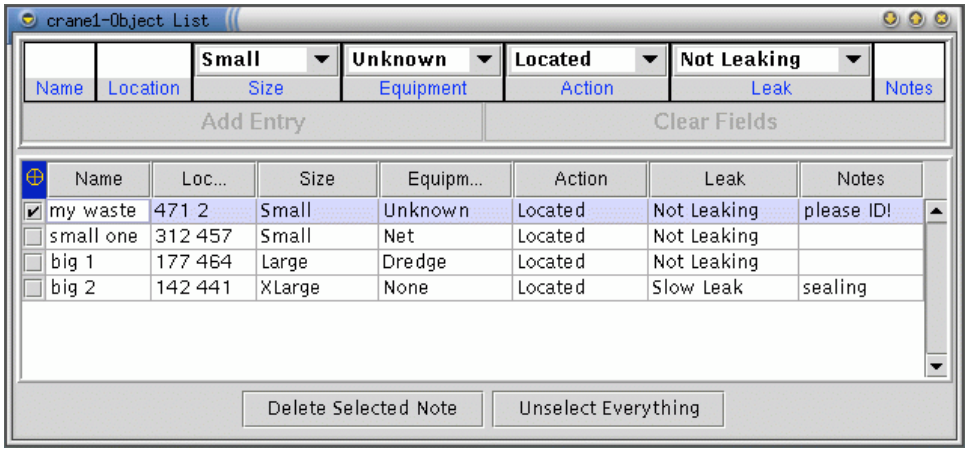


**Figure 3.9: The Object List.**

longer-term planning. Users spent a fair amount of time discussing which barges and barrels they were planning on handling, long before concrete plans for these actions were generated in the Planning window. The Strategy Window, shown in Figure 3.10, was designed to aid users in maintaining awareness of the actions of other users and publish their own actions in a rapid and structured fashion. A structured procedure for adding information to the Strategy window was chosen both to speed data entry and to reduce errors due to vocabulary problems.

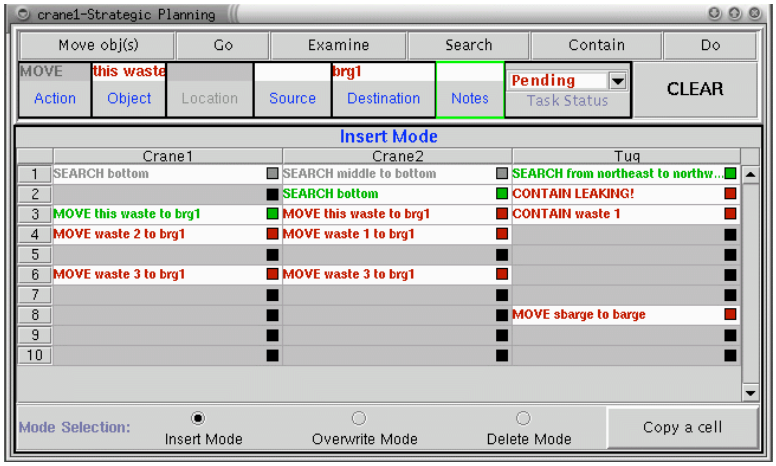


**Figure 3.10: The Strategic Planning window.**

Users could all edit the cells of the central table. Users could add items by

formulating a new Strategy item in the form at the top of the window, similar to the Object List. Users could label these items with a status, which was displayed as a color within each table cell: Pending, in red, for tasks not yet underway; Current, in green, for tasks currently being executed; and Done, in gray, for tasks which have been completed. Users could construct Strategy items by selecting an Action (verb: one of Move Object, Go, Examine, Search, Contain, and Do), and then filling in various auxiliary properties: Object of the action, an optional Location, Source, and Destination, and a free-form Notes field. Items could then be placed anywhere within the free-form agenda shown in the center. Users could also move items around and delete or overwrite items on their agenda.

## 3.5 Testing the redesign: VW3

We conducted a single-variable experiment (VW3) 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 NO-CR groups), 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.

### 3.5.1 Anticipated results

Because of the relatively small population size, 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.

The experiment produced a number of major results, summarized in Figure 3.2. The performance of the VW-CR groups was significantly better than VW-NO-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 previously identified — close coordination, domain object reference — was notably improved, with errors due to miscommunication of object information significantly reduced.

| Measure | VW-NO-CR groups | VW-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 (UI 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% ($p < 0.35$) |

**Table 3.2: Comparison of VW-CR and VW-NO-CR groups.**

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% — it is clear 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 was expected to drop significantly was the number of rounds of activity required to perform the task. However, while the reduction in this measure was promising and it would appear to be a valid result, it was found to be not statistically significant and can only be used for qualitative analysis.

### 3.5.2 Unexpected results

Although the VW3 experiment showed that overall the new coordinating representations improved performance and reduced the rate of errors, there were many problems with the new system. One 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." Users evidently found no need to use these columns, but the recurrence analysis methodology could not determine why, and failed to predict this use.

Most disconcertingly, the Strategy Window was not used at all once training in it was complete. Subjects refused to use the window at all, with one exception; and even in that group, disagreement about using the window caused serious contention, with two users flat-out refusing to utilize the representations. Subjects gave some insight into why in their exit surveys: "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 Strategy Window 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 "Ac-

tion" columns of the Object List, and the Strategy window itself, went unused by all groups while other columns and representations, based on equally strong experimental evidence, were used constantly. We were unable to anticipate these failures of design using the existing analysis methodology. Because of this, I saw the need for a more detailed form of analysis, leading to the creation of referential structure analysis.

### 3.5.3 Summary

The VW3 experiment resulted in a few major observations. First, it demonstrated that altering the representation system could dramatically improve a group's performance in ways that could not be explained by other means. This provided definitive proof that coordinating representations can in and of themselves improve performance simply by reducing the cognitive effort required of users to stay coordinated.

Second, it demonstrated that recurrence analysis could be used during the redesign process to help isolate specific problems areas in the coordination. Recurrence analysis provides a framework for observation of interaction, giving an analyst specific things to pay attention to during review of usage data. These indicators in turn point at underlying problems in the interaction that alterations of the representation system can ameliorate.

Finally, the VW3 experiment showed us that recurrence analysis alone could not explain the results of the experiment. There were a number of unexpected results, including the rejection of one of the coordinating representations, which recurrence analysis had failed to foresee and could not adequately explain. These led to further investigation of the articulation work of participants and to development of referential structure analysis.

# Chapter 4

# Referential Structure Analysis

This chapter will explain *referential structure analysis*, our method for using the references participants generate to understand information flows within a system. The chapter will begin with a survey of the theoretical background for the method, demonstrate use of the method with a number of small examples, and discuss some of the complications that arise in its application. Later chapters will demonstrate its use on larger bodies of real-world data, and show how it can be used to drive redesign of a representation system.

## 4.1 Method

### 4.1.1 Extracting referential structure

Referential structure analysis is an analysis method that tracks the referents users refer to via a thorough examination of the discourse. From this, the analyst can examine what information is communicated, and in what fashion. The method focuses on identifying the information that users share, and on following its subsequent use within the system, to build a model of the information within an interaction.

The analyst examines the discourse 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. A referent, for these purposes, represents a simple conversational item that the users refer to. Examples in the VesselWorld domain include a barrel of toxic waste; a plan to clear a barrel of waste; a realignment of discrepant personal representations; or a conventional procedure for handling a particular situation.

An example of this analysis is shown in Figure 4.1, an analysis of the conversation from the "burger joint" example. Here, we have identified a handful of the referents that are important in this discourse: the customer's order, the burger itself, and details about the order (in this case, the condiments). On line 1, the cashier's utterance contains an explicit reference to the order, and so we tag a new referent (REF–1A). There are other other available references ("I", "your"), but based on the goal of this analysis the order is the only one tagged.

| | | | |
|---|---|---|---|
| 1. | Cashier: | Can I take your order? | [REF–1A order: burger order] |
| 2. | Customer: | Uh, yeah, I'll have a burger. | REF–1A<br>[REF–2A food: burger] |
| 3. | Cashier: | What's on it? | [REF–3A detail: stuff on REF–2A]<br>REF–2A |
| 4. | Customer: | Extra mustard and pickles? | REF–3A<br>[REF–4A condiment: extra mustard]<br>[REF–4B condiment: pickles] |
| 5. | Cashier: | Coming right up. | REF–1A<br>REF–2A |

**Figure 4.1: Referential structure analysis the burger joint.**

On line two, the customer mentions a new item — a burger — which gets its own refererent (REF–2A). This also contains a contextual reference (again used in the sense of [82]) to the order itself — the utterance provides expansion on the order —

but the burger appears as a new item in its own right, and so we note the both the reference to REF–1A on this line and the new referent.

This process continues on line three; the cashier refers back to the burger itself, and starts a new referent which the analyst tags as (for lack of a better name) "detail" – specifics about the order. On line four the customer continues this referent, and creates two new ones (pickles, and mustard). Depending on the level of detail desired in the analysis, the analyst might choose to ignore these two referents; in this case we tag both for illustrative purposes.

Finally, on line five, the cashier makes a somewhat ambiguous reference; it is entirely possible to connect this reference ("[It is] coming right up") to either REF–1A (the order) or REF–2A (the burger). To account for this ambiguity the analyst records it as a potential reference to both referents, the safest approach. If later inspection – including further discourse – indicates that this interpretation is incorrect it will be easy enough to track down later.

Effectively, the analyst is building coreference chains [134]. As in Lockman and Klappholz [82], every referential object in the discourse can be tracked by examining the referential structure. Relations between references are noted by an analyst, and each reference is grounded to a specific referent. Out of this process a chain of related words is formed which all point to a fundamental underlying concept. Although automation of this process is an open research question currently under investigation, automatic extraction of coreference chains is outside of the scope of this thesis work; discussion of this possibility appears in Chapter 8.

### 4.1.2 Referent types and tokens

To allow investigation of the differences between various sorts of information, the analyst should assign each referent a type. These types are a combination of domain-

specific types — in the above example, the analyst has identified types like "order", "food", and "detail" — 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 referents where users attempted to fix mismatches in common ground, correct errors in the interaction, or disambiguate misunderstandings.

Given a set of collaborative tasks, the hypothesis embedded in the method is that referent 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 referents identified tell the analyst the sort of referents that participants discuss. The specific instances of a referent, 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 referent 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 referent types. As analysis progresses the analyst will generally have to iteratively reassess the choice of referent types until an acceptable set of types is derived for the specific goals of the analysis.

Referent typing can also be supplemented by subtypes or aspects. For example, the waste referents 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 referent types is at the discretion of the analyst. However, identifying referent types is not just an idle exercise. As will be shown, the process of identifying an appropriate set of referent types provides significant insight into the

sorts of information that participants exchange.

### 4.1.3 Sample analyses

To further demonstrate use of referential structure analysis, let us follow through an example taken from a non-CR VesselWorld session. The excerpt begins a few lines into the discourse, after pleasantries have been exchanged, and the users have begun to report waste locations using the jargon they have established in previous sessions.

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

**Figure 4.2: Applying referential structure analysis.**

In Figure 4.2, 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 referent for this waste, the analyst generates a unique name based on the current line of discourse (REF–7A), puts the referent in square brackets to indicate that this is a new or modified reference, denotes the type of the referent as "waste", and lists all information available about that waste.

On line 8, crane1 simultaneously reports on a waste nearby. The analyst again creates a referent 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 referent 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 a referent of type "repair", and makes sure to mention the referents 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 Ref–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 Ref–7a (again using square brackets, this time to indicate that the contents of the referent are being modified), and chooses the earlier of the two names for the waste (Ref–7a and Ref–8a) to disambiguate further references to the waste.

In this brief example, analysis has identified two types of referents: waste referents and repair referents. There are three referents: Ref–7a, a waste referent; Ref–8a, another waste referent; and Ref–9a, a repair referent. Likewise, the analysis has established values for certain quantitative properties of these referents (though, due to the artificially short transcript length, these are somewhat misleading), summarized

in Table 4.1.

| Referent | Type | Lifetime of Relevance | Number of References | Conversational Density |
|---|---|---|---|---|
| REF–7A | waste | 5 | 3 | 60% |
| REF–8A | waste | 2 | 2 | 100% |
| REF–9A | repair | 2 | 2 | 100% |

**Table 4.1: Results from the analysis in Figure 4.2.**

This table shows the values resulting from computing three basic measures of information use: lifetime of relevance, defined as the number of utterances between first and last reference to the referent (inclusive); number of references to the referent; and conversational density, which is the ratio of these two numbers. For this limited example, these numbers are not particularly insightful; however, in a transcript for a full session, these numbers reveal the way in which users generally use information of particular type. By averaging together these measures for all referents of a particular type, and then comparing the results, the analyst can see which information types last longest, which are referred to the most, and so forth. As will be shown, these numbers can be used to gain insight into what sort of representation is best suited for storing this type of information.

### Example two: a sticky plan

Another example of RSA appears in Figure 4.3. This is a section of transcript from the Group Homework Tool experiments; the two subjects have just started working on drawing a figure, and are presented with code in their shared editor which provides a framework for their assignment. The two are discussing whether it is better to delete the sample code and start fresh, or whether (as the instructions say) it is necessary for them to complete their assignment.

| | | |
|---|---|---|
| | ⋮ | |
| 4. | B: should i delete that stuff<br>they just copied into the main<br>window on my screen? | [Ref–4a code: that stuff]<br>[Ref–4b plan: delete Ref–4a]<br>[Ref–4c ui: main window (shared editor)] |
| 5. | A: no | Ref–4b |
| 6. | B: ok | Ref–4b |
| 7. | A: it says to ignore it | [Ref–7a instruction: ignore Ref–4a] |
| 8. | B: but its in the middle of the<br>screen now, where we type | Ref–4a<br>Ref–4b<br>Ref–4c |
| 9. | A: on the left it says that<br>we need it to start | [Ref–9a instruction: start with Ref–4a]<br>[Ref–9b ui: on the left (instructions) |
| 10. | B: ohh ok | Ref–4b |
| | ⋮ | |

**Figure 4.3: Analysis of data from the GHT experiment (see Figure 1.2).**

On line 4 — the first three lines contain greetings, and are elided — user B asks whether he can delete the sample code ("that stuff") from the shared editor ("main window"). We tag both referents, as well as the proposed plan to delete. This requires creation of a new type, "ui", meant to encapsulate references to UI elements within the system.

User A rejects this plan on the next line; we note the reference to Ref–4b. User B initially agrees (line 6) to leave the code. User A then presents further evidence gleaned from reading the instructions that the framework code should be ignored; we tag this as being of type "instruction", for when the users are discussing the instructions presented to them.

Line 8 finds user B presenting additional evidence for his plan to delete the code — it is simply in the way. This contains explicit reference to Ref–4a and Ref–4c, and indirect reference (primarily visible through use of "but", indicating a continuation of the previous negotiation of the plan) to Ref–4b. We note all three.

On line 9, user A continues to present evidence that the code should remain, refer-

ring explicitly to the UI element containing the manual ("on the left") and repeating a specific instruction from it. We note both new referents, and include in one the explicit reference ("we need *it*") to Ref–4a. Finally, on line 10, user B accepts the rejection of the plan; the users then begin modification of the code to complete the assignment.

**Example three: locations and plans**

A longer sample of tagging discourse using RSA is shown in Figure 4.4. This is a snippet starting twenty lines into a VesselWorld session; the users have greeted each other and taken some initial steps to divvy up the search space and establish names for areas in the harbor.

In this example we see a variety of plans, both short and long, as created by users. For example, on line 20, crane1 creates a plan to 'trash' (retrieve and deposit) a waste (the 'mX') on the large barge ('BB'); this will complete the waste's journey. This plan is proposed on line 20, explicitly acknowledged by crane2 on line 21, and then used as a point of reference on line 30. After this, it is executed and not mentioned in chat again. This is a fairly typical life cycle for plan referents in VesselWorld — after brief, possibly trivial negotiation, the plan is executed without comment, perhaps acting as a coordination point later on.

We see this pattern of plan discussion repeated in Ref–25a. This plan, to search for waste along the northern edge of the harbor, is proposed to no response — either implicitly accepted by the others, or unnoticed – and then eventually announced as commencing on line 29. This shorter life cycle, sometimes with the announcement of inception omitted, was quite common and drove down the average plan lifetime.

Users in most systems settled on this level of peripheral awareness of the plans and actions of others — no need to necessarily get confirmation of a plan's acceptability,

| | | |
|---|---|---|
| ⋮ | | |
| 20. crane1: | I'm going to trash that mX on the BB | [Ref–20a plan: put Ref–6a on Ref–14a] |
| 21. crane2: | k | Ref–20a |
| 22. crane2: | found the sb | Ref–2a |
| 23. crane2: | nothing else here | [Ref–23a location: here] |
| 24. tug1: | L waste in the SW corner is LX | [Ref–4a waste: lx@572,141] Ref–13a |
| 25. crane2: | moving East on N | [Ref–25a plan: crane2 moves east on Ref–17a] |
| 26. crane2: | check | Ref–4a |
| 27. tug1: | I guess I'll sweep the bottom, west to east | [Ref–27a location: the bottom] [Ref–27b plan: tug sweeps Ref–27a, w to e] |
| 28. crane2: | k | Ref–27b |
| 29. crane2: | sweeping N w-e | Ref–25a |
| 30. crane1: | after I drop of this mX (that's the one from 400 125) | [Ref–30a plan: ?? after Ref–20a] Ref–6a |
| 31. crane1: | I'll work my way across the bottom, | [Ref–30a plan: crane1 works across Ref–27a after Ref–20a] |
| ⋮ | | |

**Figure 4.4: Users spent much of their time negotiating plans for waste retrieval.**

but a definite need to communicate expected and future state in case of conflict and to allow the users to formulate expectations about each others' actions.

This example of discourse also demonstrates typical use of location referents, for naming and pointing at specific places in the harbor. Here, the experienced user group has settled on canonical names for regions of the harbor based on compass directions ("in the SW", "on N"); some groups settled on directional words (up, down, left, right) instead. We also see use of deictic locatives ('here', on line 23); as we will see, the data indicates users made use of deictic references very differently than definite references.

On line 23, for example, crane2 refers to the region nearby using the deictic referent

'here'; this referent is never picked up and used again. In contrast, the referent for the southwest corner, REF–13A, is used on line 24, which is 11 lines since it was created; it continues to be used again later in the problem session. Within the scope of this example, "the bottom" (line 27) is first referred to on line 27 (as REF–27A), and is then reiterated on line 31. These definite references were frequent and long-lived, in contrast to the deictic referents, a fact that will be explored in later sections.

Additional examples of how to apply the analysis methods can be found in Appendix C.

## 4.2 Examining the unexplained VW3 results

Using referential structure analysis, we re-examined the unexplained results from the VW3 experiment. To do this we first needed to establish the representational choices available to users of the VW-NO-CR and VW-CR systems and see what paths of information passing they afforded. This was then compared to an examination of their discourse using referential structure analysis. From these it was clear what the mismatches between these two were — mismatches between the available mechanisms for passing information and the actual articulation work the participants needed to do to maintain coordination.

### 4.2.1 Using referential structure analysis

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 log files (from the VW-NO-CR system) appears in Figure 4.2.

This analysis yielded some intriguing results. Most notable was the obvious dif-

| Referent Type | Frequency | References | 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% |

**Table 4.2: Referential structure data from the VW3 experiment.**

ferences between plans and wastes — the two most common types of referents seen. Plan referents, by far the most numerous type of referent, tended to have a short lifetime (averaging 12 lines of chat). In comparison, wastes were relevant for a much longer period (averaging about 169 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 referent is in the conversation. A referent 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.

### 4.2.2 Explaining the VW3 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. To review, in the VW3 experiment, the new coordinating representations were only partially adopted. Specifically, there were two unexpected rejections: the 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 referent 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 under-specified 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 referents. We found that, despite the fact that waste referents 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.

### 4.2.3 Representational choices in VesselWorld

VesselWorld participants have only a small set of representations available to them for coordinating their activity. Primary among these is the textual chat, which gives a very flexible means of expression. In addition, there is evidence of participants coordinating their activity by means of visibility of the actions of other nearby vessels in the harbor. Participants also made extensive use of private markers, which are a way to place an annotation (visible only to the user) on a section of the harbor. An example marker can be seen in Figure 3.1 (page 50), just below the Info window (with the annotation "lwaste tug said"). Users used these extensively to keep track of waste information.

We mapped out the procedures users created for reporting waste barrels, coordinating their waste handling activities, creating plans for action, and so forth. We developed a simple method for making this sort of representational work explicit; the information transfers are represented diagrammatically as a linked set of representations, with the types of information exchanged noted. These diagrams are then used in conjunction with the observed behavior of users, notably areas where they have dif-

ficulty coordinating, to find potential mismatches between the work practice of users and the representations. A discussion of the representation systems and information flow within VW-CR and VW-NO-CR follow.

### The VW-NO-CR representation system

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.

This process is shown diagrammatically in Figure 4.5, which shows the way information about a barrel of toxic waste is typically discovered, reported, transcribed, and remembered within the system. A barrel of waste is discovered by a single user (this information flow is labeled '1' in the Figure) in the World View; the user almost always then uses the Info Window to acquire more detailed information about the barrel (2). These two 'read' steps are prone to error.

The information is temporarily stored in the user's working memory while it is transcribed into the Chat window (3). This step requires the user to re-encode the information into a textual format, and so provides another opening for errors to creep into the process. The textualized information may then be noticed by another user, who must understand the textual information (4), and may additionally make a marker (5) and place it on his personal copy of the World View for future reference.

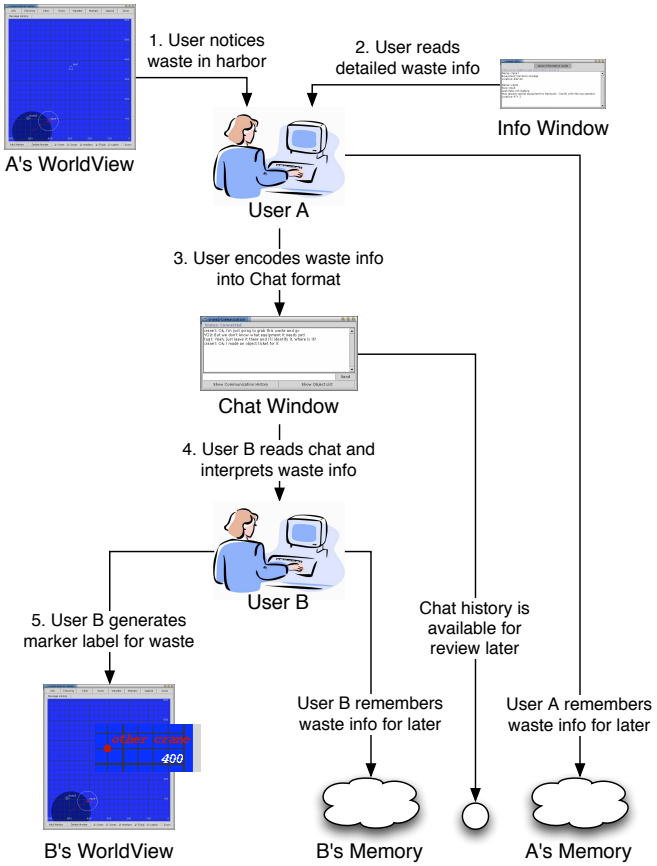


**Figure 4.5: Path of waste info in the VW-NO-CR system.**

After this exchange, information about the toxic waste is located in a minimum of four, and potentially eight different representations:

1. The World View (limited to captains whose vessel is nearby)
2. User A's memory
3. The chat history
4. User B's memory
5. (optionally) User A's marker
6. (optionally) User B's marker
7. (optionally) User C's memory (not shown)
8. (optionally) User C's marker (not shown)

This duplication is in general harmful; the more representations that are storing the same information, the more likelihood for conflicts between them, and the more communication that needs to be exchanged to keep them synchronized. However, in the VW-NO-CR system, the duplication was necessary; as we will see, one of the advantages of the VW-CR system was reducing this number dramatically.

Note that this is a somewhat simplified view of actual work practices; for example, many users created a marker locally before reporting waste information, or failed to make waste markers based on communication from other users. Likewise, it ignores the subtle but important fact that the equipment property of a barrel of waste is only visible to the Tug captain. However, this representation of information flows within the system is useful for demonstrating the basic concept of a representation system.

### The VW-CR representation system

The VW-CR version of the system included a number of new representations for information. These significantly modified the information paths within the system. For example, introduction of the Shared Planning Window meant that the information flows related to reporting current plans from the Chat Window moved into Shared Planning. Figure 4.6 shows the path of information during waste reporting in VW-CR, for comparison with Figure 4.5.

Here the difference in the two systems appears slight. The most noticeable difference is that step 5, marker creation, has become automatic: due to the structure way information is entered into the Object List, the system is able to position a marker label automatically. However, this view from distributed cognition failed to capture the complexities that are altering the task so dramatically. We began a thorough re-examination of the data to explore what we could use to extract this crucial information.

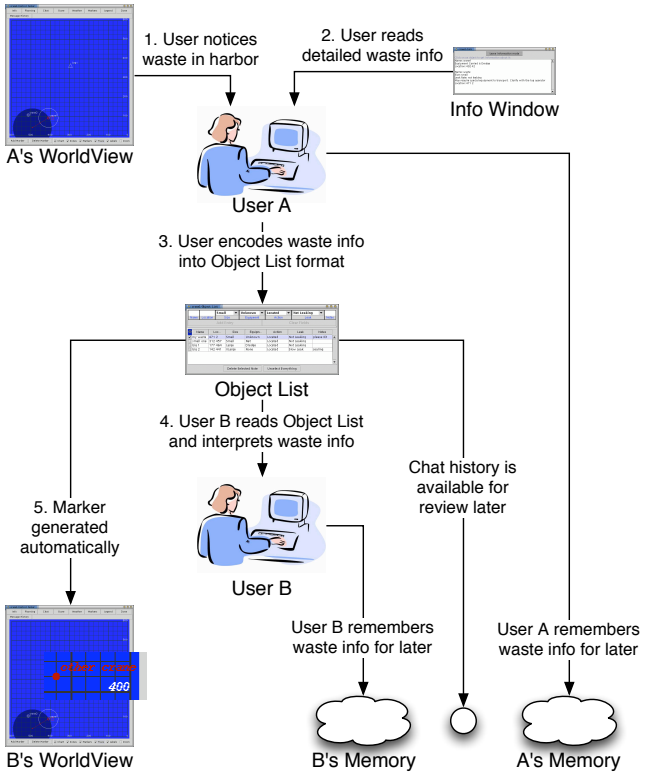


**Figure 4.6: Path of waste info in the VW-CR representation system.**

## 4.3 The "visible" VesselWorld experiment

To verify the utility of referential structure analysis, I used it to explain results from the Visible VesselWorld (VVW) system. In this experiment, a version of VesselWorld was created where the entire harbor was always visible to all users, in contrast to the very limited view presented in the original version. This was compared to the original version, and differences in system use were compared to predictions. This minor change in representation system led to significant changes in user interaction. Using referential structure analysis, and the information flow view of the interaction, I was able to successfully explain these changes to the interaction.

The "Visible" VesselWorld system was generated to test the ability of referential structure analysis to predict the impact of a small change in the representation sys-

tem on the discourse and workflow created by participants. The system itself was constructed to be identical to the regular VW system, but with one minor but crucial change: the spotting range of vessels was increased to make the entire harbor visible at once. Two versions were made, one based on VW-NO-CR, and one based on VW-CR.

This change meant that users could see all barges, vessels, waste barrels, and spills at all times, rather than having to traverse the harbor in exploration. Additionally, it meant that the harbor itself could serve as a useful representation of waste and barge information; rather than having to chat and record this information locally, users could simply use the available information in the world as auxiliary storage for the information.

### 4.3.1 Modeling and predictions

The basic flow of information in VVW is shown diagrammatically in Figure 4.7. This figure should be contrasted with Figure 4.5, on page 87, and Figure 4.6, on page 89. In this figure, we can see the great simplification this minor change has on the effective flow of information within the system. Rather than requiring constant communication between users to distribute and synchronize information, the availability of an authoritative source of information readable concurrently by all users means that information can flow unidirectionally. All users can utilize the World View and Info Window to get information about any waste, and take action on that information immediately.

Based on this model, I expected the amount of dialogue regarding waste objects to reduce dramatically; I expected the only residual communication about them to be discussion of who and when would handle their removal; no communication of waste existence or details (with the exception of the type of equipment required, which only

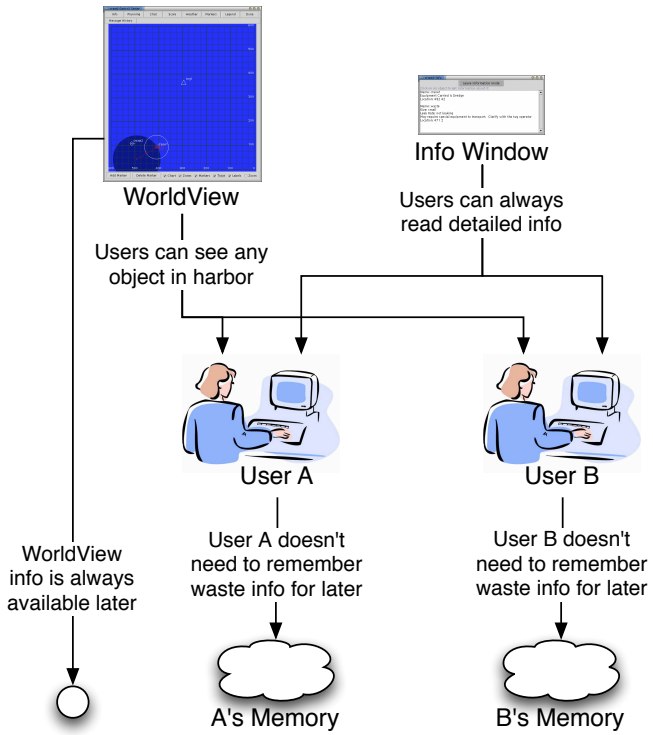


**Figure 4.7: Path of waste info in the VVW-NO-CR system.**

the tug captain had access to) would occur.

As a result, my hypothesis was that, given the availability of information in the World View, users would not chat about waste information much if at all. Specifically, I assumed some chat to establish names for specific barrels of waste, and some high-level planning to discuss the ordering of waste clean-up. We also anticipated that the users would take advantage of complete information to formulate a long-range plan for waste clean-up; however, as we will see, this was not the case.

### 4.3.2 Experimental results

To test the impact of this representational change, I performed an experiment very similar to the VW3 experiment. However, due to financial and time constraints, only four groups of three participants were used, two per condition (VVW-NO-CR

vs. VVW-CR). As before, each group was given a entrance questionnaire relating to subject population; trained in use of the system for two hours; solved problems for ten hours; and was then given an exit survey to record their reactions and feedback. Also as before, the first five hours of experimental data was quite noisy, and had to be discarded; results were drawn from the final five hours of game play. The experimental subjects were drawn from the student body, and were paid for their participation.

After performing the experiment, I analyzed the data and discovered a number of differences between the work practice of VVW users as compared to users of the regular VesselWorld system. These changes were similar for both VVW-NO-CR and VVW-CR groups. The change in representation system had three major effects on the work practice of users:

1. Elimination of exploration
2. Reduction in long-term planning
3. Repurposing of chat about barrels of waste

These three effects on the interaction are discussed in the following sections.

**Elimination of exploration**

Users of the VVW system, in contrast to users of the non-visible VW systems, did not explore the harbor to locate barrels of toxic waste, because that information was readily available. This was the most obvious and expected effect. This had the side effect of shortening the minimum number of steps required to complete most problems, which in turn reduced execution time for those problems and reduced opportunity for error.

This result was expected. However, it made it impossible to directly compare quantitative measures between VVW and VW groups. Instead I went back and

estimated the percent of each VW session that was occupied with searching for barrels and assessing the situation, and compared the data accordingly.

Because they did not need to spend time searching the harbor, VVW groups were able to complete problems of comparable complexity roughly 15% faster, as shown in Figure 4.3. This corresponds well with estimates of how much time the groups in VW3 were spending searching the harbor, and therefore indicates that users in the VVW were not able to otherwise complete problems faster.

| Change from VW3 to VVW | VVW-NO-CR vs. VW-NO-CR | VVW-CR vs. VW-CR |
|---|---|---|
| **Efficiency of solution** (rounds per unit complexity) | 35% fewer ($p < 0.05$) | 23% fewer ($p < 0.05$) |
| **Solution time** (minutes per unit complexity) | 53% faster ($p < 0.01$) | 30% faster ($p < 0.05$) |
| **Speed of play** (rounds per minute) | 14% more ($p < 0.05$) | 16% more ($p < 0.05$) |
| **Mistakes** (domain errors per minute) | 84% fewer (not significant) | 55% more (not significant) |
| **Communication** (lines per unit complexity) | 15% less ($p < 0.05$) | 19% less ($p < 0.05$) |

**Table 4.3: Comparing Visible VesselWorld data to VW3 results.**

The groups also required significantly fewer rounds of action — 35% fewer for NO-CR groups and 23% fewer for CR groups. About 15–20% of this can be accounted for by the removal of the exploration phase, but the remainder must be attributed to the altered representation system. This was probably due to the availability of perfect information about location of wastes; this allowed users to quickly come up with a general, implicit plan for ordering waste removal. However, as we will see below, users spent very little time discussing or constructing long-term plans for action.

### Reduction in long-term planning

Another significant result was the decrease in long-term planning. Because participants had access to waste information from the beginning of the session, the experiment revealed them to be much more opportunistic in their collection of waste barrels. As noted, the continuous availability of the waste locations and the locations of the other vessels meant that groups could formulate more efficient plans for collecting the waste, and were better able to opportunistically handle nearby wastes. This also allowed for more efficient interweaving of plans, reducing the number of steps that participants spent waiting for each other. Likewise, the reduction in minor errors — for example, forgetting to deploy equipment, and having to waste a few turns correcting the situation — increased the efficiency of the solution.

### Repurposing of chat about barrels of waste

Communication about waste was altered significantly. This result was primarily in line with expectations; however, as seen in Figure 4.3, chat quantity for CR groups was basically unchanged when the 15% reduction in effort due to elimination of exploration is taken into account.

Tracking types of chat by properties of waste barrels revealed more detail about what types of information were eliminated from chat. Users did not report waste existence, as expected; they also did not discuss waste size or position, since these were clearly visible to all. In general, chat was reduced to exchanges about assignment of responsibility, and occasional queries about waste equipment to the tug captain. However, because of the opportunistic planning, there was an increase in both plan discussions and negotiations; users spent more time discussing plans.

In the CR version, the change in overall chat quantity was also minimal. Whereas

in the restricted-view VesselWorld participants tended to use the Object List exclusively to track waste information, in the unrestricted-view version participants could access such data directly from the harbor. Because it was easier to access the details about a waste — for example, its size or position — participants tended to use the Object List simply as a way to name wastes, rather than a method for consolidating information about a waste.

As a result of this decreased dependence on the Object List, the data showed an increase in textual chat about waste; the information flow that had previously moved from chat to the Object List did not migrate for VVW-CR groups, because the perceived usefulness of chat vs. the Object List was changed. Since users were already primarily attending to the Chat Window, and only referred to the Object List to built a marker which assigned names to domain objects (i.e., using the harbor-marking properties of the Object List as a way to generate shared markers), they did not bother to use the other columns of the Object List to store information. As a result, the information those columns would have contained instead migrated back to the Chat window.

**Other results**

Finally, the error rate was so low that statistical conclusions about it were impossible. Informally, the VVW-NO-CR groups made very few errors, and the VVW-CR groups made more errors than their VW-CR counterparts, but these results are probably due to variation in group competency rather than a result of the representation systems. One of the CR groups made a large number of careless errors due entirely to a single subject not paying attention; this skewed the data tremendously. Likewise, one of the NO-CR groups had a session that didn't go very well, wherein they committed an unusual number of errors.

These factors impacted the data negatively, making the variance of the populations high enough to prevent any significant conclusions. Informally, when these effects are discounted, both VVW-NO-CR and VVW-CR groups committed far fewer errors in VVW than in VW3. Most noticeably reduced, as expected, were errors due to problems with information flow about wastes; errors such as that seen in the segment of discourse shown in Figure 3.4 are now easily circumvented, because each user has continuous access to definitive information about the barrels of waste.

### 4.3.3 Summary of VVW

Using this data analysis, I was able to explain the changes in participant behavior observed in the modified representation system of VVW. This showed that by carefully modeling information exchange, as revealed by communication between participants, an analyst is able to achieve an understanding of the interaction and the mediating effect of the representation system. In future chapters, these observations will be put to use in creating groupware that matches an existing interaction.

## 4.4 Visualizing experimental data

Visualizing the analysis data was another useful tool used to understand referential structure data. A scatter plot showing Lifetime of Relevance vs. Number of References is useful for revealing general differences between referent types. In the scatter plot shown in Figure 4.8, distinct populations representing plan, waste, and location referents are visible. In this figure the referents from an analysis of non-CR Vessel-World groups have been plotted with one axis being the *percent lifetime of relevance* — the ratio of the number of utterances between first and last mention of the referent to the total number of utterances in the session — and the other being the *number of*

*references* to the referent during that span. Due to the wide variance in the data, the horizontal axis is logarithmic. The source data has also been jiggled slightly (small fractions have been added to the discrete source values) to reveal instances where multiple data points overlap.

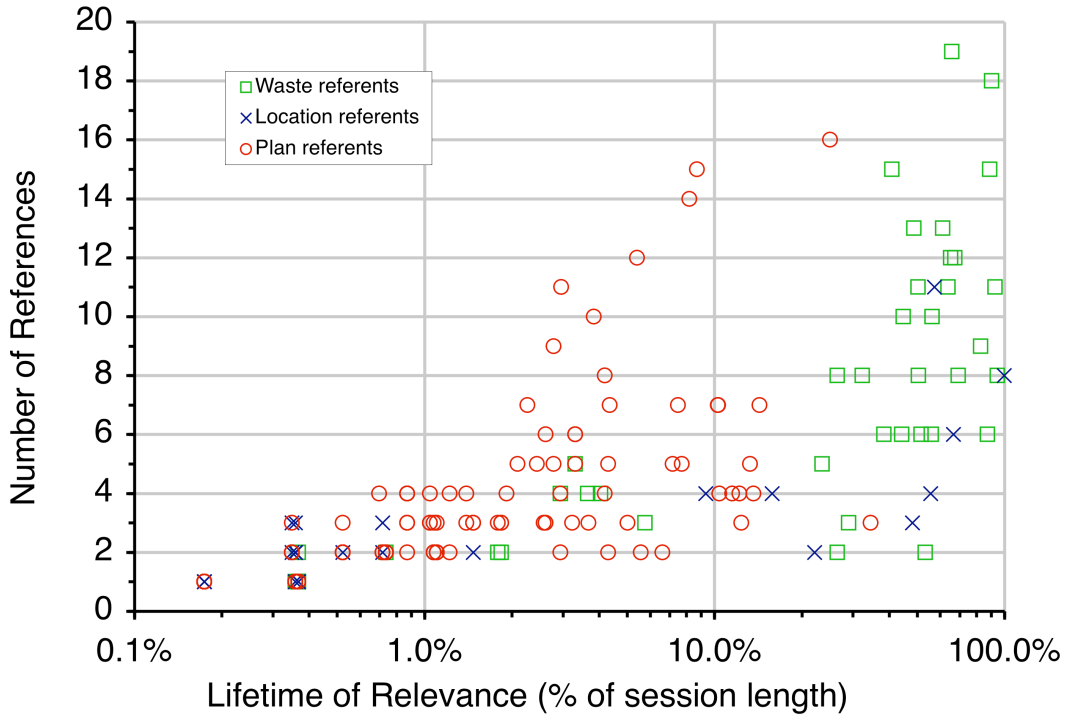


**Figure 4.8: Information feature differences are made visible in a scatter plot.**

This sort of comparison graph is useful for examining the data for outliers and to check the distinctiveness of referent types. If the populations of two different referent types are very similar, the analyst may wish to examine whether they are variants of the same sort of information.

### 4.4.1 Using visualization to explore the data

In Figure 4.9 we have plotted a representative sample of "location" referents, where users have referred to a particular location in the harbor, from VW-NO-CR groups.

After plotting the Location referents in this fashion it became clear that there were two distinct clusters of referents — those with percent lifetime less than about 1%, and those with lifetime more than about 10%. The division between the two populations was striking, and aroused curiosity.

By going back and examining the source data, it became apparent that the referents in one set corresponded to those to whom only deictic references, such as "over here", or "right there", were made. These referents had a much shorter lifetime than those in the other group, which corresponded to static location references such as "362,163", or "near lbarge" ('near the large barge').

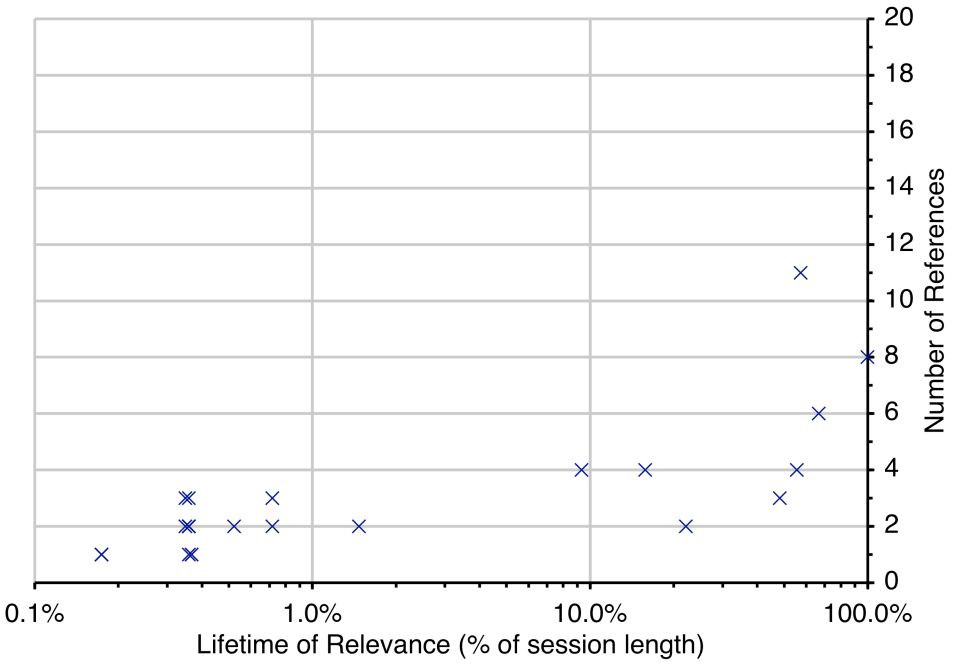


**Figure 4.9: Scatter plot of location referents reveals clusters of deictic vs. definite references.**

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.

A similar scatter plot for the waste referents can be seen in Figure 4.10. Though the result is less striking visually, I discovered that there were also two sub-populations within the waste referent data.

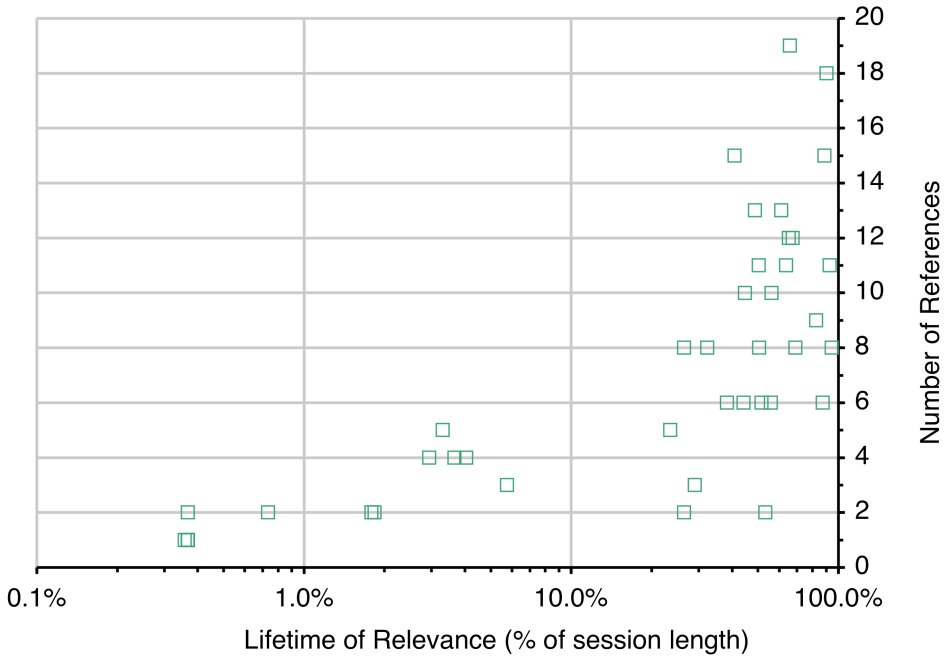


**Figure 4.10: Scatter plot of waste referents reveals short-lived wastes to be 'phantom' referents.**

Originally I was struck with the distinct populations below and above 10% percent lifetime. However, investigation of the referents in this case did not show an explanation like that found for location referents. Instead, this revealed that waste

referents with three or fewer references were invariably 'phantom' referents — referents for whom no actual waste existed, and which therefore only existed in the minds of the participants. The population was split, albeit less obviously, along a horizontal line, rather than a vertical one.

Encouraged by these findings, I examined the data for plan referents, shown in Figure 4.11. However, this analysis revealed no division of referents the way waste and location referent data did; it seemed likely that plan referents had been correctly categorized from the start. To investigate the data further I turned to other statistical techniques, in this case a histogram of plan lifetimes.

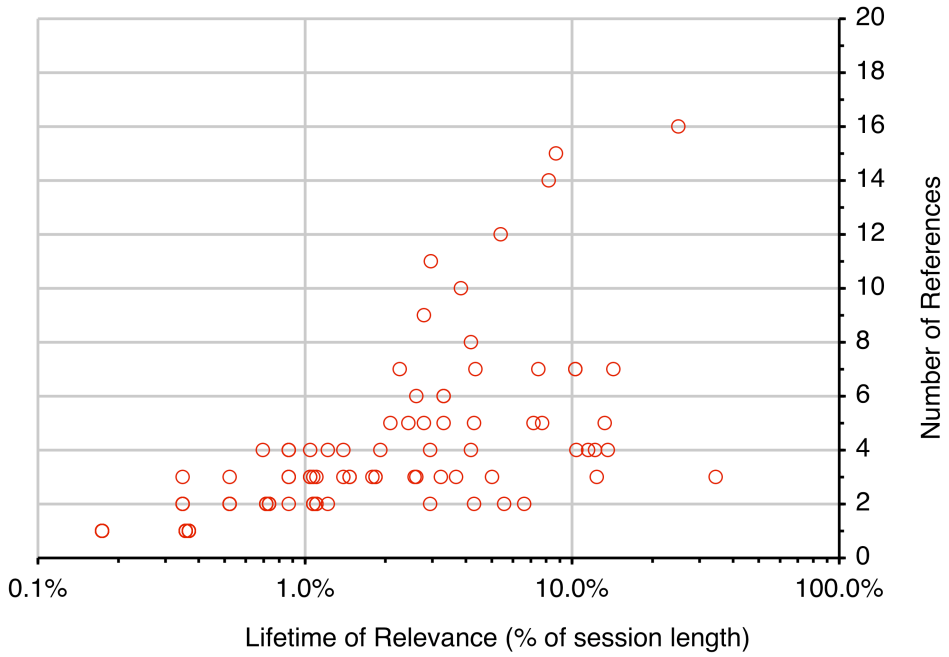


**Figure 4.11: Scatter plot of plan referents revealed no obvious structure.**

A histogram of the number of references to each plan is shown in Figure 4.12. From this, I was able to guess that the lifetime of plans follow a roughly exponential distribution — shorter plans occurred far more often than long plans, and in aggregate the graph resembled an exponential decay (with a few outliers, like the extremely lengthy plan seen at the right of Figure 4.11). This would be an expected result for a

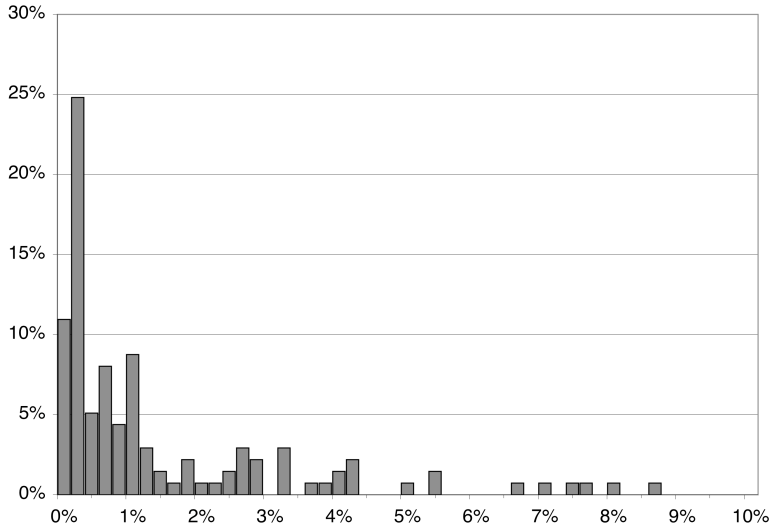


**Figure 4.12: Histogram of plan referents vs. percent lifetime.**

process which could transition to a final state in a pseudo-random fashion, such as the length of a game of Russian Roulette. This lends credence to our view of conversation about plans as a series of negotiation steps followed by acceptance; the distribution points to a rough equivalence of the probability of any one continuing step in the negotiation. A more accurate model, with more data, might also be able to extract the predicted possibility of negotiation of a plan continuing after each step; this detail can be fed into another sort of analysis, such as task analysis or a language/action model of the interaction.

This sort of statistical investigation of referent data is at the heart of referential structure analysis. By providing the analyst the ability to apply the existing body of statistical analysis techniques to usage data, referential structure analysis provides new abilities and opportunities for understanding the interaction. I have attempted to demonstrate some of the ways in which the conclusions from this analysis can be used. In the next chapter, I will demonstrate how these observations are used to drive redesign directly; however, first, I will discuss some of the limitations of the method.

## 4.5 Limitations of the method

Performing an analysis of the referential structure of an interaction does pose some difficulty. Ambiguity abounds; users exchange ill-formed or incomplete utterances; referents may resist categorization; it can be impossible even for an analyst with good playback tools to reconstruct the context of the activity completely. This section explores some of these problems and gives some guidelines about how to achieve good results despite these obstacles.

However, this is not always a straightforward process; ambiguity, similarity between referents, mistakes on the part of the participants, and (in complex domains or protracted session) the sheer number of possible referents can make this task quite difficult. The analysis method presented here attempts to aid the analyst to overcome these difficulties. This section will detail some of the more common problems an analyst may face when attempting to perform referential structure analysis, highlighted by examples pulled from the VW3 corpus.

### 4.5.1 Collecting data

Real-world audio and video, the most popular recording media, are notoriously difficult 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. Hence, ethnography using these technologies can be somewhat limited; it is difficult to record a complete picture of the interaction. Even high-fidelity techniques such as video recording may lose a fair amount of interaction detail due to limitations of the recording and playback technology.

However, with computer-mediated coordination, it is possible to use the computer itself to record the interaction. Because all interaction is mediated by the computer, constructing systems that record and play back what users do makes the entire interaction available for analysis. The experimenter's job is made easier by the very technology that enables the collaboration. In online interactions, where the activity of participants is entirely mediated by computer, it is possible to record the entire interaction of the participants simply by logging the data transmitted by the computers.

### 4.5.2 Choosing referent categories

A fundamental part of the referential structure analysis process is the identification of domain-specific referent categories. When analyzing a domain, it is important for the analyst to examine the data thoroughly and without preconception, so as to accurately identify what sorts of information the users are talking about. It is entirely possible for two analysts to come up with different sets of referents because of incomplete data, differences in judgment or interpretation, or simple disparity of opinion about the relative importance of various types of information.

One facet of this process is the necessity to decide the level of detail to use when tagging the transcript. For example, in the VesselWorld domain, the decision was made to lump together all references to waste objects. An argument can be made that this category should be split by size, into "solo-lift waste" and "joint-lift waste". Such an analysis of the data will necessarily give slightly different conclusions — perhaps it would show differences in access pattern between these categories of waste, or perhaps the subdivision of the sample set would reduce conclusions to insignificance. In general, more detail in the analysis tends toward more detailed but less confident conclusions.

Another difficulty is posed by referents that are similar in nature or share many common features, such as Date versus Time. In such a case, the analyst must examine the data carefully, to understand how each sort of information is dealt with by the participants. This insight can then guide the selection of referent categories. In a task such as planning a class syllabus, Time and Date are used quite differently — Time is a factor of in-class planning, Date more of a factor in topic choice and ordering. In contrast, a task such as plotting a satellite retrieval mission might treat Time and Date as facets of a single unit, and in such a domain the two should most likely be merged into a single referent category.

Lastly, there is the problem of what detail to analyze at all. Participants make many references in the course of their activity, and attempting to track all of them becomes prohibitively time-consuming. Notably, many participants lighten the mood with humor or non-task communication. While this behavior can be important for team-building, references made therein are unlikely to require their own referent categories.

### 4.5.3 Ambiguity

Uncertainty and outright error in analysis are a fact of life; however, the resulting analysis can nevertheless yield important and accurate conclusions when used as a tool for understanding the interaction. The method revolves around being able to consolidate references; that is, form conclusions about when a reference refers to an earlier referent, and should be associated with other references. Although grounding anaphoric references and forming reference chains can be problematic in the general case [134], repeated application of referential structure analysis has convinced me that good results can be achieved by restricting the set of referents under consideration, and aiming for investigative rather than comprehensive results.

Reference structure analysis attempts to follow a cohesive thread within the distributed cognitive system. Information is discovered by a user and internalized; some form of it can then be transferred into one or more of the various representational media available to that user; other users must then notice, interpret, and internalize the information, and then possibly re-represent it. This process is necessarily imperfect, much like an extended game of "Telephone", and this ambiguity not only hampers the users' ability to coordinate but makes the analyst's task more difficult.

Referential ambiguity is not particular to analysis of groupware; it is an inseparable problem of natural language. Ambiguity of reference can easily be demonstrated in natural language:

(1) David took Michael to the store.

(2) He wanted to buy some gum.

He, in (2), is inherently ambiguous; semantics allow us to perhaps give a slight preference to Michael, but only slight. In groupware, however, because the participants have trouble understanding the ambiguous references of each other, they often ask for (and receive) clarification in the course of the dialog. By looking both forward and back through the transcript, the analyst is almost always able to successfully ground an ambiguous reference. In VesselWorld, this sort of situation is very common:

(1) crane1: yes, next turn a small too

(2) tug1: Which sm? 394 71?

(3) crane1: tug, yes

In (1), crane1 produces an under-specified reference ("a small", referring to a small barrel of toxic waste). In this case, there are multiple small wastes that crane1

could be talking about. However, this ambiguity causes problems for tug1, who asks for clarification; by using this exchange of information the analyst can successfully ground the ambiguous reference.

Due to the extended scope of interaction, and the memory aids available in groupware systems, references may refer to incidents or information from a long time prior, making the set of possible referents quite large. In the worst case, a participant can produce a reference whose ambiguity may not be resolvable by examining contributions from other users, hints from later in the transcript, or other such tricks. In such a case, the analyst is encouraged to simply make an educated guess about the referent in question, and note the problem. While this method may occasionally produce an inaccurate analysis, examples of truly irresolvable ambiguity tend to be quite rare. Therefore, the overall quality of the analysis is not greatly reduced.

### 4.5.4 Phantom references

In addition to generating ambiguity, participants in an ongoing interaction often create utterances containing references to objects that don't exist, due to an imperfect understanding of the world. In some cases, two (or more) participants will have different partial views of an external object, and one participant will internalize a version of another participant's view without realizing that the two views are connected. Like the three blind men and the elephant, the participants are unable to immediately understand that they are talking about the same object.

It is important to note that because references in conversation may refer to an object that does not exist, the referent may represent a fictitious referent that exists only in the mind of some of the participants, and that multiple referents may exist for a single "real" referent, because different participants refer to it as entirely separate objects. Nevertheless, the analysis should not attempt to collapse these individual

referents to a single, master referent. Rather than extracting some authoritative set of objects, referents represent conversational objects that any participant refers to, and as such have no independent authoritative source outside of the discourse of the participants.

Spurious references created by these and other sorts of errors can be very persistent, as participants tend to trust in reported information rather than expecting errors in reporting. Tracking them can be very important, as erroneous behavior can often be traced back to mismatches in the common ground of participants that arises out of phantom references.

In Figure 4.13, a user reports a waste using a new convention for vocabulary. Specifically, the user has adopted a convention from the C programming language of indicating negation by use of an exclamation point, as seen in line 9 ("!leaking" means "not leaking"). However, this convention is misunderstood, or the utterance is misinterpreted, by crane1, who records the waste as leaking in her private workspace. This mismatch causes the confusion seen starting on line 31, which continues for another 62 utterances (and almost 15 minutes), and requires an enormous amount of articulation work to resolve. During this exchange, the analyst must keep track of references not only to the actual waste (which is not leaking) but to the phantom waste (which is leaking) that crane1 creates through misinterpretation.

The sort of trust in erroneous conclusions seen here has been shown to be endemic in catastrophic failures in such grave circumstances as the Three Mile Island incident and some airplane crashes [103], so it is understandable that it occurs in the context of VesselWorld. As we saw above in the analysis of wastes within the VesselWorld domain, participants talk differently about phantom references and valid references; this finding has potential for automatically discovering or highlighting these communication breakdowns.

9. tug1: I'm at 10 213, and there's a waste at 102 248. It's small, net, and !leaking.
*...about 1 minute passes...*
31. crane2: shall we explore everythign first
*(crane1 invents the leaking waste)*
32. crane1: well, we should get the leaking one
33. crane1: it needs a net and to be sealed
34. crane2: which one is leeking?
35. tug1: Mine isn't. The 207/1?
36. crane1: tug, didn't you say there was one at 100x250
⋮
61. crane2: the one at 100,250 is no longer leaking.
⋮
*...eventually the cause of the confusion is found:*
87. tug1: No, I said !leaking, which means not leaking.
⋮
89. crane2: sorry missed that
90. tug1: Okay, so no leaks.

**Figure 4.13: A phantom referent is born, wreaks havoc, is tracked down, and finally expunged.**

Phantom references also pose a methodological problem for the analyst. For these methods to be of use, the analyst needs to be able to classify references according to their type. In most cases, spurious references can be handled by creating a phantom referent and tying subsequent references to it throughout the dialog. However, in more extreme cases, the phantom referent is sufficiently underspecified as to make its type ambiguous.

Depending on the desired level of analysis, the analyst can deal with this in a variety of ways. Most simply, the analyst can ignore it, simply avoiding attaching a coding to this reference and references that might descend from it. However, it is often just such errors that the analyst is most interested in ameliorating, and so analysis of the articulation work surrounding that referent is of interest. In such a case, it is preferable to code the ambiguous reference as a typeless, underspecified referent, and

attach possible references to it as possible in the future, until the meaning is clear.

### 4.5.5 Lost and duplicated references

Another problematic case occurs when referents get forgotten for extended periods of time. Some referents can eventually be rediscovered or remembered, but become relevant again after such a long period of time that the relationship between earlier and later references can be a bit obscure. In most cases, the analyst can with clear conscience attach the two reference trees; however, in some cases, participants appear to (re)discover a referent completely anew, and treat it in a new fashion. Similar to the problem detailed earlier, participants may also then treat the new and old views of the object as equally valid and distinct referents, which causes similar problems for the analyst, or they can fixate entirely on the new view of the object. This can be problematic if the analyst is trying to follow the conversation about that object — is the new conversation part of the old one, or should they be considered distinct reference trees simply because they refer to the same referent?

In the end, it is up to the discretion of the analyst; in general, it is recommended to treat the two segments as separate reference trees, but to join them together and annotate the analysis with a note to the effect that the two referents are, in fact, the same referent. The rationale behind this is that, at heart, the analysis is centered more in how the participants in the joint activity handle information, rather than in some sort of external correctness of the information itself. Hence, it is important to draw two conclusions from such an incident: first, the way in which the participants handle the information each time it is in their attention field, and second, the fact that this information was forgotten, and for how long.

### 4.5.6 Lack of communication

With groups that have built up common ground, the amount of task-related communication can drop nearly to zero for routine operations. This leaves the analyst with little to go on to figure out what is occurring in the interaction and how to improve performance. In fact, some groups may not say anything in chat for protracted periods; in one VW3 log file, the group completed a fifteen-minute session while producing only eighteen lines of chat — four of which were discussing the previous session and twelve of them discussing whether they were in fact done with the current one.

In such cases, it is important for the analyst to be able to examine not only such text-based communication as the chat logs used primarily above, but also the other forms of communication available to the participants. In the experimental domains observed, the participants are restricted to communicating via a purpose-built system, which logs all actions. This allows the analyst the luxury of access to all the forms of communication available to the participants. In a more general case, the analyst may have to employ additional means — videotape, direct observation, or other more traditional means of data collection — to get the whole picture. Whatever the source of data, the analyst may still be able to apply the techniques developed above.

### 4.5.7 Excessive diligence

Transcripts of interaction generate a lot of data. Finding conclusions in this morass of information is difficult at best. Drawing conclusions based on systematic analysis of a corpus, rather than on isolated vignettes, requires careful analysis and summary of data. The time cost to the analyst of this sort of careful analysis should not be underestimated, and can pose a significant obstacle to drawing accurate conclusions, especially in a corporate environment where time pressures may force an analyst to

curtail investigation of the data.

Likewise, it is tempting to attempt to identify every single information flow within a system, to track every referent, and to produce a description of and representations for each one. This reveals the time/thoroughness tradeoff inherent in the analysis process: identifying every detail is prohibitively time-consuming, and usually counterproductive. Instead of attempting a top-down description of the system as envisioned, the analyst should work bottom-up from the discourse data to discover the most crucial areas.

The goal of the method is not to support every possible information flow; instead the goal is to take care of the large and important ones by providing representations, and allow the others to filter into unstructured communication channels such as free-form chat or informal meetings. In fact the entire design process can be viewed as taking information out of these 'margins' of communication and migrating it to custom-built representations. This marginal communication is examined, ranked for importance using measures such as frequency of appearance, and supported by introducing new representations or augmenting old ones. This process can then be repeatedly applied and refined until a successful system is achieved.

For these reasons, it is important to streamline the analysis as much as possible, while simultaneously highlighting the crucial portions of the data. As discussed in the explanation of the analysis technique, the level of detail the analyst goes into in the analysis is at his own discretion.

There are a few specific ways to reduce analysis effort. Tracking referents of only a particular type, or a small set of types, reduces the total overall analytic effort. After a preliminary analysis, the analyst may decide that focussing on a particular facet of coordination may provide the best return on investment of time. For example, in VesselWorld, an analyst could decide to focus especially on the discussion of waste.

This method reduces the gross time needed for analysis, but can lead an analyst to miss important conclusions about other sorts of data. Likewise, using a Part-of-Speech tagger to highlight noun phrases and other linguistic techniques may simplify analysis. Possibilities regarding this approach will be discussed in Chapter 8.

### 4.5.8 Hierarchies of knowledge

In general, items of information generally contain other items of information in a semi-hierarchical fashion – the slot/frame abstraction is one useful perspective for examining this. It is tempting to attempt to build a complete taxonomy of the information types. This process can be quite time-consuming, and usually ends up requiring effort well outside the scope of the analysis.

For the purposes of our methods, an analyst might end up examining both an encompassing item as well as one or more of its sub-items — depending on how the users refer to them in the discourse. Once again, the goal is to go from observation to recommendation, not to establish a perfect model of task communication. If users talk about both waste barrels and waste barrel locations, then it is useful for the analyst to treat these as separate information flows while also noting the implicit connection between them. In practice, rather than getting bogged down in creating a complete ontology for information types and flows within the system, focussing on the areas where participants are having the most difficulty will generally produce the best results.

### 4.5.9 Non-stationary work practice

Another serious concern is that the work-practice of individuals may evolve over time. Procedures get created which significantly alter information flows. New practices

arise. Information gets used in new ways; old ways get modified. The analysis methods presented herein can only analyze a snap-shot of a system. This hearkens back to the first figure in the thesis; it is necessary for a maintainer and analyst to support a community of users in an ongoing fashion.

## 4.6 Summary

Referential structure analysis provides a useful method for analyzing an interaction. By analyzing references found in the discourse, an analyst can construct a model of the important information flows within the system. This model reveals the way that participants use and store information, gives insight into the different types of information being passed around, and lets the analyst determine which types of information make up the bulk of participants' representation work.

In the next chapter, we will demonstrate how this model allows the analyst to recommend new representations, based on a pairing of information features and representation properties. We will also discuss some of the limitations of the method.

# Chapter 5

# Experimental Verification

A major concern when creating a new tool or procedure is that the method be useful. This can be broken down into a number of major criteria:

1. Correct: the method must produce results which are based on sound principles and which are valid.
2. Teachable: it should be possible to teach the method to others, who are then able to successfully apply it.
3. Reproducible: it should be possible for different people to get similar results when applying the method to the same inputs.
4. Domain independent: The method should preferably be applicable to more than one domain, returning similarly-useful results in each.

In this chapter, I will demonstrate the utility of referential structure analysis by showing experimental evidence of each of these criteria.

## 5.1 Correctness

I performed a statistical analysis of the VesselWorld data presented above to investigate whether the source data for the method, namely, referential structure and referent types, were valid sources of information. Validation of this hypothesis provides an important guarantee that the conclusions I am drawing from referential structure analysis are meaningful.

This verification was split into two parts. First, I showed that the referent types were distinguishable using only referential structure. By showing this, I could prove that analyzing referential structure was useful in determining the type of a referent, and that there was some significant difference between referent types that should be accounted for in the interface.

Second, I examined the referential structure of the discourse of groups with the same task but different representation systems and showed that they were different. By holding all variables other than representation system constant, I demonstrated that referential structure must be dependent on the representation system. This in turn allows us to say that changing the representation system properly will have an effect on the discourse, potentially reducing articulation work for users.

These two observations can be formulated as the following two hypotheses:

1. Differences due to referent type are significant across groups within-condition: that is, referential structure depends primarily on referent type and not on group membership.
2. The variance of data between-conditions (i.e., non-CR groups vs. CR groups) is high: that is, the differences between referent types are significant across groups.

Statistical evidence for both hypotheses follows, supported by data from the VW3

experiment.

### 5.1.1 Referential structure depends on referent type

The first hypothesis was that different types of information have characteristic information features in the discourse. These patterns are observable and distinguishable via reference tracking in the discourse. Information access patterns, in turn, provide evidence that certain representations will improve performance. Hence, proving this hypothesis verifies that the discourse can be used to investigate the impact the representation system has on the interaction.

I analyzed the VesselWorld data to investigate whether differences in the referential structure were correlated with referent type. The hypothesis in question was that groups using the same system for the same task exhibit similar statistical measures for each type of referent. Formally, I measured whether differences between referent type statistics (lifetime and references) are larger than the variation due to having different people in a group.

The data supported this hypothesis. I found that different non-CR groups exhibit slight variation in how they handle a particular type of information, but that these differences are minor in comparison to the differences between information types. To show this statistically I examined the differences between participant groups in this experiment. I chose to focus on the three most commonly occurring referent types (plans, wastes, and locations). An F-test on these referent 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 type of the referents.

### 5.1.2 Representations affect referential structure

The second hypothesis was that groups using the different systems for the same task would generate significantly different referential structure. That is, by providing alternate representations for certain types of information, the system would alter how participants talked about that information. To verify this hypothesis I used data from the VW3 experiment that compared two strongly related representation systems — the system available to the CR groups, and the system available to the non-CR groups. My goal was to show that differences in the discourse structure due to group membership are insignificant compared with the differences across information types and across representation systems. In other words, all groups using the same representation system for the same task have similar information access patterns for a particular type of referent. However, groups with different representation systems have significantly different patterns of discourse, even when they are engaged in the same task.

Formally, I compared the distributions of referents for CR groups and non-CR groups to determine whether they could have come from the same source population. I compared referent statistics (lifetime and references) from the CR groups to comparable data from the non-CR groups. A T-test performed on the data generated for the two experimental conditions showed that it was unlikely that the differences between the data for the non-CR and CR conditions were due to chance; therefore, I attribute the change in information access patterns to the change in representations. A summary of the relevant figures appears in Figure 5.1.

As expected, the effect that introducing new representations had on referents was dependent on referent type. Most noticeable is the strong effect on waste referents. As explained previously, the new representations altered the way that users talked

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

**Table 5.1: Representation system has an effect on information access patterns for some referent types.**

about waste referents much more heavily than they way they talked about plans and location referents. This is reflected in the data: plan referents (which were generally unaffected due to rejection of the Strategy window) show little effect, whereas the statistics for waste referents (strongly affected by the introduction of the Object List) show a very significant change. Location referents, whose characteristics were changed somewhat by the availability of the Object List "location" column, show a moderate degree of separation.

Overall, this test showed that there were significant alterations in the referential structure generated by participants after new representations were introduced. This is in line with the observations of other researchers, who have shown using other features of conversation that availability of alternate representations significantly alters how participants talk [28, 47].

### 5.1.3 Teachable

With the help of a class of students, I ran two experiments to gather data on how well the methodology could be taught and employed on novel domains. In this section I summarize the successes and failures of this vetting of the experimental method.

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 [78, 56]. 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 the analysis techniques described in this thesis.

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. The results of this are discussed in detail below, where they also demonstrate the ability of the methods to be applied to a variety of domains.

### 5.1.4 Reproducible results

The next criterion was that analysts could use the methods to draw similar conclusions from the same data. To test this, the students were asked to perform a referential structure analysis of four standard transcripts.

These transcripts were pulled from a data set of dyads engaged in a pairs programming session using the GREWP tool. Here, the two experimental subjects were instructed to write a program that drew a picture of a human figure. The drawing routines for each specific body part – head, body, legs, etc. – were required to be different procedures. 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, I 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 experiment allowed us to test the inter-coder reliability of the methods presented here.

#### Measuring agreement

Each of the four transcripts was analyzed by five pairs of students, with each pair of students seeing two transcripts. Student groups were not allowed to consult with each other during the process. The resulting analyses were similar, though there were minor variations in results from group to group. Each analysis was compared to each other analysis.

To quantify the agreement I employed Cohen's Kappa. Cohen's Kappa [30] is a standard method for comparing two or more analyses of a single set of data. It is

meant to be applied to a situation where independent analysts are sorting items into one of a number of categories. It computes the probability that the two classifications differ from chance, and is expressed as the positive probability that the two analyses are identical.

There were some significant complications in applying Cohen's Kappa to this data because the task was not a strict category-assignment task. To apply Cohen's Kappa to the data, I took every pair of groups for each transcript and compared their analyses. For items, I used the referent clusters described above. For categorization, I assigned each group a 1 for each cluster if they had tagged a referent belonging to that cluster, or a 0 otherwise. This resulted in a Kappa rating for each pair of groups on a particular transcript. The values for each group were averaged across transcripts, giving the values in the cells of the table; these represent the average match between this group, in this transcript, to all other groups which analyzed this transcript.

These unweighted kappa values are summarized in Figure 5.2. Note that because each group only analyzed two of the four transcripts, half of the cells are empty. Averages for each row and column are shown at the edge of the table, representing overall agreement for each transcript, and average agreement for each group. Finally, an overall average was calculated.

The kappa values ranged from 48% to 71%. The range 50% to 70% is considered "moderate" agreement on the scale for Cohen's Kappa. To ascertain the validity of these results I examined why and how analyses differed.

Analyses differed in three ways: first, analysts identified different referents as salient, and so tagged different subsets of a potential set; second, analysts identified similar referents as being of a different type; and third, analysts differed on when references to a particular referent occurred. To overcome these complications, I clustered the referents found by the student analysts into a master set of referents; most, but

| Transcript: | 1 | 2 | 3 | 4 | Mean |
|---|---|---|---|---|---|
| Group 1 | 51% | 70% | | | 61% |
| Group 2 | 61% | | 59% | | 60% |
| Group 3 | 61% | | | 68% | 65% |
| Group 4 | | 70% | 48% | | 59% |
| Group 5 | | 64% | | 62% | 63% |
| Group 6 | | | 55% | 69% | 62% |
| Group 7 | 62% | 71% | | | 67% |
| Group 8 | | | 53% | 63% | 58% |
| Group 9 | 63% | | 57% | | 60% |
| Group 10 | | 61% | | 64% | 62% |
| Mean | 60% | 67% | 55% | 65% | 62% |

**Table 5.2: Unweighted kappa values for each group.**

not all, of these referents were tagged by the expert analysis. I noted the agreement between referents in a cluster as follows:

**N/A** — only one group tagged this referent

**Poor** — referents in cluster differed significantly in location, details or type

**Fair** — referents in cluster had similar details and the same or similar type

**Good** — referents in cluster had nearly identical details and the same type

For example, in the first transcript, all five groups found a reference to a referent involved in drawing the figure's stomach. The reported referents and details about them are shown in Table 5.3. Here, agreement is quite good — all five groups agree on what the references refer to, but some give the referents differing types. Two settle on the "functionality" type, while the other three groups are split between "instruction", "output", and "bodypart" (a new referent type invented by group 0). In response I note the two with "Good" agreement, and the others with "Fair", because of the difference in type.

| Agreement | Group | Line | Ref type | Details |
|---|---|---|---|---|
| Fair | 0 | 1 | bodypart | stomach |
| Good | 1 | 1 | functionality | stomach |
| Fair | 4 | 1 | instruction | stomach |
| Fair | 5 | 1 | output | stomach |
| Good | 7 | 1 | functionality | stomach |

**Table 5.3: Fair to Good agreement on the "stomach" reference.**

In contrast, Figure 5.4 shows a case where the groups have some significant disagreement. Here, most of the groups have decided that the lines 40 and 41, "read the left side [...] it tells you", indicate a proposal of a plan. Three of the groups identify the first reference to this referent as occurring on line 40; however, group 7 puts it on line 41. For this situation, I marked their version with "Fair" agreement. Group 2, however, tracks this as a Repair referent, hearkening back to an earlier mismatch of common ground. While this may be a valid interpretation, for the purposes of determining inter-coder reliability I marked it as having "Poor" agreement with the other referents in the cluster.

| | Line 40 — user A: read the left side<br>Line 41 — user A: it tells u | | | | |
|---|---|---|---|---|---|
| Line | Cluster | Agreement | Group | Ref type | Details |
| 40 | 40a | Good | 1 | plan | read instructions on Ref–9a |
| | | Poor | 2 | repair | clarifying Ref–38 |
| | | Good | 3 | plan | read Ref–40a |
| | | Good | 9 | plan | read left side |
| 41 | | Fair | 7 | plan | read left side of instructions |

**Table 5.4: Fair to Poor agreement involving the referential phrase "read the left side".**

However, not every group found a referent for each referent cluster. Of the 172 referent clusters identified across the four transcripts, only 13% (23) were found by all five groups analyzing the appropriate transcript. In fact, 34% (59) referent "clusters"

were made up of a single referent, found by only one group. These unary clusters represent the difficulty with agreeing to and maintaining a consistent level of detail for the analysis. A particularly thorny example of such a disagreement is shown in Figure 5.5. Here, the groups have examined the utterance, but have decided on differing levels of detail.

Line 33 — user B: so we have to draw a man?
Line 34 — user B: feet, legs, torso, arms, head and face

| Line | Cluster | Agreement | Group | Ref type | Details |
|---|---|---|---|---|---|
| 33 | 33a | Good | 1 | functionality | draw a man |
| | | Good | 2 | functionality | draw a man |
| | | Good | 3 | functionality | draw a man |
| | | Fair | 7 | functionality | man is made of feet, legs... |
| | | Good | 9 | functionality | draw a man |
| | 33b | N/A | 1 | instruction | draw a man |
| | 33c | N/A | 1 | plan | implement REF–33A |
| 34 | 34a | N/A | 9 | repair | REF–33A is feet, legs, torso, arms, head and face |
| | 34b | Good | 2 | functionality | feet |
| | | Good | 3 | functionality | feet |
| | 34c | Good | 2 | functionality | legs |
| | | Good | 3 | functionality | legs |
| | 34d | Good | 2 | functionality | torso |
| | | Good | 3 | functionality | torso |
| | 34e | Good | 2 | functionality | arms |
| | | Good | 3 | functionality | arms |
| | 34f | Good | 2 | functionality | head |
| | | Good | 3 | functionality | head |
| | 34g | Good | 2 | functionality | face |
| | | Good | 3 | functionality | face |

**Table 5.5: Differing levels of detail meant that some analysts tagged different referents.**

In this example, the two users are discussing what they must do to complete the assignment. One posits an interpretation of the instructions ("So we have to draw a man?"), and then clarifies with specifics ("feet, legs, torso, arms, head and face").

The analysts chose a number of different levels of detail when tagging this pair of utterances. Group 1 finds three referents on line 33 — the program functionality to draw a man, the instructions telling the users to draw a man, and the plan proposed by user B to implement the instructions. This is contrasted with the much less detailed approach taken by the other groups, who tag only the first of these referents as the most salient — the functionality. It is interesting to note that Group 7 includes details from the next line ("man is made of feet, legs....") in their referent description, which is an incorrect application of the method; in general referents should not be forward-looking. In any case, Group 1's tagging leaves two lone referents, each in their own cluster.

This situation continues on the next line. Group 1 tags this line as a repair — a continuance of the previous utterances, meant to head off misunderstanding. Groups 7 and 9 tag no other referents in this line, treating it as below the level of detail they have selected to capture. However, Groups 2 and 3 both tag the nouns in this line each with its own referent; these groups have decided on a finer level of detail than the other groups. This leads to five referent clusters, each with a population of two.

This sort of difference in level of detail for the analysis leads to imperfect agreement between groups. However, overall the agreement observed was quite good for non-expert coders examining a noisy domain. Given that the student groups were given incomplete transcripts, were novice analysts, and were presented the analysis task as a homework assignment with unclear goals, I feel that these results are satisfactory. I am confident that in application by more experienced analysts, and with a more thoroughly specified goal for the analysis, analysts would be able to converge on roughly similar levels of detail and tag the same set of referents.

## 5.2 Domain independence

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.

The students worked in groups of two. Each group proposed a target domain centered around use of the GREWP tool by a dyad. These domains, as shown in column one of Figure 5.6, ranged from trip and wedding planners to a collaborative coding domain to co-construction of a web page. Once a suitable domain was chosen, students designed and conducted experiments using students from the campus at large as test subjects. Each group was required to collect about 10 hours of data across a few sessions, using at least three separate dyads. The groups then performed an analysis of one of the experimental sessions — generally 1–2 hours of data — using the methods they had been taught in class. Finally, they were asked to design a new system, based on GREWP, which would better support the collaboration seen in their data collection.

All of the groups were able to successfully motivate their system redesign using these methods. As summarized in Figure 5.6, every group found recurring patterns of coordination and recurring errors in the interaction, and used these observations to justify and shape their redesign. In a few groups, the students also identified the creation of secondary structure by the users. I expect that due to the relative inexperience of the analysts, and the short time available to teach the methods to students. Nevertheless, about half of the student groups were able to further refine these design ideas by pulling inferences from the referential structure analysis of their

| | Recurrence analysis | | | Referent analysis | |
|---|---|---|---|---|---|
| Project | Recurring coordination | Recurring errors | Secondary structure | Referent types | Referent 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 |

**Table 5.6: Methodological rationales used by student groups for redesign.**

data by making assumptions based on the referent types they identified. Most of these groups employed the full method, computing and comparing various measures (such as referent lifetimes and density of mentions) derived from their data. In the next few sections I will examine these results in greater detail.

### 5.2.1 Using recurrence analysis for redesign

#### Recurring coordination

Looking at the rationales for redesign presented by the students (Figure 5.6), I 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 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.

### 5.2.2 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 referent 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 referent types and new representations are summarized in Figure 5.7.

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

**Table 5.7: Students designed new representations based on finding new referent types.**

Only three groups actually drew conclusions based on the statistical analysis of referent data — i.e., lifetime, density, and so forth. I 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 referent 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 referent 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" referent. From these insights they were able to design a shared representation — a budget calculator — that they felt matched the access characteristics of their data.

## 5.3 Summary

The results from the HCI experiment showed that the methods could be taught to and applied by novice analysts. The structured approach to redesign that the analysis methods provides was very helpful for these novice analysts. By following the straightforward procedures outlined for investigation and analysis of a domain, and the recommendations for redesign generated by them, these students were able to produce better redesigns than their counterparts in previous classes which were

not taught the methods.

Additionally, the work these students did to apply the methods to a wide variety of domains, admittedly within a narrow interaction framework, demonstrate the ability of the method to be generalized beyond the VesselWorld domain it was designed for. Finally, the good agreement between groups analyzing the standard transcripts showed that the methods have a measure of reproducibility.

In the final chapter, I will outline the implications of these new analysis methods, and examine some future directions for the methodology.

# Chapter 6

# Generating Design Recommendations

In this chapter we describe the technical aspects of determining the features of information flows, and the process of using these to select appropriate representations to improve user performance. To do this, we needed to construct a (partial) language for describing discourse features, identify properties of various representations, and provide a way to match these two lists.

## 6.1 Representation

In an ongoing cooperative activity, sharing context and expectations is crucial to maintaining coordination. Participants in a recurring activity habitually create and participate in conventionalized structures for behavior [8, 45, 111, 114]. These structures simplify interaction by creating expectations in other participants. Coordinative structures such as these function as conventions [80, 26], community-specific solutions to recurring problems in coordination of talk or action. 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 are a societal solution for providing participants with mutual expectations for behavior.

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 [6]. 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 participants realize conventions for behavior is by generating *secondary structure* in the discourse; this serves to organize participants' 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 [113]. 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 maintain-

ing the accuracy and consistency of that list, provides helpful structure to simplify coordination of shopping for groceries. These external representations can serve both to encode expectations about a situation, and store information intrinsic to the task itself.

### 6.1.1 Structured representations

If this impromptu structure, and the procedures surrounding it, proves successful — that is, if it serves 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. Structures which simplify the coordination of a conventional behavior can be codified into artifacts, whether conversational, procedural, or instantiated as physical objects. 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.

Past work has examined the role of external artifacts in coordinating interaction (e.g., [59, 118, 157]). These coordinative artifacts are a special sort of artifact [144], in that their purpose is to simplify the team work of participants, rather than to directly affect the task.

*Coordinating representations* [6, 127] 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. 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.

For example, a stop sign creates expectations in the participants of a joint traffic activity, but does not determine activity completely nor directly aid participants in their driving activity. 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, much the way a notebook serves to ease the burden of remembering information [99]. It also alters the task to make problem-solving easier; for example, "complex sheets" help airport baggage handlers align multiple sources of information [128]; by distilling information from different sources into a single representation, the complex sheets vastly simplify complex cognitive processes.

On the surface, it would seem to suffice to introduce new representations mediating all possible information flows, and allow users to find their own uses for them. However, merely providing additional representations to participants in a joint activity can cause problems. Too many representations can overload the user, providing too many choices and occupying too much real estate, both in the mind and in the interface. Also, because the view of a joint activity necessarily differs from person to person, participants must continuously align their private representations during the activity to the extent necessary for the activity to be successful. 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. More representations offer more opportunity for misalignment, and can create extra team work for participants.

It is necessary to therefore examine the complete *representation system* available to participants to model the interaction. 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.

By examining these aspects of the interaction system it is possible to understand and predict the impact that it will have on an ongoing work practice.

### 6.1.2 Representation properties

Different representations of information have different properties. As information is mediated by different representations, it necessarily acquires different characteristics. Hutchins and Klausen [60] write:

> "Notice also that the various media in which information is represented have different properties (Norman, 1993). Speech is ephemeral. It requires one to attend to information at the time it is delivered. Representations in the memories of individuals endure longer than those in speech. [...]

> Finally, a portion of the information... was imposed on the airplane itself, in the tuning of the radio. This is the same information that had been represented verbally, but now it is in a relatively durable representation, because the setting of the radio is continuously available and will not change until the next frequency is tuned."

If representations impose certain properties — ephemeral, durable — on information propagation, then it is likely that there are representations that are better suited to mediating particular types of information than others. 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.

For these participants, the perceived cost to transcribe the information to the whiteboard is outweighed by the potential benefits. The features of the whiteboard — that information in it is persistent (in the short term), visible to all, in color, graphical (as opposed to purely textual), and so forth, make it suited for particular types of information. The choices that participants make about how to record, transcribe, etc. information are at least partly influenced by the principle of Least Collaborative Effort [27]; 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.

## 6.2 Modeling information

Creating new artifacts which aid coordination may prove quite difficult in practice, because the form of the interaction is necessarily changed by the introduction of a new

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, because they do not match the emergent work practice of the interaction they are introduced into.

In Hutchins' landmark paper on distributed cognition [59], he detailed how "speed bugs" reduced pilot effort. These artifacts are small indicators mounted on the perimeter of the air speed indicator in the cockpit which point to a specific speed. Use of these moved information about safe landing speed from a textual representation to a graphical one, reducing pilot effort and error by transforming an internal calculation to a visual operation. Similarly, representations such as checklists structure activity by providing persistent feedback about the steps remaining in a task and explicitly ordering steps according to implicit dependencies between the steps.

The goal, then, is to provide a representation system that reduces the collaborative effort of the group. To accomplish this, it is necessary to produce a model of information within a system that adequately provides the ability to recommend matching representations.

### 6.2.1 Information flows

A crucial abstraction for this model is that of information flows. Based on the root concept from distributed cognition, my definition of an information flow is a directed transmission of a specific type of information between a pair of representations for a particular task. Information flows therefore depend on the representational structure of origin and destination, the type of information being sent, and the purpose of that information. This means that there may be multiple information flows between the same origin and destination even for the same type of information, if the task is different. This very fine-grained view of information flows is useful for selecting

proper representations.

An example is the transmission of information between a pocket watch and the internal representation of its owner. In some cases, the watch user might look at the watch a number of times in a row, each for a different task. (I say pocket watch here, and assume that it only shows the time; the date area usually should be considered a separate representation. And don't get me started on multi-function watch faces.) If the user believes he is late for a meeting, he might check the watch with the goal of discovering whether or not this is true. A quick check of the time, some rapid mental comparisons, and the person ascertains that, no, he is not late for the meeting — but forgets the actual time, as that was not the original goal.

He may then immediately look at the watch again, to see how much time is left before the meeting, and note that he has fifteen minutes before the meeting. However, he still may not retain the absolute time, as that was not the goal; if someone then asks what time it is, he may have to look at the watch a third time, and this time retain the absolute time, to fulfill this new goal.

In this scenario (which has indeed happened to me), three separate information flows have occurred one after another, from the same source and to the same destination, but differing in their purpose and informational content (in the first case, the simple boolean 'No'; in the second case, a simple number '15', and in the final case a more complex 'time' item). This distinction is crucial to understand for a designer of a properly working system; it is quite common for the same representation to present information which can be used in a variety of ways at once, although these alternate uses might be difficult to understand without thorough domain analysis.

This abstraction gives us a way to apply the observations from referential structure analysis. The trick is to identify the information flows within a system, and match referent types to them. Using the venerable "burger joint" example, there is

an information flow which is established from customer via cashier to cook regarding the burger order and containing the burger information. This flow corresponds to referents of type 'order' in the analysis performed in Figure 4.1 (on page 72). By examining the referential structure of an information type, we can model the information flow itself.

### 6.2.2 Information features

Our goal is to provide representations based on how humans are affected by the characteristics of these information flows. Rather than addressing flows on a case-by-case basis, we have come up with a general language to talk about characteristics of flows and the information stored within them. We have identified eight features of information which are relevant to representation choice. For items within each information flow, we examine:

- lifetime of relevance — how long an item is typically relevant
- frequency of appearance — how frequently the item is referenced
- simultaneity — how many items are relevant at the same time
- creation frequency — how often new items are created by the user
- read frequency — how often the user gets info about an item
- write frequency — how often the user changes info within an item
- update frequency — how often sources other than the user change an item
- information complexity — difficulty in remembering items of this type

For each information flow, these features are measured on a (subjective) sliding scale from "Minimal/Never" through "Continuous/always". These scales are mea-

sured relative to the time scale of the task which requires the information, as that is the context in which the measurement is meaningful. A summary appears in Figure 6.1. These measures are aggregated across all the observed items of a particular type. It has been our observation that items within a type have low variance across these measures; indeed, high variance in a feature — a range of values that spills into more than two value categories, as discussed below — is a good indicator that the information flow has been misidentified, and needs to be broken into two or more separate flows. (The discovery that location referents should be split into deictic and definite groups was an example of this.)

| | Minimal / Never | Low | Moderate | High | Continuous / Always |
|---|---|---|---|---|---|
| **Lifetime of relevance** | Only persists for one operation | Persists less than one task | Persists throughout a task | Persists across tasks | Always present |
| **Frequency of appearance** | Referenced less than once per interaction | Referenced less than once per task | Referenced about once per task | Referenced many times during a task | Referenced all the time |
| **Simultaneity** | Singleton | Items can occasionally co-exist | Items routinely co-exist | Many items routinely co-exist | Very many items |
| **Creation frequency** | Created by another user | Created less than once per task | Created about once per task | Created many times during a task | Created continuously |
| **Read frequency** | Only read by other users | Read less than once per task | Read about once per task | Read many times during a task | Read continously |
| **Write frequency** | Cannot be altered | Altered less than once per task | Altered about once per task | Altered repeatedly during a task | Altered continuously |
| **Update frequency** | Never modified by other users | Rarely modified by other users | Occasionally modified by other users | Frequently modified by other users | Continuously modified by other users |
| **Information Complexity** | Info that is only useful in aggregate | Trivial information; boolean, existance | Simple information: shape, short number | Moderate information: name, phone number | Complex information: long number |

**Figure 6.1: Scales for observed information features.**

For example, update frequency varies widely in the case of the hands of an analog watch. In this case, the time-frame of the typical task, "Find out what time it is", is on the scale of seconds. In this timeframe, the second hand is updated very frequently, while the minute and hour hands have meaningful updates much more slowly, and a day indicator has minimal updating. For this reason, we find it convenient to store the date in representations which have a significant cost for updating content, such as a day-by-day desk calendar — whereas it would be quite unusual for the current second to be stored in a representation which required tearing off a sheet of paper with each

update. With a different task — say, maintaining a calendar for a multi-year project — the same representations might be classified differently. In the context of a long task, a daily update might occur frequently, while a second or minute representation might considered to be updated continuously, and need to be abstracted out of the presentation to the user.

Referential structure analysis can be used to help establish values for each discourse feature for an information type. For example, lifetime of relevance is one of the basic measurements that the analysis will produce. Other values, such as read frequency, can be obtained by examining each reference and determining its communicative purpose (create, read, write, and/or update), and then counting. Information complexity presents more of a challenge; the analyst will need to investigate the potential subunits of a piece of information, what kind of values are possible for each, and how the participants store the information internally, before assigning a complexity to a piece of information.

In the burger example, above, we would classify the lifetime of relevance of the "order" referent as "moderate" — the task under consideration being delivery of the order. The frequency of appearance is "high" — the order was referred to frequently during the task. Simultaneity is impossible to judge in this isolated example, but further analysis would probably reveal it to be "moderate" for the cashier (who often has multiple orders pending delivery), and "moderate" for the cook (who often has multiple orders cooking).

Creation frequency is "moderate" for the cashier, who creates one order per task, but "never" for the cook — the order is handed to the cook in a complete form. Read frequency is "never" for the cashier in this example — in practice it might be classified as "low" to account for erroneous orders which the cashier must re-read — and "moderate" for the cook, who must read the order, and occasionally refer

back to it. Update frequency is generally "never" for both, in most cases, though in some cases the cashier might modify the order after creation; again, for flexibility of design, an analyst might therefore assign this a "low" value. Finally, the information complexity is "moderate" to "high" – both cashier and cook have established jargon for these common orders, which reduces the effective complexity, but a complex order can still contain many pieces of information.

In the next section, we will establish formal definitions for these features. In the following section, we will show how these features can be used to drive the redesign process.

### 6.2.3 Defining information features

This section examines each feature and gives some definition and exploration of the difficulties of definitions within this framework. Giving complete examples for all the regions of the high-dimensional feature space is impractical, but we have endeavored to provide illustrative examples in places where the concepts are unclear.

#### Lifetime of relevance

Lifetime of relevance is the duration between the first access to and the final access of a piece of information. Note that this does not necessarily mean that the human must store (remember) this relevant information internally; rather, the system of humans plus artifacts stores the information for its lifetime of relevance. Using our methods, this is determined within an ongoing collaboration by simply looking at the time between the first reference to a referent and the last.

The slippery concept in this case is 'relevance'; here we use it to mean that the information is useful or required for the current activity. Information may go into long-term storage, whether in long-term memory or molding away in a locked filing

cabinet in an unlit basement somewhere; in such a case, its lifetime of relevance for the current task has been exceeded, but as a part of a larger activity it might still be relevant. For this use, we are interested in the relevance of the information for the task which defines the information flow being examined.

By way of example, a printed grade sheet stores information for (at least) two activities. The first is to notify the student of their grade, and for this purpose the lifetime of relevance is rather low. In fact, a print-out could be considered too permanent a representation for this simple task; a word from the professor would be all that was required. However, the report card also serves the purpose of communicating that information to the professor at a later date; for this purpose, the persistence of the printed record is necessary, as the information needs to remain relevant and accessible for a long duration. From the point of view of the professor and his task of tracking grades, each student grade has a high lifetime of relevance, beginning with determination of that grade and (possibly) ending with transcription of that grade onto the student's permanent record, at which point the professor's formal interest in the information is reduced.

With some information, this lifetime is easy to track; for example, in VesselWorld, a barrel of toxic waste has a strong identity. It does not change into another referent, generally, and so it is easy to identify when it is first mentioned and when it is no longer relevant. In contrast, some information is rather polymorphic and can change essence during interaction. For example, a proposed plan may undergo drastic or complete changes during negotiation, making it much more difficult to point at exactly where that plan ends and a new one begins. In this case we fall back on the ideas of 'topic' and 'conversational frame' from the literature. We examine the lifetime of the information by regarding the lifetime of the conversational topic surrounding it; while users are negotiating, or when they resurrect an interrupted discussion anew,

this plan is relevant; if discussion means the plan is relevant within a new topic, then it is a new plan.

### Frequency of appearance

Frequency of appearance counts how often an item of information of this type appears in the discourse. The more frequently a referent of this type appears, the higher this measure. At the extremes, a dialogue where every exchange of information contains a new referent of this type would have a frequency of 100%. In practice this very rarely occurs; generally there are a handful of information types that dominate the discourse, with a larger number of infrequently-appearing types.

This measure can be used to guide development effort. Supporting information that appears more often will generally have the largest payoff. In practice, our method involves calculating this measure, sorting information types by it, and working on the most frequent types first.

### Simultaneity

Simultaneity is a measure of how many items of a particular type are generally relevant at the same time. For example, a typical pocket-watch user has one 'time' item relevant; an operator in an international call center, with six clocks on the wall set to different time zones, would have multiple 'time' items relevant simultaneously (with representations to match).

This measure does depend on being able to quantify information, so as to be able to talk about informational 'items'. This is not always possible, as some information comes in continuous streams. However, in general, continuous information (e.g., the speed of a car) can be treated as a single piece of information which is being continuously written to or updated by outside sources; such information would have minimal

simultaneity.

Increasing simultaneity must be supported by stronger and stronger abstractions and tools so that the user is not overwhelmed by the information. One way to do this is to group information, provide a way to search into it, or filter it in other such ways. Another method is to provide proxies of lesser complexity which the user is better able to comprehend in aggregate.

### Information complexity

Information complexity is a very useful measure nevertheless fraught with inherent difficulty. As used here, it describes the storage space in a human's internal representation that is required to meaningfully store information for retrieval later. This is related to, but not identical with, computer science measures such as Kolmogorov Complexity or algorithmic information content [46, 81]. These definitions, while useful to system designers concerned with bandwidth issues, are less useful for human-interface designers. As the limiting factor in practice is usually human, it is better to examine this in terms of memory requirements and the work necessary to understand, remember, and express a piece of information. Humans are very capable at chunking information so as to be able to store more and more information effectively; and so, this measure must really look at information as stored by humans. For that reason, the scale is centered around the ability of a human to store the information in short-term memory, as defined by research such as the Model Human Processor.

At the simplest level are simple, single-value types of information: booleans, remembering the existence of an item. Just above this are simple items: a common word, shape, digit, and the like. Humans and computers are both excellent at recognizing and expressing digits and short words. In contrast, humans and computers diverge when it comes to recognizing, storing, and expressing complex data such as

long sequences of words or graphical information; humans can in some cases (e.g., song lyrics) recall long sequences of words, and recognize complex graphical information of certain sorts (faces) instantly, but in general computers are better at storing such data, and are generally much better at expressing it. Most people cannot accurately draw even a well-known face from memory; for a computer, displaying a stored picture is trivial. Hence it is necessary to take into account this disparity when determining effective information complexity; this effective complexity is dependent on the operations that will be performed on the data.

### Creation, write, update, and read frequency

Informational items (in a flow where information comes in atomic chunks) are usually created, often read by the user, potentially written to by the user or updated by other users or the system, and finally become irrelevant and are forgotten or otherwise removed from the system. The frequency with which each of these actions occur reveals a great deal about how the information is used by the system. We track events of each type (except deletion/forgetting, which is difficult or impossible to observe) to determine values for each of these features.

Examining the frequency of each of these transitions reveals a great deal about use of the information and how that information should be represented. High creation frequency indicates the need for simple creation procedures; low frequency allows the use of more cumbersome procedures if such are required by more pressing design constraints. Increasingly high read frequencies imply that more and more of the user's attentional space be devoted to representing this information, or that the cost of polling the information flow is otherwise reduced. Continuously-read information — for example, lane position during highway driving — needs to be omnipresent; requiring the user to take action to acquire the information will be seriously detrimental

to task performance. Conversely, information which is read less frequently can be stashed out of the way (attention-wise) until needed; even though speed is frequently accessed during highway driving, it suffices to devote a small portion of the user's sensory surround to a readout for it. Rarely used information, such as tire pressure requirements, can be stored in inconvenient locations.

Similar conclusions can be drawn from examining write frequency. Low write frequency implies that the procedure used for writing can be somewhat slow and cumbersome, if such is necessary to fit other constraints. Frequently-written information must have a streamlined procedure, and continuously-written information requires an automatic method for storing data. External update frequency must be inspected in conjunction with read frequency and task modeling to determine design constraints. If the updates are meaningful, and need to be brought to the attention of the user, then update frequency can be treated similarly to read frequency; if updates are unimportant, then notification of them can be disconnected from frequency.

## 6.3 Recommending representations

Once information features are identified the analyst can use them to narrow in on appropriate representations. I have established general guidelines for design recommendations implied by various values in each feature; these are summarized in Figure 6.2. This table provides a set of prescriptive guidelines for design based on observed use of information.

Carrying forward the "order" referent from the "burger joint" example, from the cashier's perspective, we found that it had a moderate lifetime. This indicates it probably needs a representation with persistence, though it is not a certain thing. In practice, we see examples of both – some burger joints use language (a non-persistent

| | Minimal / Never | Low | Moderate | High | Continuous / Always |
|---|---|---|---|---|---|
| **Lifetime of relevance** | Should be stored in volatile rep | ← sliding scale → | | | Should be stored in persistent, available rep |
| **Frequency of appearance** | Design is unimportant | ← sliding scale → | | | Design is critical |
| **Simultaneity** | Does not need to be referenced | No extra tool needed; user can track all items | Need strong naming scheme for items | Need search/filter mechanisms | Too many to think about: provide abstraction |
| **Creation frequency** | No creation procedure needed | Can require lengthy creation procedure | Should have fast creation procedure | Must have fast creation procedure | Paradigm is broken: redesign |
| **Read frequency** | Does not need to be presented after creation | Can require lengthy access procedure | Should be easy to read | Must have rapid reading procedure | Must have dedicated sensory area |
| **Write frequency** | Use an unmodifiable representation | Writing allowed to be slow | Should have decent writing procedure | Must have rapid writing procedure | Must have automatic writing |
| **Update frequency** | Needs no mechanism for change awareness | Should alert user of change | Might alert user of change | Should not alert user of change | Do not alert user of change |
| **Information Complexity** | Info should be aggregated | Could be stored in STM | Should not to be in STM | Cannot store in STM | Cannot store in memory; should be rechunked |

**Figure 6.2: Design recommendations stemming from specific levels of individual information features.**

representation) to communicate this information, while some fast-food joints use a persistent though short-lived computerized representation.

The high frequency of appearance indicates that a designer should spend significant time addressing the design of this representation. Moderate simultaneity indicates the need for some sort of naming scheme or other way to differentiate items; in practice, waiters often keep this information in their head, but in structured environments such as a fast-food restaurant, orders are assigned tracking numbers for this very reason. Moderate creation frequency indicates that having a rapid creation procedure is preferable, but a slower one — such as writing notes on a slip of paper — is acceptable. (Note that, as the creation and write frequencies for the cook are 'never', most cooks don't even have a pencil!) Finally, the moderate to high information complexity points to the preference for some sort of external representation — the order slip provides just such a representation.

Given this set of desired properties, the next step is to identify a representation that embodies them all, while also satisfying other constraints. For example, as the same representation is going to be used (albeit differently) by both cashier and cook, the representation's properties must match both sets of features.

I have begun the process of identifying a set of representation properties. The next section will discuss the handful of properties identified so far.

### 6.3.1 Representation properties

Representations possess specific properties regarding their storage, mediation, and presentation of information. As with information flows, these properties depend not only on the representation, but on the purpose it is being put to. For example, a piece of paper used to pass notes in class has properties (notably, it is updatable) that the same piece of paper used to take notes in class lacks. Hence, when assessing representations, we must consider not only their structure but how they will be used.

Representations combine a number of properties in a synergistic manner. On first blush, examining specific properties of a representation individualistically seems counter-productive. However, although their combination may have side effects that affect how users interact with them, the root properties of a representation primarily determine how it is best used.

For example, a stop sign / stop line combination serves at least three purposes: first, it sets the situational frame for interaction with other users, creating expectations about others' actions. Second, it provides a visual indication of where previous experience has determined a car should stop to allow for clear and safe passage of traffic. Finally, it serves as an salient visual indicator of an intersection, which might not otherwise be visible.

The properties that allow these purposes — including visual salience (determined by placement, coloration, and distinctive shape), clear demarcation of position (the stop line serves as an unmistakable visual indicator of location), and visibility (usually, all actors can see all stop signs and know this, allowing meta-level conclusions about the knowledge of other actors) — work synergistically. If the sign were not visually

salient, the demarcation of position would be immaterial, as the stop line could simply be another marking on the pavement. Likewise, the visibility of the stop sign would be in question, as other actors could not be counted on to note the sign reliably. Nevertheless, despite their interdependencies, we can assess each of these properties individually, and examine their impact on the way actors make use of representational artifacts.

### 6.3.2 Sample properties

We have only identified a handful of representation properties to date. However, there is abundant literature regarding the cognitive impact of various representations, and the properties of these. We continue to expand our list, and add examples of each property to elucidate exactly what it is. One good starting point is the list of communication features generated by Clark [27, 26]. He identifies eight specific properties of an interaction, which I have attempted to provide usable definitions for:

- Copresence — do the participants share a common physical location?
- Visibility — are the participants visually aware of the state and actions of others?
- Audibility — can the participants hear each other?
- Cotemporality — are all participants taking actions and responding in real-time?
- Simultaneity — can both participants communicate at the same time?
- Sequentiality — do the actions of participants have a temporal sequence which all can see?

- Reviewability — can participants access relevant information at a later date?
- Revisability — can participants repeatedly alter a shared work object?

These properties provided a very useful starting point for looking at representation properties. Using this as a basis, I identified a handful of basic properties: persistent, atomic, ordered, editable, focus, and so forth. The names for these properties are a work in progress, but so far they have served to help me redesign groupware systems successfully.

**Cost of creation, reading, writing, and updating** These fundamental properties indicate the effort required for a user to effect one of these operations on data stored by the representation. A speedometer is easily readable (the user can poll it for information) and automatically updating (the car's systems continuously change the value within the representation), but neither creatable (the user cannot create a new speed sensor on the fly) nor writable (the user cannot directly affect the readout; instead they must indirectly affect it by changing the velocity of the car). An exam sheet is readable and writable, but not updatable in any useful sense (other users do not alter the information therein). A shared text editor used to build a grocery list possesses all four properties; a user can create new items and read from or write to old ones, and items may be updated by other users.

**Persistent** Persistence is the quality of a representation that information entered into it will be available, unchanged, for recall at a later date. An example of a representation is a text file; what is written in it, barring system failure, can be accessed at a future date and will be found unaltered. This expectation allows users

to offload information into this representation for the purpose of reducing memory burdens; rather than needing to store details of the information, all that is needed is the remembrance of where to find it. In contrast, a representation such as speech (in the absence of recording) does not persistently store information; though it can be easily accessed during its production, there is no meaningful way to say that the information can be retrieved from the speech directly at some later time. This concept is closely related to what Clark terms reviewability and revisability. While it is dependent on a number of other properties — e.g., the representation must store information (as opposed to simply providing context like a stop sign) — the persistence property itself is separable.

**Atomic** This indicates a representation that can be used to store a collection of similar items, usually items of the same informational type. This indicates a representation that is useful for storing items with a high degree of simultaneity; by providing a structure which abstracts out the similarities between items while retaining their differences, a representation such as a list allows a user to keep tabs on a larger number of items than they might be able to in an unstructured representation such as memory. The Object List is atomic; the unit in that case is a barrel of toxic waste. A text document possesses little of this quality, although a user can create their own secondary structure within it by creating tables.

**Ordered** This feature implies that the representation can present information in a coherent and understandable order; such representations are generally atomic as well. The classic example is a checklist; conventions about reading order and temporal implication mean that items in a checklist are usually read top to bottom; a designer can therefore store information where the order is important within it. One indication

that this might be necessary is a high simultaneity of information, coupled with a sub-property of the information that has an inherent and applicable order. A counterexample would be a geographical map, which does not provide a simple way to present order; hence, ordered operations which involve geography, such as giving directions, require an auxiliary representation such as numbered lines on the map or a list of direction steps.

**Focus** This feature explains how users pay attention to the representation. A single-focus representation provides a single stream of changing information: the user can focus on that single point of attention and receive all the new information coming out of that representation. A checklist is structured to require a single point of focus, namely, the task at hand. Live conversation inherently has a single point of focus, at the current state of the conversation, though users may disengage from the current conversation to consider past points or plan future actions. Many representations allow updates and writes to happen throughout the representation; a private document provides no inherent focus, while the Object List provides a number of possible focus points (approximately, one per table cell). Strongly focussed applications are good for information flows which require strong coordination between updates and reads, but in turn restrict action, especially in a multi-person interaction.

### 6.3.3 Selecting matching representations

Using these representational properties as a guideline, the analyst can select appropriate representations and adapt them to the specific interaction. For example, an ordered checklist such as the one pilots use for take-off is good at supporting information which is atomic, has a high frequency of appearance within the task. The representation has a single focus and low simultaneity; each item is created once, read

once (in a perfect execution) but is available persistently (for problematic execution), written to once (i.e., checked off), and is never externally updated. The check boxes store the boolean information — "task complete" — in a very handy representation. However, the checklist itself can provide more structure; an automated checklist, which does not allow takeoff until all tasks have been addressed, would also track an additional piece of information, namely, the boolean "take-off preconditions met" that a pen-and-paper checklist cannot easily store (the users must visually assess the checkboxes to ensure all tasks are complete, and so forth).

Selecting matching representations can be a difficult process. There is a large body of available literature which examines the space of potential representations and examines their use. Research in the areas of air traffic control, ship navigation, cockpit displays, and nuclear power plant operation is relevant and very useful for vetting potential representations [2, 40, 59, 60, 110, 130, 150, 156]. Reviewing this literature from the perspective given by our analysis techniques and theoretical framework is very useful for establishing a set of appropriate representations for a particular task. A complete survey of available representations in this perspective, while outside the scope of this thesis, would provide a 'chinese menu'-like approach (one from column A, one from Column B) to assembling groupware. Such an approach is outlined for potential future work, in Chapter 8.

In the mean time, we have had success assessing likely candidates for representation with our methods. A large number of candidates were chosen using traditional design guidelines — prescriptive guidelines, prior art, common sense, and engineering constraints — and then winnowed and customized based on our analytic methods. As will be seen in the next chapter, this approach creates groupware that matches well, if not perfectly, to the ongoing work practice of participants. This satisfactory solution can then be subsequently refined, if necessary.

## 6.4 Summary

In this chapter we discussed the technical aspects of identifying information features, and gave an overview of the work we have done on identifying representation properties and selecting appropriate representations for information. In the next two chapters, we present experimental evidence for these methods. In Chapter 7, we present an experiment which demonstrates the design of representation systems which both match and mismatch the observed discourse features. Finally, in Chapter 5, we demonstrate general applicability of the methods, and show that they can be taught to and applied by other analysts.

# Chapter 7

# The Business Travel Experiment

This chapter describes an experiment conducted to test the ability of referential structure analysis to adequately extract discourse features, from there to recommend specific representations which would store that information appropriately, and predict the impact of these representations on the performance of users.

A mismatch between the features of provided representations and the observed properties of an interaction leads to increased work and increased potential for error on the part of the participants. In response to a mismatch, participants face increased work as they either try to fit their interaction into the available representation system, or try to subvert the available representations to fit their desired interaction. This chapter shows evidence for this sort of resistance and co-opting of representations when participants are presented with ill-fitting systems.

Using my methodology, I created two systems — one which matched the observed discourse features, and one which conflicted — and compared performance of groups using each of these systems. The chapter begins with a brief review of the design process, continues with an explanation of the experimental domain and software systems used, and concludes with an examination of the data produced.

## 7.1 Experimental design

A two-stage experiment was devised to test the utility of methods for suggesting new representations and predicting the impact they will have on the emergent work practice. The goal was to show that, using referential structure analysis, an analyst could predict which representations for information would be adopted successfully by users, and which would not be.

In keeping with the experimental methodology, I conducted a small study to gather initial data, and used this data to design a pair of domain-specific groupware systems. The first system, the "matching" system, was designed with a set of representations that matched the emergent work practice of the pilot study users, and were predicted to support the ways in which they shared information. In contrast, the "non-matching" system was designed with representations which were very similar to those in the Matching system but did not match the work practice of users. Results from this experiment, discussed below, show that the methods used were successfully able to predict representation use, and give specific explanations of why users did and did not use the provided representations.

The domain of business travel was chosen for its general familiarity among users. Specifics of the task were determined by conducting a brief ethnographic study of the methods used by real-world users to plan business trips. Subjects were given instructions such as the following:

> You and your partner have 30 minutes to plan a business trip to the Dallas/Fort Worth area. You are scheduled to arrive at DFW Airport at 7:44 am on Tuesday, Sept 20th, and depart from there on Wednesday night at 4:17 pm. A rental car is waiting for you. You must spend 9 to 5 on Tuesday at the Dallas Convention Center. Stay within a $500 total budget. You should produce a detailed itinerary with times and budget to present to your support staff. Your tasks:

1. Find a hotel.
2. Find places to eat meals.
3. Find entertainment for Tuesday evening.
4. Find a place to play golf on Wednesday.

A small study was conducted (n=4) to gather the required data to design the systems. Pairs of subjects were asked to create an itinerary, including budgeting, for a two or three day business trip over the course of a 30-minute problem solving session. Subjects were given a private text editor, a chat client, and a web browser, and trained in the domain before being asked to plan a trip. Dyads generally nominated one member to construct the requested itinerary in the text editor; a sample appears in Figure 7.1.

Adam's Mark Hotel $129 x 2 = $260 + tax
(google: hotels dallas)
Food:
Tues Lunch: near convention center ($30 each, $60 total)
Tues Dinner: bobssteakandchop.com (expected $160 total)
Tues Entertainment: random show 2tix x $45 = $90
Weds Breakfast: at hotel ($30 each, $60 total)
Weds Lunch: at Park: barbeque, $40
Sandy Lake Park – minigolf, $2 entry, $2/game: $12
- - - - - -
$682

**Figure 7.1: Itinerary produced by a test subject in the Business Travel study.**

### 7.1.1 Analyzing the base group data

Subjects reported some frustration with the task — organizing information was difficult, as was maintaining awareness of the actions of the other user. Despite an attempt to enforce the 30 minute deadline, no groups were able to complete the problem in under 45 minutes, with 50 minutes being the average time required.

Chat data from the base group was tagged using referential structure analysis. As a part of this process, I identified a number of new referent types: tasks, instances of tasks, and URLs. Some of the more generic types found in previous analyses (plans, repairs) also appeared in this data set. The most frequently-occuring referent types found as a result of this analysis are shown in Table 7.1.

**Tasks** are the generic tasks that users discussed. Finding a hotel, looking for entertainment, and calculating the budget were all considered tasks. **Instances** are specific places or events which the users find to fulfill a particular task; for example, the "Adams Mark Hotel", a hotel found by multiple groups, was an instance which satisfies the task of finding a hotel. **URLs** are just that — references to specific web pages made by users. While there were other referent types found, these three new types, together with domain-independent referent types **plans** and **repairs**, accounted for a vast majority of all referents found (85%), and so I focused my attention on them.

| Type | Freq | Refs | % of Refs | Lifetime | Density |
|---|---|---|---|---|---|
| Instance | 31% | 4.7 | 38% | 12% | 66% |
| Task | 17% | 4.7 | 21% | 43% | 24% |
| Plan | 17% | 1.9 | 8% | 3% | 80% |
| URL | 11% | 1.2 | 4% | 1% | 90% |
| Repair | 8% | 3.4 | 7% | 2% | 98% |

**Table 7.1: Referential structure data from the Business Travel domain.**

For each type of referent I calculated a variety of measures (see Table 7.1): the *frequency* of the type (number of referents of that type, divided by the total number of referents); the average number of *references* to each referent of that type; the *percent of all references* which were references to referents of this type; the average *lifetime of referents* of this type (number of lines of chat between first and last reference to a referent, divided by the total length of the session it appeared in); and *average*

*density* (what percent of lines over a referent's lifetime contain a reference to it).

I also calculated the concurrence of each referent type, shown in Table 7.2. This is the average expected number of referents of that type that are relevant at any one time. This was computed by noting how many referents of a type are relevant during each line of dialog (i.e., have both a reference before or on, and on or after, that line). As an average did not seem to adequately express the characteristics, I computed a number of additional statistics: the mode of the number of concurrent referents, and the maximum number of referents that were relevant at once, for each type.

| Type | Mode | Max | Average |
|---|---|---|---|
| Instance | 1–3 | 3–6 | 2.34 |
| Task | 3–6 | 4–6 | 3.24 |
| Plan | 0 | 1–2 | 0.19 |
| URL | 0 | 0–2 | 0.55 |
| Repair | 0 | 1 | 0.07 |

**Table 7.2: Number of concurrent referents for each type.**

## 7.2 Drawing conclusions

This section discusses the utility of these measures as they apply to each type of information. By examining the statistics shown above I was able to come up with a desired set of features for representations crafted to match the pattern of interaction surrounding observed referent types. Representations were therefore designed with features that matched the observed properties of information use, summarized in Table 7.3.

**Tasks** have the longest lifetime, at about 43% of a problem session. For such a long-lived referent type, they have a surprisingly high average density (24%); hence, it is unsurprising that their average concurrence is also high (3.24). This means that

in general users are working on three or four tasks at any one time, with some spikes to six. This long period of relevance, coupled with the number of simultaneously relevant referents, clearly indicates the need for a persistent, shared representation which allows users to enter at least six and preferably closer to ten items.

Closer examination of how users talked about task referents revealed properties of tasks. Users initially spent some time discussing assignment of the tasks (e.g., "You find a hotel — I'll work on food."). However, after this brief discussion, references to tasks only occurred in the context of instances: nominating instances for a task ("How about the Marriott?"), or querying or reporting on the status of the task ("Do we have a hotel room yet?"). In other words, generic tasks had only a short lifetime outside their attachment to instances. This negotiation over division of labor can therefore be handled in chat, and the remaining activity about tasks can be combined into the representation for instances, which I will discuss next.

**Instances** were the most talked-about referent — accounting for over one third of references — and of reasonably long lifetime. Groups generally had between one and three instances relevant at any one time; one group had six relevant for a fairly substantial period. However, the long tail at beginning and end of sessions, where most instances were not relevant, reduced the average number of relevant instances to just over two. From this, I determined that a representation meant to store instances should allow at least six items, and probably more, to be stored at once, even though in general only a third of the representation will be in active use.

Users gave short-hand names to instances, and struggled with conversationally pointing at instances, especially when multiple tasks were relevant; hence, the representation should provide a naming field, to allow canonical references to items. Likewise, users spent a certain amount of time discussing the cost of an instance, and more time calculating the total expenditure (as required by the task specification). A

representation for these therefore should include a way for users to input cost information in a structured fashion, so that the system can automatically calculate totals; this will reduce cognitive workload, reduce conversation, and reduce incidence of error by offloading budget computations.

| Type | Observed properties | Representation features |
|---|---|---|
| Task | Long-lived | Persistent, Nameable |
| | Frequent references | Dedicated screen area |
| | Coupled with Instances | Merge into Instance rep |
| Instance | Medium-length | Persistent, Nameable |
| | Frequent references | Dedicated screen area |
| | Moderate concurrence | Show multiple items |
| | "Budget" property | Structured storage for this |
| Plans | Short-lived | Ephemeral representation |
| | Almost no concurrence | No need for multiple items |
| | Few mentions | No need for naming |
| URLs | Short-lived | Ephemeral representation |
| | Very few references | Ephemeral representation |
| | Coupled with Instances | Treat as Instance property |
| Repairs | Short-lived | Ephemeral representation |
| | Very high density | Store in negotiable medium |

**Table 7.3: Mapping referent type properties to desired features of new representations.**

**Plans** occurred fairly frequently. However, as had been seen in previous domains, they had a very short lifetime (averaging 3% of a log file, or 3.6 lines of chat). Reference patterns were split; about half the time, a plan was proposed or reported once and never mentioned again. The other half of the time the plan was referred to a handful of times, indicating some negotiation. Only in a few cases did a plan continue to be relevant for more than five lines of chat. Plans almost never overlapped — in one case, a group discussed one plan while hashing out another, but otherwise only one plan at most was relevant at a time. Coupled with a high density — 80% — these statistics led to the conclusion that there was no need for a persistent, shared repre-

sentation for plans: chat, with its ability to be used as both a tool for announcement and for negotiation of plans, was the best medium.

**URLs** were used by some groups to refer to web pages. These generally had one reference, though a few had two or three. These were always exchanged in the context of an instance: either a web page providing detail for a previously-discussed instance, or as a way to begin negotiation over a newly discovered instance. As a result, these referents can be folded into the representation for instances, as an extra property. For the purposes of this experiment I decided not to implement this, instead focusing on providing representations for plans versus tasks.

**Repairs** were short-lived, and completely dominated conversation when they occurred. This is in keeping with findings for other domains. As in these other domains, I determined that chat is probably the best representation for these, as it allows pure, focused negotiation and repair of common ground.

### 7.2.1 Designing matching representations

The goal of the base group experiment was to provide data to design two systems: one with representations that matched the emergent work practice, and one that conflicted with it. Examination of the data indicated that was crucial to provide an effective representation for instances and tasks, as together they accounted for well over half of all references. Based on the features predicted by the properties of the data — as mapped out in Table 7.3 — I designed a persistent grid representation for listing tasks and instances. This matched well with the secondary structure users created in their private text editors; all users listed itinerary data in a format similar to that seen in Figure 7.1. This led us to create a new representation for storing tasks and instances: the Task Table, shown in Figure 7.2.

The table consists of three free-form text columns. The “Task” column stores

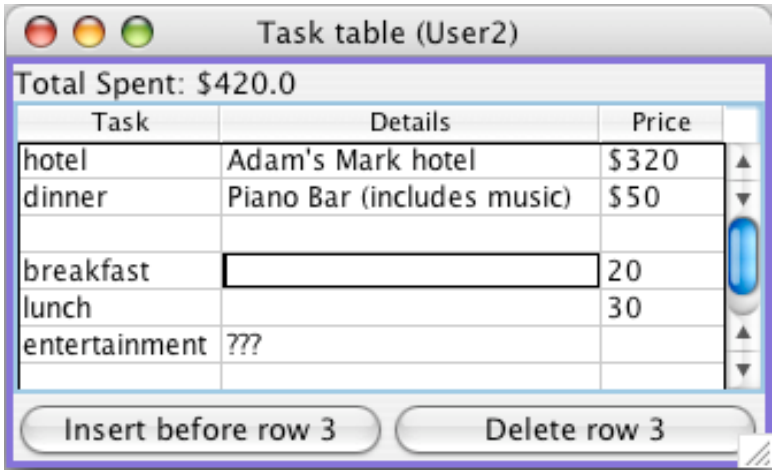


**Figure 7.2: The Task Table, designed to match observed work practice.**

Task referents; the "Details" column gives users a place to put Instance information. The "Price" column is separated out to encourage users to enter information in a structured fashion, in this case, prices in dollars and cents. By using this structure, the system is able to provide intelligent support: prices are automatically summed up, the total is shown at the top of the window and compared to the total budget for the trip, with the text turning red if the budget is currently exceeded.

These minor changes support users in attempting to plan their business trip. Tasks, stored in columns of the the task table, are given a prominent, persistent, easily-editable representation. These long-lived referents are given a dedicated location, enabling users to use visual-spatial memory to remember them; details about them are located nearby, capitalizing on the proximity effect [149] to reduce the cognitive effort of recalling and associating related information. The columns provided, "Task" (really, Name), "Details", and "Price" represent the three aspects of the information most utilized; these three information flows were the most frequent ones involving task referents. Separating out each to its own column allows users to organize their thoughts about each task instance, and to communicate this information clearly to the other user.

User planning was purposely left in the chat window. Although a complex representation might be able to improve on the interaction opportunities given by the chat window, it was deemed a poor investment of development costs. The same conclusion was reached for repairs, which in addition to having features which are difficult to design for, were extremely infrequent.

Finally, the automatic summation in the "Price" column allows automation of a common task in the domain. The secondary structure created by users to sum up their budget in the base system highlighted the need for this capability, but the structured representation of information allowed it to be automated. Study of the information flows dealing with the budget revealed that the information was used for two purposes: comparison of task instance cost to remaining budget, and a boolean check to see whether the planned activities exceeded the budget. The interface presents the results of the calculation in two ways, to allow each of these uses: a calculation of remaining money, and a change in color when the budget is exceeded. These representations were chosen to match the structure of the two necessary types of information.

### 7.2.2 Designing non-matching representations

In contrast, the goal of designing the mismatched system was to predict representations that do not match how the analysis predicts information will be used, and end up with a system which, while appearing on the surface to be extremely similar to the matched system, impairs coordination.

To this end I chose to encode plans in a persistent representation identical to the one devised for Tasks and Instances — a shared table with some automatic calculation — but alter its purpose. The result, called the Plan Table, is shown in Figure 7.3. As noted in the above analysis, plan referents tend to have a very short lifetime, with few references. This predicts that the work that participants expend to transcribe

plans into the Plan Table would be wasted work; they should not need to store this information in a persistent fashion.

The first two columns give users space to note the person responsible for the plan, and a name for the plan. To avoid having users store instance details in this representation, space in the Task column was restricted. Users were instructed to type "Future", "Active", or "Done" in the third column, Status. As with budgeting in the Task Table, this structure allows the system to provide some intelligent support: the system automatically tracks the number of incomplete tasks, the tally is displayed at the top of the window, and is highlighted in red if there are still tasks pending. While this feature is of some limited utility, asking users to make use of the Status column in this way had the side effect of increasing the number of times users had to access a putative plan. To fully utilize the feature, users would have to 'refer' to a plan (by updating the row in the shared table) at least three times, well above the observed average number of references for plan referents.

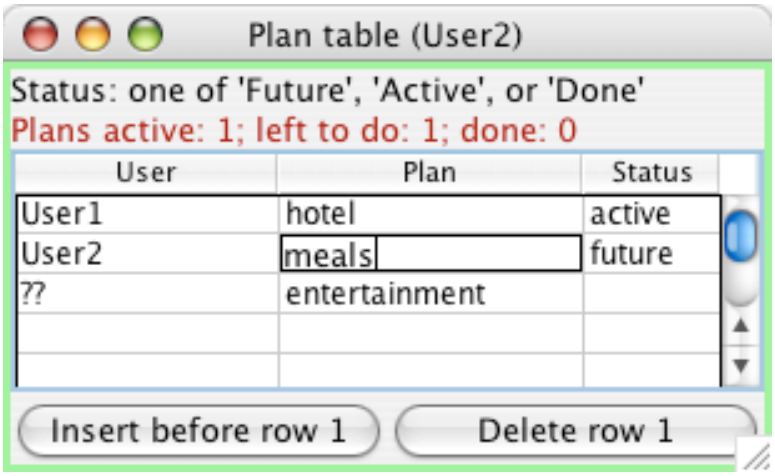


**Figure 7.3: The Plan Table, designed to conflict with observed work practice.**

In contrast, the information types that really needed a persistent representation — namely, task instances — were given no special representation, leaving users forced

to use the chat window for them. As the base system revealed, the chat window is not a particularly good representation for task instance information; its lack of effective persistence, long recall times, and inability to maintain a single point of focus for each task dramatically increases the articulation work users need to perform to create an agenda.

As an aside, the plan table does provide storage for implicit references to tasks, as it allows entry of plans to perform tasks. It is a fairly good representation for task information, and was used as such by one user group. However, as we shall see, this did not provide enough support to overcome the mismatch between users' work practice and how the representation worked, and instead was subverted by users to store task instance information.

### 7.2.3 Predictions based on referential structure analysis

My prediction was that users would be happier and more efficient when using the matching system. Since the design data showed that users needed to talk about tasks and instances, the natural representation provided by the Task Table would be used as designed, and would help users complete their task faster and more easily. Users might also fold a representation for URLs into the Task Table, given that some groups used them to talk about instances.

Conversely, I expected users to resist using the Plan Table. The data indicated that planning was really best matched with the chat window. If users used the Plan Table at all, my predictions were that they would fail to update the Status field, or perhaps subvert the representation to store the data they needed to store persistently: instance information. As we shall see below, these predictions were validated by the data.

## 7.3 Experimental results

To test the systems I organized a small-scale experiment (n=8). Dyads were trained in the domain, and then given four problems to solve; two with the 'matching' system, and two with the 'non-matching' system. Half the dyads were exposed to the 'matching' system first; the other half used the 'non-matching' system first, to counterbalance ordering effects.

I used a number of objective measures, coupled with user opinion, to evaluate the new design. These are shown in Table 7.4 and Table 7.5. The first objective measure was the length of time it took users to complete a problem. Allowing users to perform their tasks faster, all other things being equal, is a good sign of an improved system design. I found that subjects were able to finish problems on time more often when using the 'matching' system than when using the 'non-matching' system (86% complete vs. 57% complete — a 33% difference in completed problem rate). This was a good indication that the 'matching' system allowed users to perform their task more efficiently.

| Measure | Observed change |
|---|---|
| Problems finished on time | 'Non-matching' groups finished 33% fewer problems ($p < 0.1$) |
| Lines of chat | 'Non-matching' groups generated 35% more chat |

**Table 7.4: Objective results comparing 'non-matching' system to 'matching' system.**

Another useful measure of group effort is lines of chat. Past work has shown that coordination work can be reduced by providing structured representations that match the way users communicate, transcribe, and store information. One clear indicator that this is occurring is a reduction in the overall amount of information sent via

unstructured representations such as chat. Hence, by comparing the *chat output* of the initial system, where users are forced to coordinate strictly using chat, to the chat output in the new systems, where users have access to alternate representations, gives the experimenter a way to calculate how much information is being communicated via these alternate representations, a sign of their level of utility to users.

The data showed that users chatted about 35% more when using the "non-matching" systems, a statistically significant increase. This was a good indication that coordination work that had moved to the Task Table in the 'matching' system was still being done in the chat in the 'non-matching' system. As noted previously, this allows an analyst a rough measure of the quantity of coordination work that has been shifted to the new representation.

| Question | Matching | Non-matching |
|---|---|---|
| Reaction to system (1=hated it, 7=loved it) | 4.9 | 2.9 $p < 0.01$ |
| Ease of Use (1=difficult, 7=easy) | 5.6 | 3.1 $p < 0.01$ |
| Group coordination (1=poor, 7=excellent) | 5.5 | 5.0 not significant |
| Group performance (1=poor, 7=excellent) | 5.0 | 4.9 not significant |

**Table 7.5: Survey results from the initial BT study.**

User feedback and opinion were also used to gauge acceptability of the design. Designs which match the emergent work practice should be adopted with less resistance by users: they do not conflict with users' expectations about how to perform their tasks, and they allow users to coordinate more effectively. The four measures presented in Table 7.4 were gathered via an exit survey, which also provided some free-form opportunities for feedback. As can be seen, users clearly indicated a preference for the 'matching' system, rating it a 4.9 vs. 2.9 (on a scale of 1 to 7). Likewise,

they found it significantly easier to use — 5.6 vs. 3.1. Notable, however, was the lack of difference between feelings of group coordination and performance; despite objective measures to the contrary, groups generally felt their performance with both systems was above average.

### 7.3.1 Verifying predictions

The data matched my predictions fairly well. Users preferred the 'matching' system, were more efficient using it, and needed to resort to chat less. On the other side, users mostly disliked and complained about the Plan Table, although one group found it somewhat useful and praised it in the exit survey. Users generally agreed that there was a greater need to communicate the information stored in the Task Table: "Task was much easier because it provided its own concrete space to enter the completed plans themselves, rather than just whether or not the plan was done." Another user wrote, "The 'plan' system was confusing and frustrating because there was a way to record process but no way to record output." One summed it up this way: "TASK is much easier. [...] All planning really happens over chat."

The matching representations were primarily used the way they were designed to be used, with some exceptions. For example, users shoehorned URL referents into the representation for instances, and then complained about the lack of proper support for URLs. As expected, the users chafed when using the non-matching representations, and in some cases forced them to fit their own needs. While this was not unexpected, the lengths that users would go to in order to store needed information were surprising. One group started the experiments using the 'non-matching' system and promptly began storing instance information in the "Plan" column of the Plan Table. This was despite training, and the use of design techniques to discourage this behavior (column width was harshly restricted to encourage storage of short information). Likewise,

one group started storing URLs in the "Who" column. These users clearly saw a need to be communicating instance information, and created their own structure within the tools given to achieve their goals.

This data showed that I was able to use ethnographic observations of pilot study data in the design of two groupware systems, and to predict the resulting usage of these systems according to principled analysis of those observations. Experimental subjects used the representations in the fashion that I predicted, including the re-tasking of representations to store critical task information in fashions that conflicted with their basic design. This showed the higher priority for the user of completing the task as opposed to making use of the system in its designed fashion, and highlighted a recurring design difficulty in groupware systems — mismatch of representation properties with the features of information flows — that these methods can help alleviate.

# Chapter 8

# Conclusions

This thesis has demonstrated a new technique, referential structure analysis, for analyzing an ongoing interaction and building a new groupware system to support it. The technique is based on discourse analysis techniques and concepts from distributed cognition. The main thrust of the method is to identify the most important information flows, discover how users use that information, and build representations which best support those flows. It uses analysis of user references to discover what the crucial information flows are, and what the information features are for each flow.

The thesis has also shown how the method can be used to produce design recommendations, by suggesting representations which best match the observed information features. This method, as a part of a larger methodology for observation and design, significantly advances the state of the art in both understanding interaction and providing a way for analysts to guide redesign.

In this chapter, we will examine the potential impact of the methodology, and discuss some possibilities for future directions.

## 8.1 Impact of the Methodology

The analysis and redesign techniques detailed in this thesis represent a significant advancement in the state of the art for investigating interaction. By providing a structured, straightforward vocabulary for discussing information flows within a representation system, these methods provide good *descriptive* power. Additionally, through their reuse and expansion of concepts from distributed cognition, the methods allow analysts to discuss and explore concepts within interaction, giving the methods adequate *rhetorical* power.

Previous theoretical frameworks, which we build on, provide good power along these axes. Where our methods differ is in providing concrete methods for observing these flows within a specific interaction and for turning these observations into redesign recommendations. This strong *applicability* is lacking among many theoretical frameworks. Likewise, the ability of the methods to predict the impact of new representations on an interaction gives this methodology good *inferential* power, allowing an analyst to play "what if?" with a representation system.

The methods work best as a part of a larger analysis methodology. High-level methods such as Activity Theory, Workflow Analysis, and Groupware Task Analysis provide a good complement to our techniques. Although the ability to track information flows and create supporting representations for them is powerful, it must be grounded in a strong understanding of domain constraints, necessary tasks and activities of participants, and the impact of social and other constraints.

## 8.2 Future Directions

### 8.2.1 Automatic Referential Structure Extraction

The methods discussed in this thesis depend heavily on human experts to identify, understand, and classify references into lexical chains. The tagging process that generates input data for referential structure analysis requires the analyst to tag a fair percentage of the references in the discourse, which can be a cumbersome and time-consuming process. While it was outside of the scope of this thesis to attempt to improve this process through natural language processing, it is important to summarize relevant approaches which might aid an analyst in this task.

A first pass with a part-of-speech (PoS) tagger could be used to highlight the most probable candidates to be references. Such technology has become more mature during the course of this thesis, able to reliably identify parts of speech in well-formed text. However, as we have seen in the discourse samples, user chat is rarely well-formed, and often full of jargon, misspellings, and short-hand. Nevertheless, as the goal is to aid and not replace the analyst, a PoS tool could provide a valuable service.

### 8.2.2 Expansion of Information Flow Analysis

One significant area of potential development is in the area of information flow analysis. While we have established a strong theoretical background for this area, a number of open questions remain. First, we have established information flows as dependent on info type, source, destination, and purpose. However, it is clear that there are interrelations between information flows, and that in many cases certain flows are inseparable. Examination of this area should provide fruitful returns in the area of

representation recommendation; information flows which are inextricably linked can be combined into single representations to avoid cognitive dissonance on the part of users. Information flows with a logical connection — for example, the weekday and the date — should be formally connected in the theoretical model. This connection would aid a designer in creating representations which match both the structure of the information and user's expectations about that information.

Expansion of the theoretical model almost certainly requires a formal language for describing information flows and their contents. While we have established some of the vocabulary necessary for such a language, we have not defined the syntax or rules of this language. We envision a formalized description of the set of information flows within a system backed up by observational data of those flows. Observation of discourse, and other information (such as taxonomies of information and task structure) could be used to provide evidence for specific assertions within this formal language; predicates within it could be used to 'prove' certain representational solutions.

### 8.2.3 Taxonomy of Representations

Another formalism that could aid the analyst and designer is a full-fledged taxonomy of representations, potentially expressed in the formal language described above. This first requires identifying a larger set of representation properties. This would then be followed with a survey of a large number of specific representations, from the body of available software and design literature, and identifying which properties each representation possessed and for what types of information flows they would be suitable.

Another approach, and one which we have devoted a certain amount of time to, is to identify representation pieces, previously identified as 'atoms' in our historical work, which can be combined to form usable representations. These atoms closely

resemble a pattern language [3], and even more closely resemble Design Patterns [44], differing primarily in the domain they examine: representation features rather than programming constructs. These pieces can be combined in various ways to construct representations, and are easier to analyze separately. A definitional structure for these structures, along with a representative set of examples, appears in Appendix A. As this is a work in progress, the structure is open to redesign and alteration. Some units identified include: sets; ordering; sorting; incremental presentation of information; progressive display; chunked display; hierarchical display; marking; and highlighting.

By breaking down representations into these atomic units, it is easier to identify the precise properties possessed by each. Once the elements desired for a particular representation have been identified, the designer can combine the pieces to create a usable representation. This is the general approach we followed in constructing the "matching" representation system for the Business Travel experiment. Future work would involve expanding this library of representation pieces and formalizing the set of properties possessed by these pieces.

# Appendix A

# Building blocks for representations

This appendix presents a number of "representation atoms", atomic pieces which can be assembled into representations. It should be noted that this is merely a representational sample of these entities, which are quite numerous. The idea here is to explicate the definition of an atom, rather than attempt to provide a useful set of atoms from which to assemble representations.

Format:

| | |
|---|---|
| **Name(s)** | *Primary name*; Other names for the atom |
| **Type** | One of: Metaphor; Procedure; Presentation. |
| **Category** | One of: Item-based; Task-based; Info-based. |
| **Description** | A brief description of the atom. |
| **Context** | Info features that indicate the atom would be useful. |
| **Parents** | Other atoms this is made of or makes use of. |
| **Children** | Some notable atoms that use this one as a Parent. |
| **Examples** | Sample applications of the pattern. |
| **Related** | Other atoms that, while not directly related, deal with similar problems, are false analogues, or otherwise are related. |

| | |
|---|---|
| **Name(s)** | *Grouping*, Sets, Clustering |
| **Type** | Metaphor |
| **Category** | Item-based |
| **Description** | Unordered grouping of items which share some abstractable characteristic |
| **Context** | High simultaneity; atomic data; moderate to high read frequency |
| **Parents** | |
| **Children** | List |
| **Examples** | Shopping list |
| **Related** | |

| | |
|---|---|
| **Name(s)** | *Sorting* |
| **Type** | Metaphor |
| **Category** | Item-based |
| **Description** | Assign a canonical order to a group of items based on an intrinsic, shared property |
| **Context** | |
| **Parents** | Grouping |
| **Children** | |
| **Examples** | Alphabetic ordering; pre-flight checklist |
| **Related** | |

| | |
|---|---|
| **Name(s)** | *Highlighting* |
| **Type** | Presentation |
| **Category** | Item-based |
| **Description** | Call out a particular item through design, so as to attract an observer's attention |
| **Context** | Moderate to high simultaneity, low read frequency; situations where the most relevant info may not be salient. |
| **Parents** | |
| **Children** | |
| **Examples** | Bold text |
| **Related** | |

| | |
|---|---|
| **Name(s)** | *Marking* |
| **Type** | Presentation |
| **Category** | Item-based |
| **Description** | Highlight an item to indicate that the computer understands that the user's focus is there |
| **Context** | The user requires feedback about where he intends to focus attention |
| **Parents** | Feedback; Highlighting |
| **Children** | |
| **Examples** | Fronted windows; Selected icons |
| **Related** | |

| | |
|---|---|
| **Name(s)** | *Progressive Display*, Incremental display |
| **Type** | Presentation |
| **Category** | Information-based |
| **Description** | A large amount of information is presented in smaller increments. |
| **Context** | The user has requested a block of information that will exceed his ability to comprehend at once. |
| **Parents** | Display |
| **Children** | Chunked Display, Increasing Detail Display |
| **Examples** | The `more` command in Unix |
| **Related** | |

| | |
|---|---|
| **Name(s)** | *Chunked Display* |
| **Type** | Presentation |
| **Category** | Information-based |
| **Description** | A large amount of information, usually in List form, is presented in increments which are exclusive subsets of roughly equal size. |
| **Context** | The user has requested a block of information that will exceed his ability to comprehend, and it consists of a set of similar, independent items |
| **Parents** | Progressive Display |
| **Children** | |
| **Examples** | Results page of a search |
| **Related** | |

| | |
|---|---|
| **Name(s)** | *Increasing Detail Display* |
| **Type** | Presentation |
| **Category** | Information-based |
| **Description** | A large amount of information is presented in overview, and the user is given option to select a subset to explore at an increased level of detail. |
| **Context** | The user has requested a block of information that will exceed his ability to comprehend, and that information can be presented in a hierarchical fashion. |
| **Parents** | Progressive Display |
| **Children** | |
| **Examples** | Tabs for initial letters in a print dictionary; Table of Contents |
| **Related** | Index |

| | |
|---|---|
| **Name(s)** | *Verify* |
| **Type** | Procedure |
| **Category** | Task-based |
| **Description** | The plan to perform a fateful task must be ratified by another user before execution |
| **Context** | A situation where incorrect action can be harmful and where the probability of error by a single user is high; additionally, the situation must be assessable by others. |
| **Parents** | |
| **Children** | |
| **Examples** | |
| **Related** | Validate |

# Appendix B

# VesselWorld Manual

The manual for the VesselWorld groupware system appears on the following pages. This manual, originally a web page, was made available to experimental test subjects both as a print-out and as an online resource. Two versions of the manual were prepared, one for the VW-CR system and one for the VW-NO-CR system; the former is reproduced here, being a superset of the VW-NO-CR manual. The manual was produced with input from Seth Landsman and Josh Introne.

---

Welcome to VesselWorld!. This program is a simulation of a naval environment in which three ships must work together in order to remove barrels of toxic waste from a harbor. As the captain of a crane, you must travel around and pick up barrels of various sizes. As the tug captain, you will assist the cranes in waste transport by pulling small barges.

To clear the harbor, the captains of the ships must create plans, which will be submitted to the system in a step-by-step fashion. Success is achieved when all three ships coordinate their efforts to clear the area as quickly as possible and deposit all

the waste found on a barge, ready for transport from the harbor.

# B.1 Starting Up

## B.1.1 Logging in

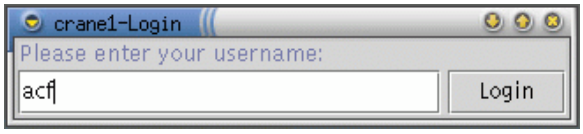


**Figure B.1: The login window**

The first window the users will see is the Login Window (Figure B.1). Enter a name and click "Login". It is important that each captain uses the same name from mission to mission.

## B.1.2 The Inhabitants of VesselWorld

Each captain will be in command of a single ship in the VesselWorld system. The ship being controlled will be drawn in red; other ships will be drawn in white. Each ship is surrounded by a white circle that denotes the affectible range of that ship. The affectible range will decrease depending on whether waste is being carried and the current weather conditions. Figure B.2 illustrates how the zone appears in the application.

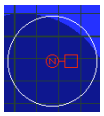

**Figure B.2: An inhabitant and its zone**

There is also a blue disk around each ship indicating its field of vision. A captain will not be able to see any objects outside this area, with the exception that all ships

know where the Large Barge is at all times, and the tug also knows where all the Small Barges are at all times. This area may also be affected by inclement weather.

### Cranes

There are two types of cranes, as pictured in Figure B.3. The top picture shows the cranes in their normal state. The main purpose of each crane is to lift and move waste. When carrying waste, the cranes will look like the second row of pictures.

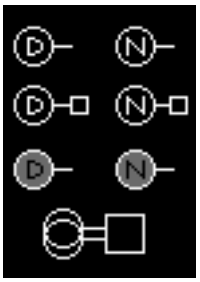


**Figure B.3: Cranes, equipment, and joined operation**

Some waste requires equipment to be deployed in order to be handled safely. For this purpose, one crane is equipped with a dredge (D), and the other is equipped with a net (N). This equipment must be deployed before waste can be lifted. When equipment is deployed, the cranes will appear differently, as shown in the third row.

Certain waste requires the cranes to work in synchronization to move. To do this the cranes must join together before moving the waste. When the cranes are joined, they will look like two interlocked circles, as in the last picture.

### Tugs

The main purpose of the tug is to aid the cranes in removing waste by transporting smaller barges around the harbor. Figure B.4 illustrates how the tug appears by itself and when it is attached to a small barge.

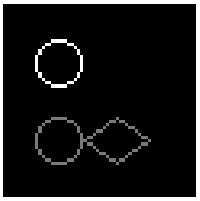

**Figure B.4: The tug**

The tug can also perform several specialized operations in order to assist the cranes with waste removal. It is the only ship that can identify the special equipment required to lift a particular barrel of waste. If leaking waste is discovered, the tug must seal it.

### Waste

Barrels of toxic waste, shown in Figure B.5, come in several sizes: Small, Medium, Large, and Extra Large. They may require special equipment or joint activity in order to be moved. Some waste is so bulky that it can not be moved by the cranes once it has been lifted and, must be loaded directly onto a Small Barge for transport. To identify the type of waste, you must inspect it using the Info window, described below.

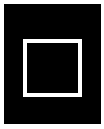

**Figure B.5: A barrel of toxic waste**

Inspecting waste is critical. You may find that special equipment is required or that some waste is currently leaking into the ocean and must be sealed before it is transported. Waste barrels will also begin to leak if they are dropped or mishandled.

| Waste size | Weight | Who can lift | Who can carry |
|---|---|---|---|
| Small | 10 | One Crane or both | One Crane or both |
| Medium | 30 | One Crane or both | One Crane or both |
| Large | 40 | Both Cranes together | Both Cranes together |
| Extra Large | 60 | Both Cranes together | **Can not** be carried |

### Leaking Waste

Leaking waste, shown in Figure B.6, causes toxic spills. These are blots on the environment that the team of ships is not equipped to clean up. If a barrel of waste is leaking, it may generate many dots of spill in its vicinity. Also, if a leaking waste is put on a barge without being sealed, it will still spill into the surrounding area.

**Figure B.6: A leaking barrel of toxic waste**

### Large Barge (brg)

The Large Barge, shown in Figure B.7, is stationary and can always be seen by every captain. The cranes may load waste on the Large Barge. In order to successfully complete the mission, all waste must end up on the Large Barge so that it can be removed from the harbor. The Large Barge has an unlimited capacity for waste.

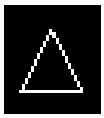

**Figure B.7: The large barge**

### The Small Barge (sbrg)

One or more Small Barges (see Figure B.8) may be available nearby. These can be pulled around by the tug in order to transport waste that is too large for the cranes

to move or to move several smaller waste barrels at once. Once loaded, Small Barges must be pulled by the tug to the Large Barge, where they can be unloaded by the cranes.

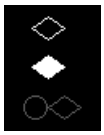

**Figure B.8: A small barge**

Small barges can carry a large, but limited, amount of waste, totaling a weight of 130. This breaks down to 2 extra-large wastes and a small waste, 3 large wastes and a small waste, or a large number of smaller wastes.

## B.2 Information and Manipulation of VesselWorld

### B.2.1 Control Center

Figure B.9 is the main Control Center window. All other information/activity windows may be opened by clicking on the corresponding button located across the top of the Control Center. These buttons will flash whenever new information is available in the corresponding window.

### B.2.2 Messages

The status of the system can be ascertained by checking the Messages panel, near the top of the Control Center. This area will display important messages about the current mode of the system, and the results of submitting a plan.

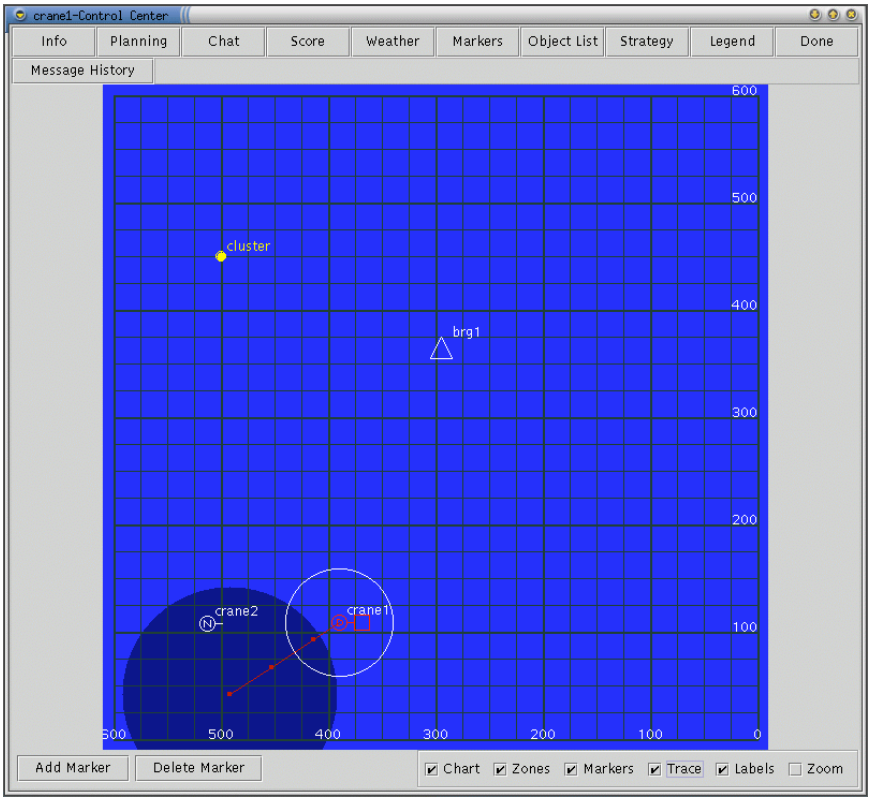


**Figure B.9: The control center**

### B.2.3 World View

This grid, shown within the Control Center, provides all visual information about ship and waste location. The check boxes in the lower right corner allow customization of the World View.

**Chart** shows/hides the lined grid overlaying the ocean

**Zones** shows/hides the white and blue circles surrounding the ship

**Markers** shows/hides any markers you have placed

**Trace** shows/hides a line and point trace of the planned motion of a ship

**Labels** shows/hides labels from objects; hiding labels is useful if the area becomes too cluttered

**Zoom** Zooms in to show the area around the ship; useful in tight situations

### B.2.4 Markers

Markers (Figure B.10 are signal points placed on the World View to keep track of object locations. A Marker can only be seen by the captain that placed it. They serve as a temporary way to mark a location.

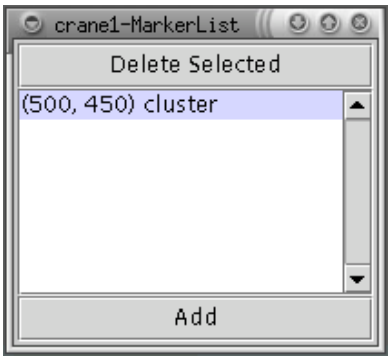


**Figure B.10: The marker list**

Clicking the "Markers" button along the top of the Control Center will open the Marker List window. This window shows the names and locations of all currently placed markers, and provides ways to manipulate the markers.

#### Adding a Marker

Click the "Add" button in the Marker List window (Figure B.11. Next click the location in the World View where the marker is to be placed. A Marker Label window will pop up; enter the label of the marker and submit it to VesselWorld. For convenience, it is also possible to click the "Add Marker" button in the lower left corner of the Control Center instead of clicking the "Add" button in the Marker List window.

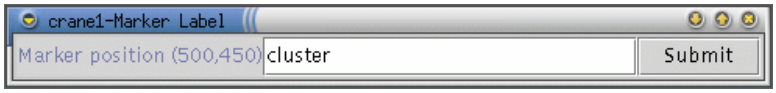


**Figure B.11: Adding a marker**

#### Deleting a Marker via the List

Click on the marker's listing in the Marker List Window. That marker will be highlighted in the World View. Clicking "Delete Selected" will then delete this marker.

#### Deleting a Marker from the World View

Click the "Delete Marker" button in the Control Center. Then click on the marker to be deleted in the World View.

### B.2.5 Weather Display

Clicking on the "Weather" button opens up the weather window display (Figure B.12. This area provides information on the current and upcoming weather conditions in the area. Keeping track of weather is very important, as weather can seriously affect the capabilities of a ship. The current weather is reported, as well as an accurate prediction for the next step.

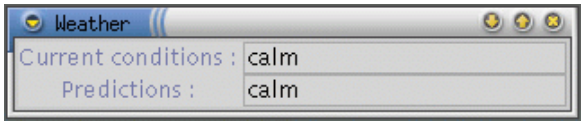


**Figure B.12: The weather window**

Weather is classified into five levels; it will not change more than one level per step taken. The levels have the following effects:

**Calm** All actions are possible

**Light** All actions are possible; watch for changing weather

**Heavy** Loading and unloading small barges requires stabilization of the barge by the tug. Joint carrying is not possible.

**Very Heavy** Stabilization required. No joint actions are possible. Affectible range and movement speed are reduced.

**Gale** Only movement is possible, at a reduced speed. Winds will cause any waste carried to be dropped.

### B.2.6 Information Window

Clicking on the "Info" button opens up the Information window (Figure B.13. This area allows closer examination of objects in VesselWorld. To enter Info Mode, click the "Enter Info Mode" button at the top of the window. Once in this mode, clicking on an object in the World View will provide information about that object. For example, clicking on barrel of waste will provide the size, location, and leak rate. If the tug is the one requesting information, it will also display what, if any, special equipment is needed to handle the barrel.

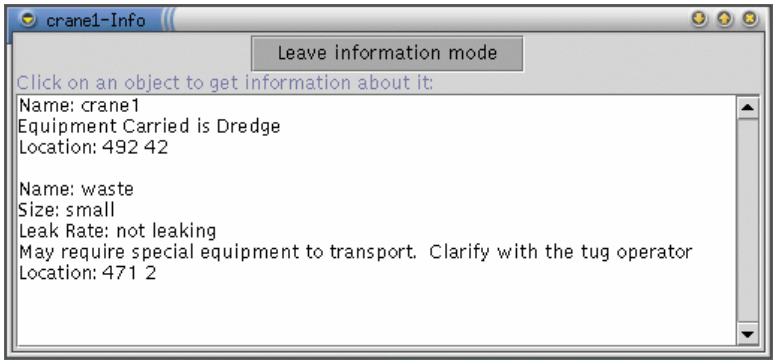


**Figure B.13: The information window**

After information is collected, click "Leave Info Mode" to resume other activities.

***NOTE* Info mode is denoted by a dark green World View. When in Info Mode, clicking the World View will not make plans; it will only give information.**

### B.2.7 Legend

Clicking on the "Legend" button opens up the Legend Window (Figure B.14. This window shows the meaning of the various symbols present throughout VesselWorld.


**Figure B.14: The legend window**

### B.2.8 Score

Clicking on the "Score" button brings up the Score window (Figure B.15, which keeps track of the group score for the current mission. The system keeps track of score to provide feedback about the efficiency of the solution. A high score is achieved in VesselWorld by moving all waste barrels in the harbor to the deck of the Large Barge as quickly and efficiently as possible.

Points are scored by loading barrels onto the large barge. Points are lost for each spot of toxic spill in the harbor. Points can also be lost by making mistakes such as


**Figure B.15: The score window**

dropping waste, taking too long, and not removing all waste from the harbor before declaring the task completed. Score is updated after each step.

Do not be overly concerned about an early negative score, as it may be difficult to find the first barrel of waste and thereby increase your score. In time, the team will be proficient enough to achieve high scores.

## B.3 Planning and the Planning Window

The various activities executed by each ship in VesselWorld must be planned out in a step by step fashion. This is done through use of both the World View and the Planning Window. A plan must be submitted by each captain each step. The system will then determine what will actually happen based on the three plans. At most one step will be executed for each captain.

### B.3.1 Planning in the World View

Clicking on the World View will create a list of steps in the Planning Window required to make that action happen. You can only perform actions on objects in your affectible range.

Remember, a step will not actually be performed until each captain submits it via the Shared Planning window.

**World View Activities**

*All Vessels*

**Move** Click on an empty location in the World View. An unburdened ship will move at most 100 grid points for each step in fair weather

*Cranes Only*

**Lift** Click on a barrel of waste to lift and hold it

**Carry** Click on an empty location while holding a barrel

**Drop** Click on the crane while holding a barrel

**Load** Click on a barge while holding a barrel

**Unload** Click on a barge that contains a barrel while not holding a barrel

**Deploy/Undeploy** Click on the crane while not holding a barrel. Special equipment must be deployed before it can be used to aid in the lifting a barrel.

**Join Action** Click on the other crane, then click to perform the desired action. Must be done by both cranes in the same step

*Tugs Only*

**Attach to a small barge** Click on the barge. Remaining attached will stabilize a barge

**Detach from a small barge** Click on the tug or barge while attached to a barge

**Seal waste** Click on a leaking barrel to seal it

### B.3.2 Planning Window

As each activity is performed in the World View, it will be recorded in the Planning Window (Figure B.16) in a step by step sequence. Notice that the plans of the other two ships are also displayed. The steps of this plan will not be executed until they have been submitted via the Planning window. Clicking on an individual step in the plan will update the World View to represent how the world will look like once all steps up to the selected one have been submitted.

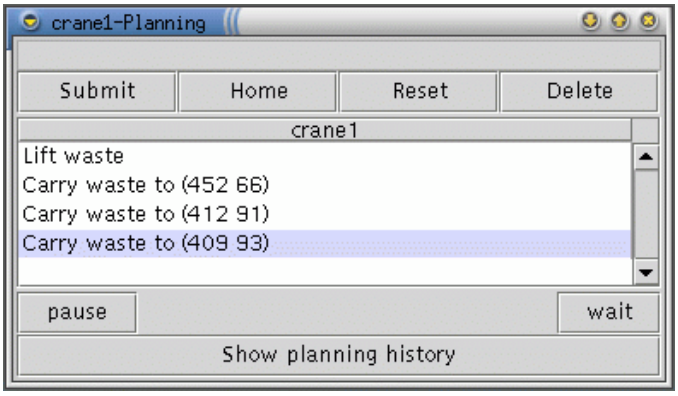


**Figure B.16: The planning window**

#### Editing and submitting the plan

There are several buttons in the Planning Window which manipulate how the plan is constructed:

**Submit** Executes the first step in the plan. The system will then lock the World View until each captain has submitted a step

**Home** Restores world view to the current status if it was displaying a future step

**Reset** Clears all plan steps from the plan

**Delete** Deletes the selected step from the plan. This may make the remaining steps in the plan invalid

**Pause** Adds a Pause to the plan. The ship will not do anything for one step

**Wait** Behaves just like a Pause step; however, it will not be removed from the plan when the plan is submitted. This allows a ship to wait indefinitely

***NOTE* Submit must be clicked *each* step, so that the other two captains can submit their plans.**

**Results of plan submission**

After all captains have submitted a plan, the system will return control of the World View to the captains. It is important to check whether you plan step was completed successfully. A status message explaining what happened will be displayed in the Message areas in both the Control Center and in the Planning window. If the plan failed, the system will explain why.

## B.4 Coordinating with other Captains

### B.4.1 Chat

Clicking on the Chat button at the top of the Control Center opens the Chat Window (Figure B.17. This allows for free form communication between captains. Messages can be typed freely in the text area at the bottom of the window. Clicking the Send button will send the message so it will be seen by all captains.

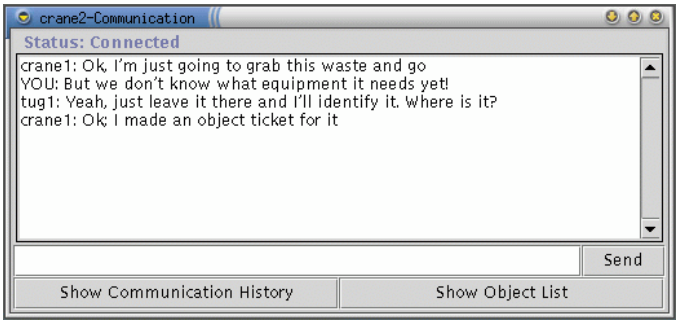


**Figure B.17: The chat window**

### B.4.2 Object List

The Object List (Figure B.18 is used to keep track of the various barrels of waste in the harbor. All captains have access to this list, and all changes made to this list will be available immediately to the other captains. No submission of changes is needed.

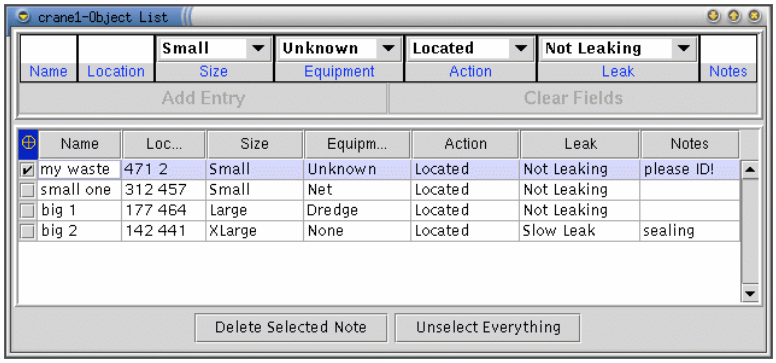


**Figure B.18: The object list window**

To add a new object, click the "Add Object" button. This creates a blank row in the list. Fill in the Object name by clicking on that column and typing in the appropriate label. Location is designated by clicking the Location column and clicking the proper location on the World View. Equipment and Status are entered using pull down menus. Notes can be added by clicking in the column and typing on the keyboard.

During the course of the mission, the objects can be updated via the same methods

that were used to enter them. Objects can also be deleted by selecting the desired object and clicking the "Delete" button.

The Object List automatically sorts the objects; by default it sorts them according to their Name. To sort the list by other columns, click in the header of the desired column. For example, to sort by the equipment needed, click in the "Equipment" header. It is also possible to rearrange the order of the columns for convenience.

To help keep track of each object, a yellow cross-hairs and label will be drawn in the World View at the location listed for each object. Note that any inaccurate information listed in the Object List will carry over to the cross-hair location. You may turn off the viewing of individual cross-hairs by unchecking the checkbox in the cross-hairs column of the appropriate object.

### B.4.3 Strategic Planning

Clicking on the Strategy button at the top of the Control Center opens up the Strategic Planning Window (Figure B.19. This area can be used to keep track of long term planning goals, and to construct complex plans to organize clean-up of the toxic waste.

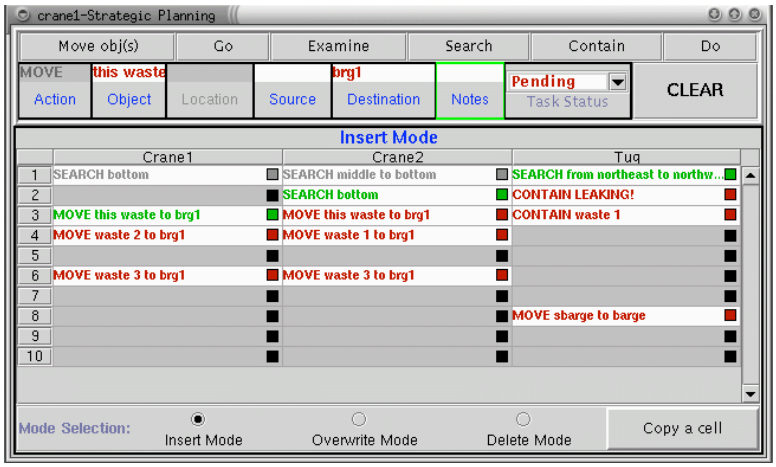


**Figure B.19: The strategic planning window**

Strategic plans are made up of entries. These are created by selecting an action, filling in the particulars of that action, and then placing that action in appropriate spot in the grid.

The action buttons are on the top of the window. Directly below this is a palette used to fill out the particulars of an action. Below this is a table that shows the strategy that all captains see. At the bottom of the window are mode buttons that affect how you interact with the table. When you click on an action button, the palette located directly below the action buttons will change to reflect that new action. Only the fields appropriate for the given action will be enabled. At this point, you may add the entry to the table immediately, or you may fill out the fields of the action.

There are three ways to fill in a field. First, you may click on the field and type. If you are typing a location you must type a valid location in the format 100 200. You may also type anything else you want to in this field, but if you do, the cross-hairs marking the location of the object will not show up.

Second, you may copy an object from the Object list; to do this, first click on the field you want to fill in. This puts you in Field Instantiation mode. Then, click on the name of the object you want to copy in the Object List itself. The name given to that object, or its location as appropriate, will be copied over.

Third, you may copy information directly from the World View; to do this, enter Field Instantiation Mode as above by clicking on the field you wish to fill in, and then click on the appropriate object or location in the World View itself. If you want to cancel the copying of information to a field, exiting Field Instantiation Mode, simply hit "Enter" when your mouse is in that field.

Note that you do not have to fill in all fields before the item can be added to the plan. Also, if you wish to clear all the fields at once, you may used the Clear button, at the far right of the palette.

When the item is filled out, you can place it in the Strategic plan by clicking on the cell in the table where you want to put it. Depending on what table mode you are in, clicking will have different effects:

| Mode | What happens when you click |
|---|---|
| Insert Mode | Inserts the new entry after the spot clicked on |
| Overwrite Mode | Overwrites the spot clicked on with the new entry |
| Delete Mode | Deletes the cell or row clicked on |

The Copy A Cell button lets you copy the contents of a cell back into the palette. To do this, click on the button and then click on the cell you want to copy. To abort copying a cell, click on the Copy A Cell button again.

The final field on the far right of the palette, beyond the Notes field, is used to indicate the color priority of the entry. The three colors, red, green, and gray, indicate the intended priority of that entry. The colors and their meaning are summarized below:

| Color | Meaning |
|---|---|
| Red | (P)ending; entries to be done later |
| Green | (C)urrent; entry currently being done |
| Gray | (D)one; entries that have been completed |

At the far right of each entry there is a small square with the current color priority, and a letter indicating that priority, of that entry. Clicking on this square will cycle the color of that entry, allowing it to be changed on the fly as entries are completed in the course of the mission.

***NOTE* Field instantiation mode is denoted by a gray World View,**

**and a gray Object List. When in this mode, clicking on the World View or Object List will not make plans or allow you to edit the object; instead, your first click will fill in the previously-selected field in the Item Palette.**

# Appendix C

# Lyze Manual

**NOTE:** this instructional manual was written and distributed to the Fall 2003 HCI class to explain how to use the Lyze tool for use in the ICR and HCI experiments. At the time, what are now called "referents" and denoted like this: REF–1A, were instead called "iotas" and denoted like this: IOTA–1A.

---

### Title: How to analyze referential structure

The purpose of this document is to teach you how to analyze the referential structure of communication in a distributed coordination situation.

## C.1 Iotas and the tagging scheme

The heart of this method is to track any conversational object (like a plan) or task object (like a waste) that the participants refer to. We call these objects iotas. Let's take an everyday example.

1. "Let's have a group meeting tomorrow at 9 in the library."

- *Let's => Let us* refers to a group of people; this is a form of task object, of a type which we could call "participants".
- *a group meeting* is, again, a task object; let's call it type "meeting".
- *tomorrow at 9* is a "time" iota
- *the library* is a "location" iota
- *the whole utterance* represents a plan for future action; this makes it an iota of type "plan". It is important to track the plans that participants talk about to perform their tasks, whether those plans are carried out or not.

2. "No way, there's class then."

   - *No way*, though it doesn't look like a reference, is a rejection of the plan to meet today, and hence implicitly refers to the "plan" iota from before.
   - *class* is a new iota of type "meeting"
   - *then* is a reference to the "time" iota ("tomorrow at 9") from the previous line
   - *the whole utterance* also represents a situation where the users need to repair their common ground; this is a common situation, and we code it as an iota of type "repair".

3. "No, it's a Brandeis Monday."

   - *No* is a continuation of the "repair" iota from the previous line.
   - *Brandeis Monday* is an iota of type "time"

4. "Oh, ok, see you then."

- *Oh, ok* acknowledges and finishes the "repair" from line 2.
- *see you then* serves to confirm the "plan" from line 1.
- *then* refers to the "time" iota from line 1 ("tomorrow at 9").

As you can see, even a simple conversation is laced with implicit and explicit references. The iota method lets you keep track of these references and code the transcript in a simple, reliable, and reproducible fashion.

### C.1.1 Iota types

Since this example was concerned with scheduling, it is not unexpected that we came up with a number of iota types dealing with scheduling: "time", "location", "meeting", "participants", and so forth. For analyzing the GREWP coding assignment you are to track iotas of the following types:

- **task work** — For the coding context problem we have been discussing in class, these include:
  - **code** — sections of the code that the users refer to
  - **variable** — variables in the code that the users refer to
  - **functionality** — program capabilities that the code is supposed to support
  - **output** — a segment of output from the code, whether correct or erroneous
  - **instruction** — an item from the instructions they were given to follow
- **plans**
- **repairs**

Depending on what your users are doing, you may need to add new types for iotas. For example, if your domain involves web browsing, you might find that the users talk about iotas of type "web page" or "web site" or "search results". If you feel you need to add new iota types, try to add as few as possible, make sure that you have strong examples for each iota type in your transcript, and justify your choices carefully.

### C.1.2 Grammar

Because you will be using the Lyze tool to automatically visualize your analysis, you need to adhere to a very particular format when performing the analysis. First, the chat should be in the format shown below. The GREWP log file converter will take GREWP log files and convert them to this format automatically.

```
61 Sarah:  I am going to get rid of your draw-pie code
```

Note that whitespace at the beginning of the line is required.

A line of chat can have zero, one, or more than one iotas after it. Each iota should be on its own line. These lines must not begin with a space.

```
61 Sarah:  I am going to get rid of your draw-pie code
[IOTA-61a code:  draw-pie code]
[IOTA-61b plan:  Sarah gets rid of IOTA-61a]
```

**If you are creating a new iota**, use the following syntax: (including the square brackets)

```
[IOTA-id type:  details]
```

e.g., from above:

```
[IOTA-61a code:  draw-pie code]

[IOTA-61b plan:  Sarah gets rid of IOTA-61a]
```

Here, *id* is a unique identifier for the iota. Use the chat line number for this id and appending a, b, c, etc. for successive iotas created on a single line. *type* is the type of referent: code, plan, repair, etc. *details* is a place to record information about the iota to help you remember what it is. You should also mention other iotas in the *details* section if the new iota incorporates them, as in this example, where the plan involves the code section.

**If you wish to simply refer to an iota**, rather than create a new one, mention its name without the square brackets:

```
64 George:  my draw-pie code?

IOTA-61a
```

Remember that for Lyze, white space is important; lines of discourse should begin with it, and your analysis should not. Then you can feed it into Lyze, which can summarizing the iota data in dot-line graphs, gives you summary information that you can copy into your favorite statistical analysis package.

## C.2 An Example

Let's go through the sample dialog and mark it up.

```
61 Sarah:  I am going to get rid of your draw-pie code

[IOTA-61a code:  draw-pie code]

[IOTA-61b plan:  sarah gets rid of IOTA-61a]
```

```
62 George:  i think so

63 Sarah:  hmmm

64 George:  my draw-pie code?

IOTA-61a

65 Sarah:  maybe I should just comment it out

IOTA-61a

[IOTA-65a plan:  comment out IOTA-61a]

66 George:  oh the default stuff...go ahead

IOTA-61a

IOTA-65a
```

On the first line, 61, Sarah indicates that she is planning to comment out a section of code. In doing so she refers to a specific section of code, giving it a name ("your draw-pie code"). We create a new iota for this code snippet, and name it Iota–61a. We also create an iota to track her plan to comment out the code, Iota–61b, and write down some details of the plan in case we need them later.

The next line, 62, refers to an earlier portion of the dialog. For the purposes of this analysis we ignore it.

Line 63 is important to the interaction, but does not make any specific references. Since we are only interested in tracking references during this analysis, and we do not note any here, we can continue straight to the next line.

On line 64, George refers to the same iota, using (basically) the same words. We note his reference by writing down the iota (Iota–61a).

Line 65 sees Sarah propose an alternate plan to deal with the draw-pie code. She refers to Iota–61a again by use of the word "it" (in "comment it out"), and also

creates a new plan, which we note as IOTA-65a.

Finally, on line 66, George gives IOTA–61A a different name ("the default stuff"), incidentally indicating that he has grounded her reference correctly. He also gives the go-ahead on her plan to comment out the code, which counts as a reference to IOTA–65A.

## C.3 Practical Issues

In performing the analysis you will run into many complications and portions of the discourse that are difficult to analyze. To help with such trouble areas we have come up with a few rules of thumb.

- **If you can't figure it out, don't panic.** In general, analyzing a subset of the whole interaction is better than perfectly analyzing only part of the interaction. If you're stuck, move on.
- **Use the whole transcript** Ambiguous references can often be solved by looking ahead or back in the dialog. In the worst case, just leave that reference aside and go on to the next.
- **Record more information rather than less.** In a situation where you are unclear as to the 'correct' interpretation, pick the more likely one, and simply note the alternate interpretation in the iota information.
- **Analyze what you see, not what you think should be there.** It is extremely tempting to attempt to 'fix' problems with the analysis; resist the urge, and instead analyze what is actually said. One symptom of this is that you will often see that two users are talking about the same thing, but don't realize it. Don't play god and

fix their misapprehension; record separate iotas for each item, even though one is a duplicate.

- **If you're not sure if two iotas are the same, record them as separate iotas.** Keep track of each iota separately; if the users really do treat them as the same referent, you can always merge them later.

## C.4 Using the Tools

The major tools you will use to do the analysis are the Lyze tool and the GREWP playback tool (VCR). The VCR tool is what has been demonstrated in class; its use is fairly straightforward. Both tools require java to run, and will run fine on the same patch machines that you have been running the GREWP tool itself on.

You can download the Lyze tool from the class web page. To run the Lyze tool, double-click on the Lyze.jar file, or type *java -jar Lyze.jar* from the command line. Using the GREWP converter script, you can export the chat transcript from a GREWP session in the format that Lyze requires. After running the Lyze tool, click on "Load Text" to load the transcript into the built-in text editor, and start entering lines of analysis lines. Remember to save your analysis using "Save Text" frequently; Lyze is NOT production software, and may have bugs that cause it to crash.

At any time you can click the "Compile this Analysis" button to extract information from the analysis you have performed so far. Lyze will then go through your analysis and summarize the IOTA references you have entered into the Line Graph, which is discussed in more detail below. If you have made a syntax error it will attempt to tell you so in the output window. (If you ran Lyze from the command line, this will be the window you ran it from; if you double-clicked the jar file, it will show

it in the java console — make sure you have access to the java console when running the app) which should help you locate the problem. Lyze is fairly picky about format, but will attempt to ignore minor errors and proceed with the rest of the analysis.

While using the Lyze tool you may find that you need to refer to what the users were doing at a specific point in the chat. Follow the instructions on how to run the GREWP VCR with appropriate log file; you will need to fast-forward to the appropriate point in the analysis manually.

### C.4.1 The Line Graph

Lyze will automatically analyze your IOTA data and present a visualization of the iota references. Each line represents the life of a single iota, with the 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 sort of iota is handled. Sorting by iota type can reveal similarities among iotas; sorting by, say, lifetime, might reveal similarities across iota categories.

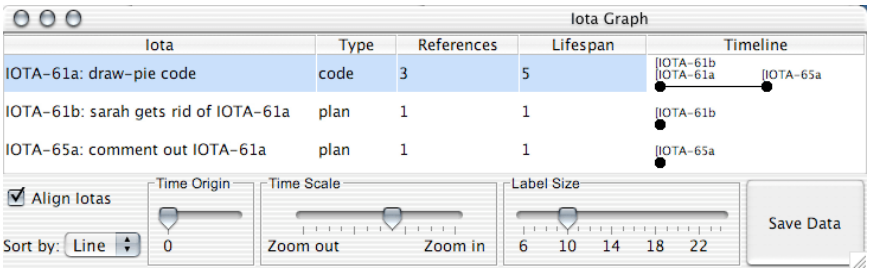


**Figure C.1: A view of the Lyze tool.**

In Figure C.1, the iotas from the example above are being graphed; not very interesting with such a short sample. Each iota is listed in order, one per row; the columns summarize information about each iota:

- **Iota** — the text of the iota, as it was in the transcript. If this looks

odd, your syntax might be incorrect in the transcript.

- **Type** — the type you assigned to this iota
- **References** — (also called Mentions) the number of times this iota appears in the analysis
- **Lifetime** — the number of utterances between first and last reference to this iota (inclusive)
- **Timeline** — a graphical representation of the access pattern for each iota.

Using the "Time Origin" slider to move the timelines left and right; "Time Scale" to zoom in and out on the data; and "Label Size" to get the reference labels to readable sizes for your monitor.

Checking "Align Iotas" will make it so that the first reference to each iota is draw as if it occurred at time 0. This is helpful for comparing iotas that occurred at different times.

You may also sort the list of iotas by various measures by selecting the desired sort order from the popup menu.

### C.4.2 Visualizations using a spreadsheet program

If you click the "Save Data" button, Lyze will export a CSV (comma-seperated values; a format that Excel and most other spreadsheet programs will understand) file with all the iota information in it. You can use the CSV file in combination with the mathematical and graphing tools of the spreadsheet to extract some conclusions from the data. Some useful numbers to look at:

1. Number of mentions to an iota (number of lines it appears on)

2. Lifetime (number of utterances between first and last reference)
3. Density of reference (1 divided by 2)
4. Lifetime (clock time between first and last utterance)
5. Importance — $f(mentions, lifetime)$ (I've used $f(x, y) = x * y$ in the past)
6. Mean time or utterances between references
7. Number of iotas of this type, in comparison to other types
8. ...many other possibilities

Graphing the data is a good way to reveal patterns. Plotting two of the above measures versus each other using a Scatter graph can reveal correlations; graphing the distribution of iotas along various axes may reveal conclusions you would otherwise miss. However, beware false correlations and other such issues in your data — try to come up with a plausible explanation for your observations, and then test them to see if they are possible.

The idea is to play with different ways to analyze the data and try to find interesting patterns. Do not change the source data itself — only change how you look at it. Perhaps you will need to average a measure over time, or discard the sections of the log files where you were training your users, or compare different sets of users to each other. In general, you are looking for significant differences between the way one sort of information is used versus another, or anything that you find surprising. By its very nature, you can not always tell what you are looking for ahead of time; you will need to explore the data to extract your conclusions.

# Bibliography


[1] Ackerman, M. Augmenting the organizational memory: A field study of answer garden. In *Computer Supported Cooperative Work (CSCW '94)* (Chapel Hill, N.C., 1994), ACM Press, pp. 243–252.

[2] Ainsworth, S. A functional taxonomy of multiple representations. *Computers and Education 33*, 2-3 (1999), 131–152.

[3] Alexander, C. *The Timeless Way of Building.* Oxford University Press, 1979.

[4] Alterman, R. The dynamics of cognition and intersubjectivity. Unpublished; available from, 2005.

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

[6] Alterman, R., Feinman, A., Landsman, S., and Introne, J. Coordination of talk and action. In *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society* (2001), W. Gray and C. Schunn, Eds., Lawrence Erlbaum Associates, p. 26.

[7] Alterman, R., Feinman, A., Landsman, S., and Introne, J. Coordination of talk: Coordination of action. Tech. Rep. CS-01-217, Brandeis University, Waltham MA, 2003.

[8] Alterman, R., and Garland, A. Convention in joint activity. *Cognitive Science 25*, 4 (2001).

[9] Apple Computer. *Human Interface Guidelines: The Apple Desktop.* San Francisco CA, 1987.

[10] Austin, J. *How To Do Things With Words.* Oxford University Press, 1962.

[11] Bannon, L., and Bødker, S. Beyond the interface: Encountering artifacts in use. In *Designing Interaction: Psychology at the Human-Computer Interface*, J. Carroll, Ed. Cambridge University Press, 1991, ch. 12, pp. 227–253.

[12] Bannon, L., and Kuutti, K. Shifting perspectives on organizational memory: From storage to active remembering. In *Proceedings of the 29th Annual Hawaii International Conference on Systems Science* (1996), vol. 3, pp. 156–167.

[13] Bannon, L., and Schmidt, K. Cscw: Four characters in search of a context. In *Studies in Computer Supported Cooperative Work: Theory, Practice, and Design*, J. Bowers and S. Benford, Eds. North-Holland, 1991, pp. 3–16.

[14] Bartle, R. Early mud history. Posted to rec.games.mud, 15 Nov 1990, 1990.

[15] Bartle, R. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD Research* (1996).

[16] Basu, A., and Blanning, R. A formal approach to workflow analysis. *Information Systems Research 11*, 1 (2000), 17–36.

[17] Bødker, S. *Through the Interface — a Human Activity Approach to User Interface Design*. Lawrence Erlbaum Associates, Hillsdale NJ, 1990.

[18] Borchers, J. *A Pattern Approach to Interaction Design*. John Wiley and Sons, New York, 2001.

[19] Bradner, E., Kellogg, W., and Erickson, T. The adoption and use of babble: A field study of chat in the workplace. In *Proceedings of the European Computer Supported Cooperative Work (ECSCW '99) conference* (1999).

[20] Breazeal, C. *Sociable Machines: Expressive Social Exchange Between Humans and Robots*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 2000.

[21] Card, S., Moran, T., and Newell, A. *The Psychology of Human-Computer Interaction*. Erlbaum, Hillsdale NJ, 1983.

[22] Carey, M., Stammers, R., and Astley, J. Human-computer interaction design: the potential and pitfalls of hierarchical task analysis. In *Task Analysis for Human-Computer Interaction*, D. Diaper, Ed. Ellis Horwood, Chichester UK, 1989, pp. 56–74.

[23] Castronova, E. Virtual worlds: A first-hand account of market and society on the cyberian frontier. *CESifo Working Paper Series*, 618 (2001).

[24] Castronova, E. *Synthetic Worlds: The Business and Culture of Online Games*. University of Chicago Press, 2005.

[25] Chuah, J., Zhang, J., and Johnson, T. The representational effect in complex systems: A distributed representation approach. In *Proceedings of Twenty Second Annual Meeting of the Cognitive Science Society* (2000).

[26] Clark, H. *Using Language*. Cambridge University Press, New York, 1996.

[27] Clark, H., and Brennan, S. Grounding in communication. In *Perspectives on Socially Shared Cognition*, J. Levine, L. Resnik, and S. Teasley, Eds. American Pyschological Association, New York, 1991, pp. 127–149.

[28] Clark, H., and Wilkes-Gibbs, D. Referring as a collaborative process. *Cognition*, 22 (1986), 1–39.

[29] Cockayne, A., Wright, P., and Fields, R. Supporting interaction strategies through the externalization of strategy concepts. In *INTERACT '99, Proceedings of the International Conference on Human-Computer Interaction* (1999), M. Sasse and C. Johnson, Eds., pp. 582–588.

[30] Cohen, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* (1960), 37–46.

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

[32] Cooper, A., and Reimann, R. *About Face 2.0: The Essentials of Interaction Design*. Wiley, 2003.

[33] Dibbell, J. A rape in cyberspace. *The Village Voice, December 21, 1993* (1993).

[34] Drew, P., and Heritage, J. *Talk at Work*. Cambridge University Press, New York, 1992.

[35] Dwyer, N., and Suthers, D. A study of the foundations of artifact-mediated collaboration. In *Computer Supported Collaborative Learning 2005: The Next 10 Years!* (2005), T. Koschmann, D. Suthers, and T. Chan, Eds., Lawrence Erlbaum Associates, pp. 135–144.

[36] Ellis, C., Gibbs, S., and Rein, G. Groupware: Some issues and experiences. *Communications of the ACM 34* (1991), 38–58.

[37] Engeström, Y., Miettinen, R., and Punamaki, R., Eds. *Perspectives on Activity Theory*. Cambridge University Press, New York, 1999.

[38] Fairclough, N. *Critical Discourse Analysis*. Longman Group UK Limited, Harlow, 1995.

[39] Feinman, A., and Alterman, R. Discourse analysis techniques for modeling group interaction. In *Proceedings of the Ninth International Conference on User Modeling* (New York, 2003), Springer-Verlag, pp. 228–237.

[40] Fields, R., Wright, P., and Harrison, M. Air traffic control as a distributed cognitive system: a study of external representations. In *Proceeedings of INTERACT '97, the International Conference on Human-Computer Interaction* (1997), T. Green, L. Bannon, C. Warren, and J. Buckley, Eds., pp. 85–90.

[41] Fjeld, M., Lauche, K., Bichsel, M., Voorhorst, F., Krueger, H., and Rauterberg, M. Physical and virtual tools: Activity theory applied to the design of groupware. *A Special Issue of Computer Supported Cooperative Work (CSCW): Activity Theory and the Practice of Design 11*, 1-2 (2002), 153–180.

[42] Flor, N., and Hutchins, E. Analyzing distributed cognition in software teams: a case study of collaborative programming duringadaptive software maintenance. In *Empirical Studies of Programmers: Fourth Workshop*, J. Koenemann-Belliveau, T. Moher, and S. Robertson, Eds. Ablex Publishing Corp., 1992.

[43] Foster, G., and Stefik, M. Cognoter: Theory and practice of a collaborative tool. In *Proceedings of the 1986 ACM Conference on Computer-Supported Cooperative Work* (Austin TX, 1986), ACM Press, pp. 7–15.

[44] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1997.

[45] Garland, A., and Alterman, R. Autonomous agents that learn to better coordinate. *Autonomous Agents and Multi-Agent Systems 8*, 3 (2004), 267–301.

[46] Gell-Mann, M. What is complexity? *Complexity 1*, 1 (1995), 16–19.

[47] Gergle, D., Kraut, R., and Fussell, S. Communicating with action. In *CSCW'04: Proceedings of the ACM Conference on Computer Supported Cooperative Work* (2004), ACM Press, pp. 487–496.

[48] Goodman, B., Linton, F., Gaimari, R., Hitzeman, J., Ross, H., and Zarrella, G. Using dialogue features to predict trouble during collaborative learning. *User Modeling and User-Adapted Interaction 15*, 1 (2005), 85–134.

[49] Gould, J., and Lewis, C. Designing for usability: Key principles and what designers think. *Communications of the ACM 28*, 30 (1985), 300–311.

[50] Greenbaum, J., and Kyng, M., Eds. *Design at work: cooperative design of computer systems*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1992.

[51] Grosz, B., Joshi, A., and Weinstein, S. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics 2*, 21 (1995), 203–225.

[52] Grosz, B., and Sidner, C. Attention, intentions, and the structure of discourse. *Computational Linguistics 12*, 3 (1986).

[53] Grudin, J. Why cscw applications fail: Problems in the design and evalutation of organizational interfaces. In *Proceedings of CSCW '88, Portland, Oregon, USA* (1988), pp. 85 – 93.

[54] Halverson, C. Traffic management in air control: Collaborative management in real time. *SIGOIS Bulletin 15*, 2 (1994), 7–10.

[55] Halverson, C., Erickson, T., and Sussman, J. What counts as success? punctuated patterns of use in a persistent chat environment. In *Proceedings of GROUP 2003* (2003).

[56] Hickey, T., Langton, J., Granville, K., and Alterman, R. Enhancing cs programming lab courses using collaborative editors. In *Proceedings of CCSCE04, 15-16 Oct 2004* (2004).

[57] Hollan, J., Hutchins, E., and Kirsh, D. Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction 7*, 2 (2000), 174–193.

[58] Hutchins, E. *Cognition in the Wild*. MIT Press, Cambridge, 1995.

[59] Hutchins, E. How a cockpit remembers its speeds. *Cognitive Science 19* (1995), 265–288.

[60] Hutchins, E., and Klausen, T. Distributed cognition in an airline cockpit. In *Communication and Cognition At Work*, D. Middleton and Y. Engeström, Eds. Cambridge University Press, 1996, pp. 15–34.

[61] Introne, J., and Alterman, R. Leveraging collaborative effort to infer intent. In *Proceedings of the Ninth International Conference on User Modeling* (2003), pp. 133–137.

[62] Introne, J., and Alterman, R. Leveraging a better interface language to simplify adaptation. In *Proceedings of the International Conference on Intelligent User Interfaces 2004* (2004), N. Nunes and C. Rich, Eds., pp. 262–264.

[63] Introne, J., and Alterman, R. Using cognitive artifacts to bridge the tool-agent divide. Technical Report CS-05-260, Brandeis University, 2005.

[64] Jacobson, D. Impression formation in cyberspace: Online expectations and offline experiences in text-based virtual communities. *JCMC 5*, 1 (1999).

[65] John, B., and Kieras, D. The goms family of user interface analysis techniques: Comparision and contrast. *ACM Transactions on Computer-Human Interaction 3* (1996), 320–351.

[66] John, B., and Kieras, D. Using goms for user interface design and evaluation: Which technique? *ACM Transactions on Computer-Human Interaction 3*, 4 (1996), 287–319.

[67] John, B., Prevas, K., Salvucci, D., and Koedinger, K. Predictive human performance modeling made easy. In *CHI 2004* (2004), pp. 455–462.

[68] Kammer, P., Boicer, G., Taylor, R., Hitomi, A., and Bergman, M. Techniques for supporting dynamic and adaptive workflow. *Computer Supported Cooperative Work 9*, 3-4 (2000).

[69] Kaptelinin, V., and Nardi, B. Activity theory: Basic concepts and applications. In *Proceedings of CHI 1997 Electronic Publications: Extended Abstracts* (1997), E. Francik and J. Larson, Eds.

[70] Kirwan, B., and Ainsworth, L., Eds. *A Guide to Task Analysis*. Taylor and Francis, London, 1992.

[71] Kraut, R., Fussell, S., and Siegel, J. Visual information as a conversational resource in collaborative physical tasks. *Human-Computer Interaction 18* (2003), 13.49.

[72] Kraut, R., Patterson, M., Lundmark, V., Kiesler, S., Mukopadhyay, A., and Scherlis, W. The internet paradox: A social technology that reduces social involvement and psychological well-being? *American Psychologist 53*, 9 (1998), 1017–1031.

[73] Kuutti, K. Activity theory as a potential framework for human-computer interaction research. In *Context and Consciousness: Activity Theory and Human-Computer Interaction*, B. Nardi, Ed. MIT Press, 1996.

[74] Kwan, M., and Balasubramanian, P. Dynamic workflow management: a framework for modeling workflows. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences* (1997), vol. 4, pp. 367–376.

[75] Landsman, S. *A Software Lifecycle for Building Groupware Applications: Building Groupware On THYME*. PhD thesis, Brandeis University, 2006.

[76] Landsman, S., and Alterman, R. Analyzing usage of groupware. Tech. Rep. CS-02-230, Brandeis University, Waltham MA, 2002.

[77] Landsman, S., and Alterman, R. Building groupware on thyme. Tech. Rep. CS-03-234, Brandeis University, Waltham MA, 2003.

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

[79] Leont'ev, A. *Activity, Consciousness, and Personality*. Prentice-Hall, Englewood Cliffs NJ, 1978.

[80] Lewis, D. *Convention: a Philosophical Study*. Harvard University Press, Cambridge, 1969.

[81] Li, M., and Vitányi, P. *An Introduction to Kolmogorv Complexity and Its Applications*. Springer, 1997.

[82] Lockman, A., and Klappholz, A. Toward a procedural model of contextual reference solution. *Discourse Processes 3* (1978), 25–71.

[83] Lockman, A., and Klappholz, D. Control of inferencing in natural language understanding. *Computers and Math with Applications 9*, 1 (1983), 59–70.

[84] Malone, T., Grant, K., Lai, K., Rao, R., and Rosenblitt, D. Semistructured messages are surprisingly useful for computer-supported coordination. *ACM Transactions on Office Information Systems 5*, 2 (1987), 115–131.

[85] McCabe, R., Heath, C., Burns, T., and Priebe, S. Engagement of patients with psychosis in the consultation: Conversation analytic study. *Bristish Medical Journal 325*, 7373 (2002), 1148–1151.

[86] Medina-Mora, R., Winograd, T., Flores, R., and Flores, F. The action workflow approach to workflow management technology. In *Proceedings of the 1992 ACM conference on Computer-Supported Cooperative Work* (1992), ACM Press, pp. 281–288.

[87] Morris, M., Nadler, J., Kurtzberg, T., and Thompson, L. Schmooze or lose: Social friction and lubrication in e-mail negotiations. *Group Dynamics 6*, 1 (2002), 89–100.

[88] Muller, M., and Kuhn, S. Participatory design. *Communications of the ACM 36*, 6 (1993), 24–28.

[89] Mumford, E., and Henshall, D. *Participative Approach to Computer Systems Design: A Case Study of the Introduction of a New Computer System.* Halsted Press, New York, NY, USA, 1978.

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

[91] Nielsen, J. The usability engineering life cycle. *IEEE Computer 25*, 3 (1992), 12–22.

[92] Nielsen, J. Heuristic evaluation. In *Usability Inspection Methods*, J. Nielsen and R. Mack, Eds. John Wiley and Sons, 1994.

[93] Nielsen, J. *Designing Web Usability: the Practice of Simplicity.* New Riders Publishing, Indianapolis IN, 2000.

[94] Norman, D. Design principles for human-computer interfaces. In *Applications of Cognitive Pyschology: Problem Solving, Education, and Computing*, D. Berger, K. Pezdek, and W. Banks, Eds. Lawrence Erlbaum Associates, Hillsdale NJ, 1986, pp. 0–0.

[95] Norman, D. Cognitive artifacts. In *Designing Interaction: Psychology at the Human-Computer Interface*, J. Carroll, Ed. Cambridge University Press, 1991, ch. Cognitive Artifacts, pp. 0–0.

[96] Norman, D. *Things That Make Us Smart.* Addison-Wesley, 1993.

[97] Norman, D. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer is so Complex, and Information Appliances are the Solution.* MIT Press, Cambridge, 1998.

[98] Olson, J., Olson, G., Storrosten, M., and Carter, M. How a group-editor changes the character of a design meeting as well as its outcome. In *Proceedings of ACM CSCW'92* (1992), pp. 91–98.

[99] Perkins, D. Person-plus: a distributed view of thinking and learning. In *Distributed Cognitions: Psychological and Educational Considerations*, G. Salomon, Ed. Cambridge University Press, New York, 1993, pp. 88–110.

[100] Petri, C. Kommunikation mit automaten. In *Schriften des IIM* (1962), Institut für Instrumentelle Mathematik. Kommunikation mit Automaten. Petri, C.A., Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.

[101] Picard, R. *Affective Computing.* MIT Press, Cambridge, MA, 1997.

[102] Rasmussen, J. *Information Processing and Human-Machine Interaction and Approach to Cognitive Engineering.* North-Holland, New York, 1986.

[103] Reason, J. *Human Error.* Cambridge University Press, 1990.

[104] Rich, C., and Sidner, C. Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction 8*, 3-4 (1998), 315–350.

[105] Rogers, Y. Coordinating computer-mediated work. *Computer Supported Cooperative Work (CSCW) 1* (1993), 295–315.

[106] Rogers, Y., and Ellis, J. Distributed cognition: an alternative framework for analysing and explaining collaborative working. *Journal of Information Technology 9*, 2 (1994), 119–128.

[107] Sacks, H. *Lectures on Conversation.* Basil Blackwell, Oxford, 1992.

[108] Sacks, H., Schegloff, E., and Jefferson, G. A simplest systematics for the organization of turn-taking for conversation. *Language 50* (1974), 696–735.

[109] Salvucci, D., Zuber, M., Beregovaia, E., and Markley, D. Distract-r: rapid prototyping and evaluation of in-vehicle interfaces. In *CHI 2005* (2005), pp. 581–589.

[110] Scaife, M., and Rogers, Y. How do graphical representations work? *International Journal of Human-Computer Studies 45* (1996), 185–213.

[111] Schank, R., and Abelson, R. *Scripts, Plans, Goals, and Understanding: an Inquiry into Human Knowledge Structures.* Erlbaum, 1977.

[112] Schegloff, E. Identification and recognition in telephone conversation openings. In *Everyday Language Studies in Ethnomethodology*, G. Psathas, Ed. Irvington Publishers, New York, 1979, pp. 23–78.

[113] Schegloff, E. Preliminaries to preliminaries: 'can i ask you a question?'. *Sociological Inquiry 50*, 3-4 (1980), 104–152.

[114] Schegloff, E. The routine as achievement. *Human Studies 9* (1986), 111–151.

[115] Schegloff, E. Conversation analysis and socially shared cognition. In *Perspectives on Socially Shared Cognition*, J. Levine, L. Resnick, and S. Teasley, Eds. American Pyschological Association, New York, 1991, pp. 150–171.

[116] Schegloff, E., and Sacks, H. Opening up closings. *Semiotica 7* (1973), 289–327.

[117] Schmidt, K., and Simone, C. Coordination mechanisms: Towards a conceptual foundation for cscw systems design. *International Journal of Computer Supported Cooperative Work 5*, 2-3 (1996), 155–200.

[118] Schmidt, K., and Wagner, I. Coordinative artifacts in architectural practice. In *Cooperative Systems Design: a Challenge of the Mobility Age, Proceedings of the Fifth International Conference on the Design of Cooperative Systems (COOP 2002)* (2002), M. e. a. Blay-Fornarino, Ed., pp. 257–274.

[119] Schmidt, M. The evolution of workflow standards. *IEEE Concurrency 7*, 3 (1999).

[120] Schraagen, J., Chipman, S., and Shalin, V., Eds. *Cognitive Science Analysis*. Lawrence Erlbaum Associates, Mahwah NJ, 2000.

[121] Schuler, D., and Namioka, A., Eds. *Participatory design: Principles and practices*. Lawrence Erlbaum Associates, 1993.

[122] Searle, J. *Speech Acts: an Essay in the Philosphy of Language*. Cambridge University Press, New York, 1969.

[123] Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd ed. Addison-Wesley Publishing, Reading MA, 1998.

[124] Short, J., Williams, E., and Christie, B. *The Social Psychology of Telecommunication*. John Willey and Sons, Ltd., London, 1976.

[125] Sidner, C., Lee, C., and Lesh, N. The role of dialog in human robot interaction. In *International Workshop on Language Understanding and Agents for Real World Interaction* (2003).

[126] Stubbs, M. *Discourse Analysis: the Sociolinguistic Analysis of Natural Language*. Basil Blackwell, Oxford, 1983.

[127] Suchman, L., and Trigg, R. Understanding practice: video as a medium for reflection and design. In *Design At Work*, J. Greenbaum and M. Kyng, Eds. Erlbaum, Hillsdale NJ, 1991, pp. 65–90.

[128] Suchman, L., and Trigg, R. Artificial intelligence as craftwork. In *Understanding Practice: Perspectives on Activity and Context and Consciousness: Activity Theory and Human-Computer Interaction*, S. Chaiklin and J. Lave, Eds. Cambridge University Press, New York, 1993, pp. 144–178.

[129] Suthers, D. Collaborative knowledge construction through shared representations. In *Proceedings of the 38th Hawai'i International Conference on the System Sciences (HICSS-38)* (2005).

[130] Suthers, D., Girardeau, L., and Hundhausen, C. The roles of representations in online collaborations. In *Annual Meeting of the American Educational Research Association (AERA)* (2002).

[131] Suthers, D., and Hundhausen, C. The effects of representation on students' elaborations in collaborative inquiry. In *Proceedings of Computer Support for Collaborative Learning 2002* (2002), pp. 472–480.

[132] Tannen, D. Gender differences in topical coherence: Creating involvement in best friends' talk. *Discourse Processes 13*, 1 (1990), 73–90.

[133] Taylor, F. *The Principles of Scientific Management*. Harper Bros., New York, 1911.

[134] van Deemter, K., and Kibble, R. What is coreference and what should coreference annotation be? In *Proceedings of the ACL'99 Workshop on Coreference and its Applications* (1999), pp. 90–96.

[135] van der Aalst, W. Woflan: a petri-net-based workflow analyzer. *Systems Analysis Modelling Simulation 35*, 3 (1999).

[136] van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., and Barros, A. Workflow patterns. *Distributed and Parallel Databases 14*, 3 (2003), 5–51.

[137] van der Aalst, W., and van Hee, K. *Workflow Management: Models, Methods, and Systems*. MIT Press, 2002.

[138] van der Veer, G., Lenting, B., and Bergevoet, B. Gta: Groupware task analysis - modeling complexity. *Acta Psychologica 91* (1996), 297–322.

[139] van der Veer, G., and van Welie, M. Task based groupware design: putting theory into practice. In *Proceedings of DIS2000* (2000).

[140] Van Dijk, T. A., and Kintsch, W. *Strategies of discourse comprehension*. Academic Press, 1983.

[141] van Labeke, N., and Ainsworth, S. Representational decisions when learning population dynamics with an instructional simulation. In *Intelligent Tutoring Systems*, S. Cerri, G. Gouardères, and F. Paraguaçu, Eds. Springer-Verlag, 2002, pp. 831–840.

[142] van Welie, M., van der Veer, G., and Koster, A. Integrated representations for task modeling. In *Proceedings of the Tenth European Conference on Cognitive Ergonomics* (2000), pp. 129–138.

[143] Vicente, K. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Lawrence Erlbaum and Associates, Mahwah NJ, 1999.

[144] Vygotsky, L. *Thought and Language*. MIT Press, Boston, 1962.

[145] Walther, J. Impression development in computer-mediated interaction. *Western Journal of Communication 57* (1993), 381–398.

[146] Walther, J. Computer-mediated communication: Impersonal, interpersonal, and hyperpersonal interaction. *Communication Research 23*, 1 (1996), 3–43.

[147] Walther, J. Group and interpersonal effects in international computer-mediated collaboration. *Human Communication Research 23*, 3 (1997), 342–369.

[148] Whittaker, S., and Sidner, C. Email overload: Exploring personal information management of email. In *Proceedings of CHI'96 Conference on Computer Human Interaction* (NY, 1996), ACM Press, pp. 276–283.

[149] Wickens, C. The proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors 37* (1995), 473–494.

[150] Wickens, C., and Hollands, J. *Engineering Psychology and Human Performance*. Prentice Hall, 1999.

[151] Winograd, T. A language/action perspective on the design of cooperative work. *Human-Computer Interaction 3*, 1 (1988), 3–30.

[152] Winograd, T., and Flores, F. *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley, 1986.

[153] Wright, P., Fields, B., and Harrison, M. Analyzing human-computer interaction as distributed cognition: the resources model. *Human-Computer Interaction 15*, 1 (2000), 1–41.

[154] Wright, P., Fields, R., and Harrison, M. Distributed information resources: a new approach to interaction modeling. In *Cognition and the worksystem. Proceedings of the 8th European Conference on Cognitive Ergonomics (ECCE 8)* (1996), EACE Press, pp. 5–10.

[155] Zhang, J. The nature of external representations in problem solving. *Cognitive Science 21*, 2 (1997), 179–217.

[156] Zhang, J. The coordination of external representations and internal mental represenatations in display-based cognitive tasks. In *Diagrams* (2000), M. Anderson, P. Cheng, and V. Haarslev, Eds., vol. 6.

[157] Zhang, J., and Norman, D. Representations in distributed cognitive tasks. *Cognitive Science 18* (1994), 87–112.