Designing for lifeworlds: genre and activity in information systems design and evaluation

Clay Ian Spinuzzi

Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Computer Sciences Commons, Library and Information Science Commons, Rhetoric and Composition Commons, and the Speech and Rhetorical Studies Commons

Recommended Citation
Spinuzzi, Clay Ian, "Designing for lifeworlds: genre and activity in information systems design and evaluation " (1999). Retrospective Theses and Dissertations. 12170.
https://lib.dr.iastate.edu/rtd/12170

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700  800/521-0600
NOTE TO USERS

This reproduction is the best copy available

UMI
Designing for lifeworlds:
Genre and activity in information systems design and evaluation

by

Clay Ian Spinuzzi

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Rhetoric and Professional Communication
Major Professor: David R. Russell

Iowa State University
Ames, Iowa
1999

Copyright © Clay Ian Spinuzzi, 1999. All rights reserved.
Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of
Clay Ian Spinuzzi
has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate College
TABLE OF CONTENTS

ACKNOWLEDGMENTS ............................................................... v
ABSTRACT ................................................. vii

INTRODUCTION. A RESEARCH METHODOLOGY FOR INFORMATION SYSTEMS DESIGN AND EVALUATION ........................................ 1

CHAPTER 1. TOWARDS A THEORY OF ARTIFACTS: CONSTRUCTING A THEORETICAL FRAMEWORK FOR INFORMATION SYSTEMS DESIGN AND EVALUATION ........................................ 6

CHAPTER 2. METHODS ............................................................... 48

CHAPTER 3. GENRE ECOLOGIES AND THE LOCATION OF USABILITY: A STUDY OF SOFTWARE DEVELOPMENT AS MEDIATED ACTIVITY .......... 70

CHAPTER 4. THE ALAS ACTIVITY NETWORK ..................................... 104

CHAPTER 5. CONTRADICTIONS, DISCOORDINATIONS, AND BREAKDOWNS: A STUDY OF PC-ALAS USE ........................................ 153

CHAPTER 6. EMBEDDED CONTRADICTIONS: TWO STUDIES OF GIS-ALAS GENRE HYBRIDS ........................................ 191

CONCLUSION. DESIGNING FOR LIFEWORLDS ..................................... 238

APPENDIX A. INTERVIEW QUESTIONS FOR PILOT STUDY ..................... 256

APPENDIX B. QUESTIONNAIRES FOR PC-ALAS AND GIS-ALAS USERS .... 257

APPENDIX C. INSTRUCTIONS FOR PC-ALAS Raters .......................... 262

APPENDIX D. PC-ALAS GROUPS .................................................. 266

APPENDIX E. PC-ALAS QUANTITATIVE DATA .................................. 267

APPENDIX F. QUESTIONNAIRE FOR STUDENT PARTICIPANTS ............. 270

APPENDIX G. INSTRUCTIONS FOR GIS-ALAS RATER ........................ 275
ACKNOWLEDGMENTS

A dissertation has only one person's name on it, but many people's labor goes into it. That is doubly true for this dissertation. During this project I have benefited enormously from the comments, advice, and help of many people.

Chief among those people are my dissertation committee: David R. Russell, Rebecca Burnett, Dorothy Winsor, Dan Douglas, and Roger Baer. Dr. Russell in particular has been unstinting in his dedication to, and enthusiasm for, the project. He has been a mentor, sounding board, reviewer, and editor; without his careful guidance, this dissertation would be greatly diminished.

Others have dedicated their time to making this project successful. Mark Zachry, Lee-Ann Kastman, and Lee Tesdell helped me to shape my research and gave me early feedback on chapters 1 and 3. Bonnie Nardi and Barbara Mirel were kind enough to review drafts of the same chapters; their illuminating comments helped me to rethink some of my ideas about genre ecologies in particular. Lee Honeycutt shared his knowledge on Cohen's kappa. Donna Kain lent her time unselfishly as my rater, helping me to shape my video coding scheme and to achieve rather high kappas.

At Schlumberger Well Services, Sam McLellan and Al Roesler arranged for me to intern with the Usability Services team in 1998, giving me the chance to conduct the study described in Chapter 3. Their guidance made the study a success; their pragmatism helped me to remember that this project should have practical as well as theoretical ramifications.

At the Center for Transportation Research and Education, Reg Souleyrette and Michael Pawlovich allowed me access to GIS-ALAS, put me in contact with users, and answered multiple questions. Tim Strauss gave me the chance to include Iowa State students from his GIS class in the study.

At the Iowa Department of Transportation, Joyce Emery and John Nervig gave generously of their time and archives, making the historical study possible and enriching the dissertation in surprising and important ways.
Multiple users agreed to be questioned, observed, videotaped, interviewed, and otherwise inconvenienced during the dissertation research. Although I cannot thank them by name, I am grateful to the 22 Schlumberger software developers, the 26 PC-ALAS users from all corners of Iowa, and the 13 Iowa State University students who participated in this project.

These entities helped me in collecting and thinking through the data. But my analysis of the data would not be nearly as detailed or thoughtful if it were not for the one-semester research stipend awarded to me by the Haggard Fellowship Committee at Iowa State University (fall 1998). The Haggard Fellowship has made an enormous difference in the scope and depth of the dissertation. In particular, I would like to thank Michael Mendelson, under whose guidance the Fellowship was (and is still being) administered.

Last but certainly not least, I want to thank my wife, Gail Bayeta. Her observations about desktop publishing software interfaces inspired the train of thought that led to this project. She has served as a sounding board, critic, and balance, leaving her thoughtful traces throughout the dissertation.
ABSTRACT

Increasingly, professional communicators design and evaluate information systems (ISes) (e.g., online help, websites, databases). Yet the dominant theoretical frameworks (computational psychology, formalism) and research methodologies for designing and evaluating ISes are limited in important ways. These Cartesian frameworks tend not to explore rhetorical issues: society, culture, history, interpretation. I address these issues with a theory and a research methodology (a theory-guided research approach) that I developed by conducting five empirical studies of IS use.

I critique existing theories and research methodologies, then draw on scholars such as Mikhail Bakhtin, Lev Vygotsky, Yrjö Engeström, Edwin Hutchins, and Charles Bazerman to develop a theory of artifacts adapted from genre theory and activity theory. My theory posits five tenets: activities are cyclical and developing; artifacts are interpreted segments of the material environment; artifacts' rules are complex, relatively stable, co-developing, and socially distributed; artifacts instantiate cultural-historical traditions; and artifacts occupy ecologies.

I develop the theory and research methodology through qualitatively and quantitatively analyzed empirical studies of how software developers write and interpret programs (i.e., program production and comprehension); how personnel in city engineering, law enforcement, and traffic safety interpret and use a database; and how workers and college students interpret and use a geographic information system.

The studies suggest that any IS is part of a cluster of co-developing genres that collectively mediate activities. In such genre ecologies, off-screen genres are often imported into the interface, mingling with on-screen genres to produce genre hybrids.

These studies also suggest that a given IS's usability, though customarily conceived as a quality of the IS itself, is distributed across the entire genre ecology and the activities it mediates. I use a tripartite structure to investigate usability: contradictions in the activity engender discoordinations among genres, manifesting as interpretive breakdowns.
Through these studies, I refine the research methodology that involves tracing breakdowns users encounter when using an IS, then analyzing those breakdowns to diagnose and address deeper design issues. The methodology can help designers to (a) design ISes that support users' activities; (b) recognize and design for users' genre ecologies; and (c) design the very activities that the artifacts mediate.
INTRODUCTION
A RESEARCH METHODOLOGY FOR
INFORMATION SYSTEMS DESIGN AND EVALUATION

Computerized information systems are becoming increasingly important to a variety of fields. Systems such as online help, online documentation, World-Wide Web sites, and databases tend to support users' labor by providing instructions, data, or details relevant to their workaday tasks. In this dissertation I discuss how such information systems impact fields and disciplines such as human-computer interaction studies, computer documentation, computer-supported cooperative work, program comprehension and production, law enforcement, civil engineering, and traffic safety; but I could just as easily have examined graphic design, instructional technology, library and information science, or rhetoric and composition. Despite their differences, these fields and disciplines are all increasingly reliant on computer-based information systems. Consequently, all have a stake in information system design and evaluation — especially in terms of usability, the question of whether an artifact such as an information system can suitably support its users' tasks.

Yet current research methodologies guiding the design and evaluation of information systems have certain limitations. In John M. Carroll's landmark collection Designing Interaction (1991), human-computer interaction specialists working within a variety of research traditions and theoretical frameworks point out some of these limitations. In particular, they argue that current methodologies do not provide a strong framework for systematizing some types of observations (Brooks, 1991) and they insufficiently address the meaningfulness of artifacts for their users (Bannon and Bødker, 1991; Norman 1991). Such criticisms are leveled at the research methodologies that have been adopted by human-computer interaction researchers, but they could just as easily apply to methodologies developed by those in various other fields such as computer documentation.
In this dissertation, I suggest that part of the problem with such methodologies is that many are rooted in a Cartesian idea of cognition. Once its central tenet is accepted—Descartes' *cogito ergo sum*, the "I think, therefore I am" which separates the individual mind from the world—various other oppositional dualisms follow: mind-body, organism-environment, actor-society, situation-history, text-context, and form-content, to name just a few. Such dualisms lead researchers to privilege one of each pair over the other. They result in research methodologies that do not always thoroughly explore some of the issues with which rhetoricians are concerned: society, culture, history, and interpretation. I argue that exploring these issues in a systematic way—through a well-articulated research methodology—can lead to rich insights into information system use, design, and evaluation. This dissertation represents my attempt to develop such a methodology, *a non-Cartesian research methodology to guide research into information systems design and evaluation*.

In Chapter 1, I explore Cartesian dualisms through a critique of various theories of artifacts and their attendant research methodologies. I then construct a theory of artifacts that can define a cultural-historical unit of research; account for the relation between artifacts and social practices; account for the relatively stable clusters of practices that form around information systems; and account for the holistic semiotic dimension of artifacts such as information systems. I derive this theory primarily by synthesizing two theories that have some bearing on this matter—Vygotskian activity theory and Bakhtinian genre theory. This theory becomes the basis for the research methodology developed in later chapters.

Chapter 2 introduces four research questions that I explore as I continue to develop and test the methodology I am developing:

- Where is "usability" located? That is, working within a cultural-historical framework, how can we locate and conceptualize the difficulties that people encounter as they attempt to use information systems?
• How do participants in an activity come to perceive and manage the various genres used in that activity to mediate their work?
• How do participants come to encounter difficulties in perceiving and managing the various on- and off-screen genres?
• If information system designers import some of the many off-screen genres into the interface — transforming them into on-screen genres — how do these changes affect how users perceive and manage the genres?

Chapter 2 also outlines the methods that I use to investigate these questions through five empirical studies.

The theory of artifacts can be used to explore issues revolving around usability, but in the process it calls the very notion of usability into question. In Chapter 3, I examine how software developers share information; the results of the research are applied to the design of an information system. Yet I find that if we examine artifacts from a non-Cartesian standpoint, we find that no single artifact can be implicated as having usability problems. Rather, usability is a quality that must be seen as distributed across the entire activity in which the artifact is used, including not only the artifacts but also the users, their tacit and explicit rules for operating the artifacts, the division of labor, the community, and so forth. Such a realization pushes us to consider artifacts as members of genres, genres that themselves can be grouped into what I call ecologies, clusters of artifacts which collectively mediate activities. This conception of genre ecologies allows comparisons within and between ecologies, resulting in a more multilayered and complex understanding of usability than other methodologies.

Such an understanding of usability points to a more complex theoretical apparatus than the dominant understandings of usability. In a cultural-historical study of traffic accident location and analysis (Chapter 4), I discuss such an apparatus, the tripartite structure of contradictions, dis coordinations, and breakdowns. In particular, contradictions among different activities and goals can motivate actors to reorganize their activities in various ways: by redividing the labor, enlisting more actors, adapting
and using more genres, and fundamentally changing tools such as information systems. Of special interest is the point that a given information system is always supplemented by other genres. As users adapt other genres to supplement the information system, they import those genres into the relatively stable genre ecology surrounding their work.

When contradictions become severe, software developers develop a new information system, and in doing so they tend to reproduce genres within the interface. Genres that have been adapted by users often migrate to the computer screen, where they mingle with established screen genres to produce hybridized genres. These hybridized genres tend to retain their history, their addressivity, and their relationship to other genres in the ecology. In Chapter 4 I theorize hybrid genres and provide a study of them.

Contradictions give rise to discoordinations, difficulties in interpreting artifacts and managing the actions that those artifacts mediate. These difficulties in turn manifest themselves as breakdowns, points at which a user finds the present interpretation of an artifact to be inadequate for the task at hand. In Chapter 5 I discuss discoordinations and breakdowns extensively and illustrate how the terms can be used to provide a more nuanced notion of usability. Chapter 5 describes observations and interviews that I conducted with users of the second generation of this information system (PC-ALAS). I examine two types of discoordinations, those related to genre perception and genre management, and link them to the contradictions that originate them as well as the breakdowns that are their fruit. I discuss the discoordinations' relation to three activity networks interpenetrating the activity network being studied. And I examine how users deal with discoordinations by adapting genres to distribute work interpersonally and temporally.

As I argue in Chapter 4, when users adapt such genres, developers later tend to migrate those genres to the interface, where they mingle with existing screen genres to produce hybrid genres. Chapter 6 explores the ramifications of hybridization, particularly as it relates to the contradictions among interrelated activity networks. Here, I describe the last two studies, which explore how PC-ALAS users and students familiar with
similar information systems use prototypes of a third-generation system (GIS-ALAS). I then examine the results of these observations, along with those of posttest interviews, in terms of how the information system has changed and what the ramifications of those changes are for discoordinations and breakdowns. I find that hybridized genres are contact points of different activities, activities that sometimes have very different motives and goals. These fundamental differences lead participants to encounter discoordinations and breakdowns, which users often attempt to mitigate by adapting still more genres. Chapter 6 ends by exploring the ramifications of hybridization for the design of information systems.

Finally, the Conclusion summarizes the research methodology I propose, supporting it with findings from the dissertation research.
CHAPTER 1
TOWARDS A THEORY OF ARTIFACTS:
CONSTRUCTING A THEORETICAL FRAMEWORK FOR
INFORMATION SYSTEMS DESIGN AND EVALUATION

In the last several years we have witnessed a startling proliferation of computer-based
information systems that are meant to be browsed by users as part of their larger
activities. Information systems encompass online help, online documentation,
World-Wide Web sites, databases, and the like. They tend to support users’ labor by
providing instruction, data, or details relevant to the users’ tasks.

Information systems have become the object of study of many researchers operating
out of many different traditions. Two dominant strands are those of human-computer
interaction, which focuses on the form of the information system interface, and computer
documentation, which focuses on the content of the information (although it does lend
attention to page design issues). Researchers working from these two traditions have
investigated information system use for the purpose of informing design and evaluation.
Yet these investigations have certain limitations due to certain assumptions about human
cognition that underpin them (assumptions that I address later in this chapter).

One way to deal with these limitations is through constructing a methodology: "the
underlying theory and analysis of how research does or should proceed" (Kirsch and
Sullivan, 1992, p. 2). As Shoshana Zuboff puts it,

    Behind every method lies a belief. Researchers must have a theory of
    reality and of how the reality must surrender itself to their
    knowledge-seeking efforts. Each epistemology implies a set of methods
    uniquely suited to it, and these methods will render the qualities of data
    that reflect a researcher’s assessment of what is vital. (1988, p. 423)

In this case, I argue that a new research methodology is needed, one which takes into
account certain aspects of information system use that have often been passed over by
previous research into information systems: culture, history, society, and interpretation.
Exploring these issues in a systematic way, I argue, can lead to rich insights into information system use, design, and evaluation. This dissertation represents my attempt to develop such a methodology.

The methodology I develop in this dissertation is composed of a set of cultural-historical theories and analyses based on those theories. Since the methodology involves the design and evaluation of particular types of artifacts — information systems — a cultural-historical theory of artifacts must provide underpinnings for the entire methodology. In this chapter I develop such a theory. I suggest that information systems can be studied as artifacts: parts of the material environment that become meaningful to us as they are used in our various activities, "an aspect of the material world that has been modified over the history of its incorporation into goal-directed human action" (Cole, 1996, p. 117). This theory of artifacts should help researchers to avoid at least four problems that exist in current research into information system design and use. I sketch out these problems below, then discuss them in more detail later in this chapter.

**Problem 1: Unit of Analysis.** Currently, researchers in both human-computer interaction and computer documentation tend to focus on the individual as the molar unit of research. Sometimes this assumption is quite explicit in the research design: experiments, for instance, isolate users from their customary environments and tasks. But even in ethnographic research, the researcher often concentrates on the "situated" individual who reacts to circumstances. Such research does not systematically take into account the complex social, cultural, and historical relations among users, artifacts, and activities.

**Problem 2: The Association of Artifacts and Practices.** Since the individual has been taken to be the molar unit of research, and since information system researchers have tended not to examine semiosis in their studies, researchers have tended not to address the question of how cultural-historical practices cohere to an information system. That is, they assume that ways of working with the information system are built into it: the information system affords certain practices. Thus, researchers have assumed that the
information system can be safely studied out of context or can be investigated within traditional ethnomethodological studies. Yet, as I discuss later in this chapter, socially oriented research increasingly suggests that artifacts such as information systems do not have practices "built into" them — these practices are culturally and historically developed and distributed.

**Problem 3: The Forming of Coherent Collections of Practices.** Just as the relationship between information system and individual practices has been relatively unexplored, so have the relationships among clusters of practices that coalesce around the information system. Researchers using an experimental methodology, for instance, concentrate on a few quantifiable factors (typically including speed and comprehension) rather than the coherent strategies that users employ when using the information system. And socially oriented research such as ethnographies tend not to use a strong framework for analyzing and discussing coherent sets of social practices. As I discuss later in this chapter, when clusters of practices are explored, they are frequently discussed as stepwise or hierarchical procedures, or else are not discussed in a systematic framework suitable for detailed analysis.

**Problem 4: The Role of Semiosis.** Certainly professional communicators have discussed the semiotic dimension of the text, and even its page design. And certainly human-computer interaction researchers have discussed the interface in semiotic terms. But a firm line tends to be drawn between the interface and the text it presents; they are rarely explored as a unified artifact with a unified semiotic dimension. Consequently, research that seeks to understand the information system's significance tends to focus on either form or content.

These four problems, as I argue in the rest of this chapter, limit research into information systems which might prove fruitful for design and evaluation. As I note in the next section, each problem has roots in the Cartesian separation between the user and her or his lifeworld. An alternate research agenda could include a methodology predicated on the theory of artifacts that I develop later in this chapter: a theory that can
define a cultural-historical unit of research; account for the relation between artifacts and social practices; account for the relatively stable clusters of practices that form around information systems; and account for the holistic semiotic dimension of artifacts such as information systems. Such a theory provides the underpinnings for a research methodology that takes into account the cultural, historical, social, and interpretive aspects of information system use.

Below, I discuss the current state of information system research in more detail, focusing on the four problems and providing examples. I then explore two theories that have some bearing on this matter — Vygotskian activity theory and Bakhtinian genre theory — and use them as a basis for constructing an appropriate theory of artifacts. I discuss implications for research methodology. Finally, I overview the remainder of the dissertation.

The Current State of Information System Research

Researchers into the design and usability of information systems tend to posit their research either explicitly or implicitly on a Cartesian idea of cognition. Once its central tenet is accepted — Descartes' *cogito ergo sum*, the "I think, therefore I am" which separates the individual mind from the world — various other oppositional dualisms follow: mind-body, organism-environment, actor-society, situation-history, text-context, and form-content, to name just a few. Such dualisms lead researchers to privilege one of each pair over the other. They result in research methodologies that do not always thoroughly explore some of the issues with which rhetoricians are concerned: society, culture, history, and interpretation. For instance, privileging organism over environment results in research that studies the organism (the individual test subject) apart from the environment in which she functions, research that (as we shall see) provides descriptions, evaluations, and predictions that are often limited in their value for information system design and evaluation.

Cartesianism is widespread in information system research. And even researchers who reject Cartesian dualisms have not developed a detailed theory of artifacts to guide
research, despite some important efforts (Carroll and Campbell 1989; Carroll, Kellogg, and Rosson 1991). In this section, I discuss Cartesianism in information system research by focusing on the dominant separation: the separation between form (the domain of human-computer interaction) and content (the domain of computer documentation).

**The Form of the Interface: Human-Computer Interaction in Information System Research**

Human-computer interaction (HCI) is a complex, multidisciplinary field. Yet it is currently dominated by an approach that has been critiqued under the names of information-processing psychology (Carroll, Kellogg, and Rosson 1991; Nardi 1996), cognitive science (Winograd and Flores, 1986), and computational psychology (Bruner, 1990). This approach, which I will discuss henceforth under the name *computational psychology*, tends to reify the Cartesian mind-body split: in this view, mind and machine are essentially similar information-processing devices that are both independent of a body or a world. (See Anderson et al., 1993 and Button et al., 1995 for other critiques of computational psychology in human-computer interaction studies.)

Computational psychology provides a strong basis for HCI research. It offers a detailed theory of mind as well as a well-defined set of theoretical tools for analyzing human-computer interaction. It offers methodologies and methods for conducting HCI research. And its long history in HCI has allowed it to develop into a rich, complex approach with many different variants.

Yet as a whole, computational psychology (as practiced in HCI) has limitations. Some of these limitations are pointed out in John M. Carroll's landmark collection *Designing Interaction* (1991) by human-computer interaction specialists working within a variety of research traditions and theoretical frameworks. In particular, they argue that current methodologies (based on computational psychology) do not provide a strong framework for systematizing some types of observations (Brooks, 1991) and insufficiently address the meaningfulness of artifacts for their users (Bannon and Bødker, 1991; Norman 1991).
Unfortunately, many (if not most) of the HCI studies based on computational psychology adopt a conception of the interface as a form into which content is inserted — content that is considered neutral or inconsequential to the interface itself. For instance, in a study evaluating an information system, Egan et al. (1989) tell us that they have converted a variety of texts to SuperBook format: textbooks, computer documentation, the works of Shakespeare, and the Bible, among others. These works are read for very different purposes and, one might imagine, in very different ways. But the researchers only describe tests using one of the texts, a statistics textbook, and concentrate on quantitative measures of efficiency (proportion of correct responses and mean search times). Is search speed really important to a pastor preparing a sermon from her online Bible? Is the ability to score well on a standardized test relevant to a literature professor who reads *Hamlet* critically? Egan et al. don’t say. Others similarly bypass content to focus on efficiency measures (Hertzum and Frøkjær, 1996; Wenger and Payne, 1996).

Below, I explore and critique the notion that form can be investigated apart from content, focusing on the four problems that I identified earlier. I examine the unit of analysis used in human-computer interaction research; the explanation of how practices become associated with artifacts; the explanation of how coherent practices are formed around artifacts; and the role of semiosis.

*The Unit of Analysis*

As I mentioned above, computational psychology provides a well-defined set of methods and methodologies for studying human activity. In human-computer interaction, empirical research based on computational psychology tends to involve controlled observation of individuals performing specified and constrained tasks with a computer. The observational data are typically quantified and analyzed statistically. The analyses are often grounded in the assumption that participants’ minds are essentially alike, and that if the environment is sufficiently controlled, the minds of all participants should function in essentially the same way — no matter the differences in culture, history, or
activity. Thus the unit of analysis in computational psychology is the individual mind, the information-processing device divorced from body, world, and history. According to Adler and Winograd's critique of usability studies, the user is seen as a "system component with a particular repertoire of actions and potential for breakdown" (1992), a cog in a larger system rather than an actor who uses the information system to support his or her various activities.

Cartesian separations between the individual user and his or her cultural-historical environment are problematic because they focus on the user-machine dyad rather than on the cultural and historical aspects that make human-computer interaction meaningful. This point is brought home by a counter-tradition of non-Cartesian research that focuses on culture, history, society, and interpretation (a tradition that provides much of the basis for my work). These researchers have found that outside the usability lab, users typically communicate with each other as much as they interact with their machines (Bellamy, 1996; Heath and Luff, 1991; Nardi, 1993). Furthermore, the way users approach their machines is affected by the activities in which they engage (Button, 1993; Dourish, ')

Many experiments do take into account factors that appear to be cultural-historical, such as level of experience. Yet these factors are treated as such — quantitative factors that can be isolated from the working of the mind and controlled for. I argue throughout this dissertation that such an isolation is not possible: the mind is not simply affected by experiences such as education, it is in part defined or instantiated through them.

Although computational psychology dominates HCI studies, some researchers have turned to other approaches, such as situated cognition. Empirical research based on situated cognition tends towards ethnomethodological research such as ethnographies, stimulated recall interviews, and the like; data are typically analyzed via discourse analysis, protocol analysis, or other social science methods. Yet in many cases the unit of analysis is still the individual, either decontextualized or performing within a particular situation (a setting bounded by time and location). For instance, Suchman's (1987) ethnomethodological study of photocopier use critiques computational psychology's notion of mind as information-processing device, but exchanges it for the notion that individual actors or dyads find themselves in various situations and make up strategies as they go along. Thus it does not explore the historical connection between actor and environment. (See Nardi, 1996c for a critique.)

Although not in the context of human-computer interaction, the 1998 exchange between Peter Smagorinsky (a representative of the Vygotskian cultural-historical tradition) and K. Anders Ericsson and Herbert A. Simon (representatives of information-processing cognitive psychology, or IPCS) is particularly instructive. Smagorinsky suggests that IPCS does not adequately take into account culture and history; Ericsson and Simon insist that it does because culture and history are both incorporated into the individual's memory. Computational psychology envisions memory as a sort of peripheral device that feeds information into the central processor of the human mind.
1995a), the artifacts with which they already have experience (Bødker, 1991; Carroll, 1991c), and the organizations in which they use the technology (Clement and Gottlieb, 1987; Orlikowski and Gash, 1994; Sellen and Harper, 1997).

In response, some human-computer interaction researchers have abandoned the individual as a unit of research. Instead, they are attempting to design systems that facilitate user-user communication (Fischer et al., 1992; Kaplan, 1990; Terveen, Selfridge, and Long, 1995); support tasks that are grounded within a particular cultural-historical milieu (Bødker and Grønbæk, 1996, Greenbaum and Kyng, 1991); and scaffold users from previous artifacts to new ones (Erickson, 1990b).

The Association of Artifacts and Practices

As we have seen, artifacts are used in joint activity by users and are interpreted as being meaningful within those activities. Users of information systems jointly form practices that they use to deal with these artifacts. Thus we need an account of how practices and artifacts come to be associated.

Human-computer interaction researchers have produced various accounts based on computational psychology. These tend to provide strong frameworks for describing individual use, particularly at the level of ergonomics. Yet few provide a strong framework for the study of cultural-historical issues in computer use.

For instance, some approaches stress affordances (rules or possibilities built into the design) (Norman, 1988) and posit mental models (mental reproductions of the artifact in which the user internally rehearses the artifact's affordances). Affordances and mental models, however, are limiting in that they focus on the individual's understanding of the possibilities inherent in the artifact-in-itself, abstracted from the activities in which it is used. For example, Carroll and Olsen stress that a mental model of an artifact is a "process which operates on the inputs [the information entering the mind of the individual user] to produce outputs [transformed information that guides the user's actions]" (1988, p. 50). The result is that the individual can "mentally try out actions [using the artifact] before choosing one to execute" (p. 51). Mental models, then, are
internalist processes that reside in the individual and center on the artifact rather than the activities in which the artifact is used. Yet naturalistic research into human-computer interaction (Bellamy, 1994; Nardi, 1993; Suchman and Trigg, 1993) suggests that artifacts are encountered, used, and perceived as meaningful in joint activities — activities in which communities of users have evolved practices to govern individuals’ use of the artifacts. For instance, spreadsheet users in Nardi’s (1993) cultural-historical study used spreadsheets not just to calculate numbers, but also to provide each other with examples; the spreadsheets took on an added layer of meaning as they were used in these joint activities.

*The Forming of Coherent Collections of Practices*

Not only do users associate artifacts with practices, they bring coherent collections of practices to artifacts. Many researchers have assumed that the artifact itself is the nexus of the collection of practices, that is to say, that discrete practices are related *simply because they happen to address different features of the same artifact*; the practices in themselves have no relation to each other. Such a view is Cartesian because it assumes a necessary structural connection between artifact and practice, essentially a stimulus-response connection, and this view leaves little room for the complex interpretation that goes on in information systems use.

For instance, it is easy to assume that software developers read program code in the same way that a computer compiles the same code: through scanning each line and matching its formal elements with predefined rules. The form (the code itself) has a one-to-one association with the content (the rule for interpreting the code). That is, each programmatic element is matched with its machine language equivalent in a more or less serial process; each element is decoded through simple pattern-matching.⁴ Yet research into program comprehension and production over the last fifteen years suggests that programmers tend to read code cyclically, forming hypotheses about the program’s

---

⁴Program compilation is actually a bit more complex than a simple serial process, and varies from language to language. But in its essence this explanation still obtains.
meaning through complex and flexible strategies they use to interpret textual features (e.g. Brooks, 1983; Soloway and Ehrlich, 1984). These strategies do not involve a simple one-to-one correlation between artifact feature (a command in the code) and practice (a single rule associated with the command). Rather, the interpretive strategies are applied as flexible complexes that allow users to perceive and interpret familiar programming habits stretching across multiple lines of code.

Thus, although practices do tend to coalesce around an artifact, they also cohere to each other and are transferable to other artifacts with quite different material makeup, used in quite different activities. For instance, Orlikowski and Gash (1994) note that new users of groupware tended to conceptualize it in terms of familiar artifacts: email programs, typewriters, standard databases. In associating the new artifact with other artifacts with which they had experience, the users applied sets of practices to the new artifact. Those sets of practices tended to govern and constrain how the users employed the groupware. And since collections of practices do tend to govern and constrain how users approach software artifacts, information system researchers need to account for how collections of practices form at both the community and individual level, cohere to each other, and co-develop.

*The Role of Semiosis*

Another consequence of computational psychology’s Cartesian stance is its undeveloped account of the semiotic treatment of artifacts. If human-computer interaction is taken to be a meeting of two essentially similar information-processing devices with inherent hardwired characteristics, communication more or less entails getting the output and input to synchronize. This variation on the conduit model of communication sidesteps the problem of interpretation altogether. Artifacts such as information systems are assumed to be things-in-themselves, forms that (when properly structured) unambiguously carry the content of the message. Even when a different semiotic framework is brought in to explain user interpretation of interfaces, it is often reconceptualized in Cartesian terms. For instance, de Souza (1993) uses a framework
based on Eco's theory of sign production, but the framework is implemented as a conduit model in which the sender "encodes" a message and the receiver "decodes" it. That is, meaning (content) is encoded in the text and assumed to be independent of the interface (form). The roles of society, culture, history, and interaction are left largely unexplored.

Yet artifacts do not seem to be unproblematic: users carry certain assumptions about artifacts; interpret them; and apply practices to them based on these interpretations. Differences in interpretations can result in qualitatively different artifacts: In Orlikowski and Gash (1994), for instance, two groups working within the same large organization using the same groupware had radically different interpretations of that groupware, and therefore used it in quite different ways.

Interpretation is key to understanding artifact use because it provides a way to understand how practices are applied to an artifact. But computational psychology-based HCI research tends to analyze interpretation at the level of an individual processing information rather than a group or community forming interpretive strategies.

A non-Cartesian stance would necessitate viewing these artifacts as meaningful only within a certain cultural-historical milieu: they are not separable from the environment in which they are produced, interpreted, and used.

As we have seen, the four problems have roots in the Cartesian separation between user and lifeworld, a separation that underlies computational psychology's dominant metaphor of mind as machine. The four problems have caused various difficulties for human-computer interaction studies based on computational psychology. In the next section, I demonstrate that the four problems also manifest themselves in computer documentation studies.

The Informational Content: Computer Documentation in Information System Research

Human-computer interaction researchers working in the tradition of computational psychology tend to address the form of the information system interface, leaving the informational content to professional communicators in specialized fields such as
computer documentation. These professional communicators, in turn, tend to be happy with (or at least accepting of) this arrangement. Computer documentation research in information systems has traditionally been concerned with form primarily in terms of page design choices: "chunking" information into paragraphs, using white space, using lists, selecting readable fonts, and inserting suitable graphics (Brockmann, 1990; Carliner, 1990). Yet the software that "presents" or "delivers" the informational content is not generally considered to be part of the professional communicator's bailiwick.

Computer documentation research into information systems tends to follow a more or less formalist tradition. In its more unsophisticated variants, formalism entails understanding the text as a sort of code that ciphers meaning. Anyone who can decode the text can receive the meaning. Consequently, form is conceived of as a factor that influences the clarity of the message, but not the message itself. Just as computational psychologists sometimes consider content as inconsequential to the use of the interface, formalists sometimes consider the interface to be inconsequential to the understanding of the content.

The formalist tradition has made substantial contributions. In particular, it has underpinned important research into issues such as style, clarity, and page design. Yet its communicative model limits its potential for guiding research into information systems design and evaluation. Below, I criticize the formalist approach to information systems in terms of the four problems I have identified: the unit of analysis used in computer documentation research; the explanation of how practices become associated with artifacts; the explanation of how coherent practices are formed around artifacts; and the role of semiosis.

The Unit of Analysis

Like human-computer interaction researchers, professional communication researchers tend to use the individual as a unit of analysis. This is a natural outgrowth of the formalist conception of language that drives much research in this field. Since formalists tend to conceive of text as encoded meaning that individuals decode when
they read, it follows that any properly informed individual, under similar conditions, should get more or less the same meaning from the text.

The formalist understanding of text is reflected in much of the literature on usability testing for the computer industry. For example, much advice on conducting a usability study assumes that observing an individual performing a prespecified reading task is sufficient for determining usability. Thus researchers such as Grice and Ridgway (1991) and Dumas and Redish (1994) give detailed directions on how to set up a usability lab (which isolates the individual user from the customary work environment) and how to monitor and quantitatively measure tasks (which denatures tasks that normally have a place in the user’s larger activities) — but they give little advice on how to perform naturalistic observations. As long as the lab and the task are similar to what users might encounter in their daily work, they are regarded as sufficient for accurate usability testing.

Yet the very act of isolating the user changes the task considerably. As Sellen and Harper (1997) point out in their study of document use at the International Monetary Fund, readers of workplace documents tend to use these documents to support their collaborative activities. And even in customarily individual tasks, readers do not always stay on task — they juggle various tasks and even find themselves redefining tasks.

These limitations of traditional, lab-centered, individual-based usability testing have driven some researchers to take a more naturalistic approach to research. For instance, contextual inquiry (Holtzblatt and Beyer, 1993; Raven and Flanders, 1996) involves gathering information on users’ tasks for design and observation. However, such ethnomethodological research approaches tend to underdefine the notion of task (Mirel, 1996a), making tasks harder to identify and study systematically. Such approaches also tend to oversituate the research, producing "a snapshot in time" rather than a developmentally oriented and user-centered understanding of the user-artifact interaction (Hackos, Hammar, and Elser, 1997). It’s difficult to generalize the findings of such research across organizations and periods of development.
The Association of Artifacts and Practices

In computer documentation studies, the association of artifacts and practices is explored using various devices: analogy (Lieberman, 1991), examples, metaphors, and cases (Brockmann, 1990). All of these devices are used to explain how past practices are applied to the information system at hand. However, such devices provide no way to systematically explore the artifact-practice relationship and its development. For instance, even when metaphor is explored in a nonformalist fashion (Selber, 1995; Selfe and Selfe, 1994), it does not yield insights detailed enough to provide specific recommendations for information system design and evaluation. Neither does it explain the development of the artifact-practice relationship. Metaphor might allow us to predict that users will regard an information system's "table of contents" as similar to a book's, but it does not explain how users begin to treat the "table of contents" in ways different from a book's table of contents; it does not allow us to track or predict the development of practices. Finally, metaphoric approaches typically do not take into account the changing relationship between the representation and the user. An interface that features a "desktop," for instance, may be seen as a representation of a real desktop — but eventually the user begins to see the interface as simply an interface rather than a substitute for something else. Metaphors "die" (Davidson 1984, p.261).

The Forming of Coherent Collections of Practices

Finally, current research into writing for information systems tends to provide only a few simple conceptual tools to describe the coherent clusters of practices that coalesce around information systems. One is the notion of stepwise tasks, which I have already critiqued in my discussion of computational psychology. Another is that of schema, a concept that shows promise but that has thus far been used only sporadically in computer documentation research (e.g. Lieberman, 1991).

Another, less formalist way to conceptualize collections of practices is through the notion of rhetorical strategies. For instance, Mirel (1996b) found that users developed
rhetorical strategies that inform how they search for, retrieve, and report electronic data — strategies that go beyond surface "decoding." Others have similarly suggested that online information requires new rhetorical strategies (Bolter, 1991, Landow, 1992). Yet the conceptual tool of rhetorical strategies has limitations. (1) It focuses almost exclusively on communicative acts. (2) Consequently, its object is nearly always a text: researchers examine texts and ways of creating texts, but they tend not to examine nontextual artifacts in terms of rhetorical strategies. (3) Thus, it has no tradition of exploring acts and artifacts that are not strictly seen as communicative. For instance, if we are to explore information systems such as the database discussed in Chapter 5, we have to account for how users employ mouse cursors as well as database reports; checkboxes as well as labels; and coffee cups as well as maps. These artifacts do not all share what might traditionally be considered communicative or rhetorical aspects, yet they all figure in the use of the information system.

The Role of Semiosis

Much of the research into writing for the computer industry conceives of the computer — and the information system — as a "neutral platform" (a form) that carries but does not affect the writing content (see Porter and Sullivan, 1996 for a critique). Consequently, these researchers suggest page design changes when revising paper-based information for information systems. For instance, Carliner (1990) suggests that "computer-based information needs its own style" because the computer display is different from paper and because reading the interface is a "different experience" from paper: it is interactive and intangible. There is no discussion of the activities or the social, cultural, and historical influences that might cause such a style to be useful. The problem is conceived of as reconfiguring formal elements. Literature on online documentation is rife with such suggestions (Brockmann, 1990; Petrauskas, 1991; Semple, 1991).

\[^{5}\text{Contrast Carliner's understanding of style with that of Bazerman (1988), who examines the historical reasons for APA style and critiques the cultural-historical assumptions embedded within that style.}\]
The formalist conception of text-as-code reflected in such research is contradicted by other, more socially situated research into how users approach online information. For instance, O'Hara and Sellen (1997) found that paper documents were more useful than online documents for the task of summarizing articles, but the reason had little to do with formalist concerns such as screen resolution, contrast, or viewing angle. Rather, the advantages lay in factors such as ease of annotation and flexibility of spatial layout: factors that are specific to the activity in which the users are engaged. Usability cannot simply be measured, then, by how little the medium interferes with the transmission of the text.

In the sections above, I have critiqued approaches to information system design and evaluation, approaches based on computational psychology and formalism. Although these approaches have advantages, the Cartesian assumptions that underlie them give rise to the same four problems. In the following section, I argue that these problems can be addressed through a non-Cartesian theory of artifacts.

**What We Need from a Theory of Artifacts**

What information system researchers need is a theory of artifacts that covers the ground I have described — a theory that surfaces the cultural, historical, and semiotic dimensions of the artifact, rather than seeing the artifact as a thing-in-itself. This theory should articulate a view of artifact as a portion of the material environment segmented from the rest of the environment and made meaningful through the activities of its users. The theory should (1) propose a cultural-historical unit of analysis beyond the individual; (2) account for the association between artifacts and social practices; (3) account for the coherent collections of practices that coalesce around artifacts; and (4) account for the semiotic dimension of artifacts.

Thus far, theories that address all of these points are rare. The most impressive of these is Michael Cole's (1996) detailed account of artifacts, based on a synthesis of Deweyan, Hegelian, and Marxian thought. The theory I advance below has strong affinities with Cole's, but draws from different sources and focuses more consciously on
accounting for coherent collections of practices. Below I describe my theory of artifacts, which is based on two theoretical traditions that have separately been used to critique computational psychology and formalism: Vygotskian activity theory and Bakhtinian genre theory. I discuss these traditions, then synthesize them to produce a theory of artifacts. (Other Bakhtinian-Vygotskian syntheses exist [e.g. R. Engeström, 1995; Russell, 1995, 1997; Spinuzzi, 1996], but these have not been elaborated in such a way as to support the theory of artifacts I envision.)

Vygotskian activity theory and Bakhtinian genre theory have both been applied to human-computer interaction and to professional communication. In this section, I review the two theories and discuss how they might be united to provide a framework suitable for the study of artifacts: activity theory provides a suitable unit of analysis and a framework for investigating the relationship between artifact and practice, while genre theory deals with the problem of systematically investigating coherent collections of practices and the problem of semiosis. Particularly of interest to me in this section is how the two theories might contribute to meeting the four characteristics that I have identified as being necessary to a non-Cartesian theory of artifacts.

Activity Theory

Activity theory is a cultural-historical psychology concerned with labor activity. The particular strand of activity theory I am using stretches from L.S. Vygotsky’s work in the Soviet Union in the 1930s, through A.N. Leont’ev’s work in the 1940s - 1970s, to the work of Yrjö Engeström’s group in Scandinavia and North America in the 1980s and 1990s. In this strand, activity is conceptualized as inherently social and intricately bound up with joint (community) labor.

Activity theory provides two of the four things we need from a theory of artifacts for information systems design and evaluation: a suitable unit of analysis and an account of how artifacts and practices are associated. In this section I discuss these two contributions, drawing primarily on Vygotsky, Leont’ev, Engeström, and Cole.
Unit of Analysis

A central concept in activity theory is the activity network. Activity theory posits that in every sphere of activity, one or more subjects use mediational means (that is, either physical or psychological tools) to transform a particular object with a particular outcome in mind (Figure 1). For instance, a developer of World Wide Web pages may use various tools to transform an older set of Web pages into a newer set. She does so to achieve various outcomes.

Yet the developer’s activity does not spring from herself alone. It is intimately related to her community, the company where she works; her relationship to the community is mediated by rules, such as the standard features to be included in a webpage. Similarly, the relationship between the community and the object is mediated by the division of labor within that community: web page developers develop websites, often helped by software engineers, marketers, graphic designers, managers and others in various ways. So the developer’s job is intimately tied to the cultural-historical sphere of human activity in which it is performed (Figure 1.1).

A word about rules is in order here. Rules are recurrent, socially distributed actions that activity networks have developed to govern subjects’ handling of artifacts within the activity. These rules are, then, bound to the artifacts within the activity. But they are not bound in the sense of being built into the artifacts (as affordances): rather, they are culturally-historically bound.

For instance, Norman (1988) explains affordances by pointing out that a door "affords" either pushing or pulling (depending on which side you stand on). If you stand on the side of the door that affords pulling, then pulling is inherently the rule that you should apply to the door; actions such as pushing or sliding the door are simply the wrong rules to apply. Yet an activity theory analysis of the door might point out that

6 In much activity theory literature, this unit of analysis is referred to as an activity system. However, the term system implies a rigid hierarchical structure, whereas activity theory postulates far more fluid and contingent relationships within the activity. I follow Russell (1995) by using the term activity network, a term with shadings of Latour’s actor-network theory.
rules such as pulling and pushing are developed in a culture that has to deal regularly with doors. We can easily think of cultures which have no experience with doors, and for which the rules of pushing and pulling doors simply do not exist — for whom a closed door would not be qualitatively different from a wall. We can also think of people within our own culture who might, within certain circumstances, simply ignore the affordances (e.g. firefighters who break down the door to get to a fire).

Such instances are even easier to find when artifacts are less stable than doors — that is, when we study a class of artifacts that is developing at a relatively rapid pace, meaning that new rules are being learned and distributed far more frequently. For instance, Mackay (1995) notes in a study of an electronic mail program that although the program did not afford certain actions, users exploited a bug to carry out those actions. The new rule was so successful that the program's designers turned the bug into a regular feature in the next release. In this case, users made an affordance by exploiting a bug; communicated that affordance to each other, distributing the affordance among the

Figure 1.1 An activity network. Lines indicate avenues of interaction among elements in the system. (Adapted from Engeström, 1990.)
members of the community; and convinced the designers to institutionalize that affordance.

Rules govern both mediating tools and objects. For instance, a web designer might transform her object — a webpage — by following rules of web design: use hyperlinks to link to relevant information; use headings to announce document structure. Her labor is mediated by a webpage editor, which is also governed by certain rules: click on the "Link" button to turn the selected area into a link; use the <H1> style to designate a first-level heading. These rules are not meaningfully separable from the artifacts they govern. Indeed, it is only through applying rules that users can recognize artifacts in the first place. I discuss these notions more fully below.

The activity network provides a suitable unit of analysis for a theory of artifacts. Below, I explore two salient points of this unit: its object-orientation and its emphasis on development.

**Object-Orientation**

Every activity is oriented towards an object, something that the agents seek to transform through their activity. The object gives the activity "a determined direction. ... the object of an activity is its true motive" (Leontyev, 1981, p. 62). That object is recognized as existing and being shared by others in the world. Objects can be material (and in some aspect, at least, they always are), but they are also cultural-historical: they mean nothing outside of a particular cultural-historical milieu in which they are used. Finally, the transformation of an object is cyclical: an activity is aimed towards repeatedly transforming an object, with the aim of producing an outcome.

For instance, the web developer's object is her website. Her activity as a developer is oriented toward transforming that site: designing it, writing it, debugging it, upgrading it to the next version. Furthermore, the website is not transformed only once. Rather, it will be transformed time and again by the developer as new features are needed or new bugs are found.
Whereas computational psychologists tend to view artifacts as things-in-themselves, activity theorists consider them to be socioculturally apprehended. In this view, our relations with the "world of objects" is mediated by our relations with other persons and with society (Leontyev, 1981, p. 296): the same object may be approached and transformed differently (and with differing results) depending on the subject’s understanding of it. For example, A.R. Luria found that illiterate peasants had no difficulty solving simple mathematics problems if they were discussed in the context of the peasants' daily activities, but the peasants could not solve the same problems if they were presented as school problems (Luria, 1976). Luria also found that level of education affected whether the peasants could see optical illusions. The way we apprehend and treat objects, then, is affected by the rules or habits we have acquired for apprehending and treating them. Those rules are learned through our development in a culture and cohere to the objects. Objects are not separable from the rules that cohere to them: it is through these rules that objects become meaningful. For instance, in Luria’s study, educated peasants had learned interpretive strategies that made optical illusions meaningful to them; uneducated peasants had not learned these strategies. The pictures, though materially the same for both groups, were qualitatively different artifacts.

Finally, objects have histories. As an object changes, the activity (which is oriented toward that object) also must change. For instance, the web developer might first be engaged in the activity of initially planning and implementing a website. As the site takes form, her activity might shift to maintaining it, and later to upgrading it.

**Development**

Activities are continually changing, shaped by historical development. Activity theory reminds us that the "world of objects and phenomena" in which we find ourselves is not static (Leontyev, 1981, p. 294) — it is in constant flux as various parts of the activity network change and as the cultural-historical milieu changes. Certainly the object and tool can change and the subject can develop new skills and new goals. But rules, community, and division of labor can also change. Because of this constant flux,
human interaction with the material environment must be considered developmentally. As Kaptelinin and Nardi argue, "It is important to understand how tools are used not in a single instance of trying them out in a laboratory (for example), but as usage unfolds over time" (Kaptelinin and Nardi, 1999) — and, of course, as changing conditions affect usability.

For instance, suppose our webpage developer switches webpage editing programs. She does not do well at first, but eventually she learns how to use the new editor and she becomes quite proficient. The program, then, cannot be measured for some sort of abstract usability at any single point: at one stage in its development it is less "usable" than at other times.

Accepting the principle of development means committing to observe the development of the artifact, the activity, and the users’ skills with the artifact in that activity. The cultural-historical activity network is activity theory’s unit of analysis. It has three advantages:

(1) Instead of focusing on the decontextualized individual, activity theorists take the activity network as a unit of analysis. Consequently, connections among subject, community, and object are seen as integral and developmental.

(2) Artifacts’ uses are forefronted: rather than supposing artifacts to be things-in-themselves, activity theory recognizes them as deriving their characteristics from the social-cultural milieu in which they are apprehended and used.

(3) Consequently, I argue, an activity theory analysis of artifact use should provide more thorough and useful explanations of artifacts, and perhaps better predictive power than computational psychology has thus far given us.

The Association of Artifacts and Practices

Activity theory also has much to say about how artifacts and practices come to be associated. Below, I discuss three principles of activity theory that are relevant to this point: mediation, internalization/externalization, and hierarchical structure of activity.
Mediation

Tools mediate activities; they mediate between the subject (both individual and collective) and the object. Indeed, according to Vygotsky, the tool "is externally oriented; it must lead to changes in objects" (1962, p. 55).

Like objects, tools are not socially neutral. As Leont'ev points out, "handling" a tool entails far more than merely holding it: "handling" implies knowing how to use it in specific ways for specific activities (Leont'ev, 1981, p. 216). A tool, Leont'ev says, carries within itself a certain socially developed mode of action; the hand wielding the tool is subordinated to the tool's mode and its socially developed system of operations is "fixed" in the tool (p. 216-217). It embodies socially developed rules. The subject appropriates the tool's operations by developing his or her own abilities.

But this is not to say that tools deterministically affect the activities in which they are used. Activity theorists do not deal with universal affordances. Rather, a tool embodies modes of action when it is encountered within a specific cultural-historical milieu and used with a particular objective. Under these conditions it will tend to reproduce certain ways of acting in the user. In other words, it is unproductive to draw a clear distinction between the tool and the rules that govern it within a specific activity (see also Bødker, 1997).

For instance, suppose that the web developer is using a tool (a program) that is set up to help her create webpages. The tool encourages certain actions that comply with the implicit rules governing webpages. A button bar includes buttons to help her accomplish certain things easily (format lists; make text bold or italicized), but does not include buttons for less common actions (e.g. embedding sound files), making those actions more difficult. Thus tools structure as well as implement the activity.

Tools, like objects, are not static. In fact, tools and objects are mutually co-constructive. Changes in objects necessitate changes in tools, and vice versa. For instance, in Mackay's (1995) study on the email program that I mentioned earlier, users worked with a particular email program that filtered email messages before they were
read. But the users discovered that the debugging feature would let them filter their mail after they read it. The new use of the debugging feature was popular enough that it was turned into a user "feature" in the next release. Consequently the email program became more popular. In this example, users' objects changed: they needed a higher level of control over their mail filtering. They found a way to effect this control, and in doing so managed to change the tool. Similarly, access to easier webpage editing tools has contributed to an explosion in the number of webpages. This explosion has resulted in increased attention on web design, which in turn necessitates more sophisticated editing tools.

Tools are not things-in-themselves: they are not meaningful apart from the subject who uses them and the activities in which they are used, including the rules that are applied to them. When a webpage editing tool is used in other ways — to view others' webpages, for instance, or to write a quick memo — the editor is used (in a sense) as an entirely different tool, mediating an entirely different activity and being governed by different rules.

**Internalization/Externalization**

Although the terms "internalization" and "externalization" sound Cartesian, they are meant to express non-Cartesian concepts: the forming and development of a consciousness within a cultural-historical environment. Below, I discuss how artifacts might be internalized (that is, how the subject develops an understanding of these artifacts as inextricably bound to her labor, and in doing so, qualitatively transforms her relationship to the artifacts) and how, once internalized, those artifacts might be externalized again (that is, how the subject expresses the artifact's workings or recreates the transformed artifact for specific purposes).

The mediational means typically begins as a physical tool. But as the subject continues to use that tool in her activity, she begins to internalize the activity and the tool that mediates it. That internalization is
a transition that results in processes external in form, with external material objects, being transformed into processes that take place on the mental plane, on the plane of consciousness; here they undergo a specific transformation — they are generalized, verbalized, condensed, and most importantly, they become capable of further development, which exceeds the boundaries of the possibilities of external activity. (Leont'ev 1978, p. 58)

For instance, the webpage tool is at first an external, physical tool to the subject. But as she begins to learn it, she finds that she can envision what certain effects of her actions might be before she makes them. This understanding, unlike a mental model, is object-oriented. Included in this understanding are the activities in which the tool is used, the socially developed rules that define it, the specific actions the subject takes, and so forth.

Internalization is not the transference but the qualitative transforming of an activity. In being transformed, the activity is no longer simply mediated by a physical tool (the webpage tool). It is also mediated by the developer's knowledge of the webpage tool. The developer's internalization of the editor will be shaped by the uses to which she has put the editor, that is, her developmental history with it. For instance, if she routinely creates new pages, she will be more familiar with the procedures that go into setting up a new page. If she spends most of her time editing existing pages, she will learn the editor's commands for opening existing files.

The internalized webpage tool has another important characteristic that the physical tool does not: whereas the physical tool is used to change the developer's environment, the internalized tool can also be used by the developer to regulate herself. She uses it to judge what is or is not possible and chooses certain courses of action based on those judgments. The internalized webpage tool, then, "possesses the important characteristic of reverse action (that is, it operates on the individual rather than the environment)" (Vygotsky, 1978, p. 39).

Just as internalization involves converting external actions to mental processes, externalization involves the re-conversion of mental processes to external actions.
"Mental processes manifest themselves in external actions performed by a person, so that they can be corrected if necessary" (Kaptelinin, 1996a). Externalization often occurs "when an internalized action needs to be repaired or scaled" (Kaptelinin and Nardi, 1998). For instance, people who have internalized a system of mathematics might still have to resort to pen and paper when performing difficult calculations, and programmers may still have to write pseudocode (e.g. sketches on whiteboards) when constructing a complicated algorithm. Yet the externalization bears the marks of the qualitative changes it underwent during internalization: pseudocode, for instance, is very different from the computer code that the programmer had originally internalized.

Externalization is also important when a group of people must collaborate in the use of an artifact. For instance, researchers have pointed out the uses of pseudocode in coordinating programming among members of programming teams (Bellamy, 1994; Suchman and Trigg, 1993). Similarly, a webpage developer might teach a novice how to use a webpage tool by describing, charting, and demonstrating its functions. She teaches the rules for using the tool by externalizing her previously internalized activity.

Hierarchical Structure of Activity

Activity is hierarchically structured at various functional levels. Leont’ev distinguishes among three levels: activities, actions, and operations (Table 1.1).

<table>
<thead>
<tr>
<th>Level of Activity</th>
<th>Governed By</th>
<th>Time</th>
<th>Automaticity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Motive</td>
<td>Indefinite</td>
<td>Usually unconscious</td>
</tr>
<tr>
<td>Action</td>
<td>Goal</td>
<td>Definite</td>
<td>Conscious</td>
</tr>
<tr>
<td>Operation</td>
<td>Conditions</td>
<td>Definite</td>
<td>Unconscious (habitualized)</td>
</tr>
</tbody>
</table>

(The levels of activity in Table 1.1 will become quite important later in the dissertation, when I discuss breakdowns.)
Activities are undertaken to fulfill certain motives, or general objectives around which the activities are structured. As we have seen, activities are oriented towards transforming a certain object; the motive is that for which the community strives as it works to transform the object. Activities are typically undertaken by an entire community and are not always conscious.

For instance, the development of a website is an activity. Yet the people involved in that activity — webpage developers, graphic designers, managers, and so forth — are not always conscious of the overall activity. There is a division of labor: the people have different jobs in the complex labor of development, and are concerned with these specific jobs. Developers are concerned with individual pages, graphic designers are concerned with graphics, and technical support personnel are concerned with the servers on which the webpages are stored. None of these subjects are always, constantly thinking of the overall objective (the new website) — even managers, whose job it is to oversee the division of labor, often concentrate on more localized actions (scheduling meetings, writing memos, making telephone calls, etc.).

Actions are consciously undertaken to fulfill certain goals, or localized objectives that are fulfilled as part of the general activity. As Leont’ev tells us, "An act or action is a process whose motive does not coincide with its object (i.e. with what it is directed to) but lies in the activity of which it forms part" (Leontyev, 1981, p. 400). For instance, the web developer may perform actions such as preparing a directory for her files, starting the webpage editing program, and loading a file. These actions do not directly affect the activity’s object, but they are necessary links in the chain of actions that makes up the activity (Leont’ev, 1978, p. 64).

Operations are carried out in response to conditions, or specific configurations of the work environment. Leont’ev states that an operation is "the mode of performing an act" (Leontyev, 1981, p. 406): an unconscious step in carrying out an action within certain conditions. The notion of operations is necessary because even if an action is the same in
two different scenarios, if the conditions are different, the subject might automatically use operations that respond to those conditions.

For instance, suppose that the web developer wants to save a file. Let's consider two different scenarios. In the first scenario, she has just finished using her mouse for some other action. Her hand is still on it, so she automatically uses it to select Save from the File menu. In the second scenario, she has just finished typing text. Both hands are on the keyboard, so she automatically presses a key combination to save the file. In both scenarios, the action was the same: she saved the file. But conditions are different, and those conditions influence the operation she carries out to perform that action.

Every operation is a transformed action. That is, the web developer once had to consciously learn how to save the file using her mouse and how to save it using a key combination. As she used these actions over and over again, and as she included them as part of her larger actions (e.g. exiting the webpage tool), they became automatic operations. Similarly, operations can become actions again: users can become conscious of them.

Notice that as I mentioned in the last sections, rules are learned and associated with artifacts. The levels of activity give us a vocabulary for describing how these rules have been assimilated and have become part of the user's operations.

Activity theory, then, provides an account of the artifact-practice relationship. This account has four advantages over the ones we have seen thus far.

1) By explicitly theorizing relations among rules (practices) and tools and objects (artifacts), activity theory recognizes their connections.

2) Artifacts are seen as interpreted and used according to socially distributed rules, not as simply things-in-themselves.

3) Artifacts are acknowledged to "carry" rules, but these embedded rules are not unproblematically built into the artifact, as affordances are often conceived. Rather, they are embedded in the milieu that includes the artifact, user, and activity.

4) Interpretation is recognized to be essential to an account of artifact use.
What Activity Theory Cannot Provide

Although activity theory provides a valuable unit of analysis and a detailed account of the artifact-practice relationship, it makes only limited headway on the other two problems.

Activity theory provides only a weak account of how rules cohere to each other. Certainly activity theory’s hierarchical levels of activity give us some help here: activities are made up of actions, which themselves are made up of smaller actions and operations, and all of these are instantiations of rules. But this is not the whole story. The levels of activity tell us how rules might be sequentially or hierarchically related, but not how nonsequential or nonhierarchical rules cohere. For instance, our developer knows the unwritten rules that web pages should have headings, blue hypertext links, lists, and tables. If the object is a webpage, all of these rules apply, even though they are not sequentially or hierarchically linked to each other — there is no set order or procedure for ensuring that all of these rules are met.

Activity theory avoids the mistake of assuming that the artifact itself is the nexus of the practices that surround it; rather, an activity theory approach would see the nexus as being the activity in which the artifact is used and in which it becomes meaningful. Yet activity theory does not thoroughly explore how practices cluster around activities and form relatively stable, coherent collections.

Activity theory does not provide a well-developed account of the semiotic dimension of artifacts. Although it allows for such a dimension, it does not provide a detailed framework for the genetic analysis of signs.

These two weaknesses can be shored up with Bakhtinian genre theory.

Genre

In the Bakhtinian conception, utterances — which are artifacts charged with meaning (such as books, newspapers, the spoken word, paintings, information systems) — are produced and interpreted according to collections of socially developed and shared
habits, habits that have sprung up in response to recurrent situations (Morson, 1986b; see also Voloshinov, 1973; Bakhtin, 1981; Bakhtin, 1984; Bakhtin, 1986). These relatively stable collections of habits are called *genres*. The Bakhtin Circle applied genre mainly to literary works, but Bakhtin makes clear that it can be applied to other artifacts as well (Bakhtin 1986). One example is a Windows online help file, which would tend to have certain features (hypertext links that are displayed as green and underlined; buttons to navigate the different screens; an introductory page which lists the major topics) with which users are familiar, and which they can use to understand and operate the help system. These features are material symptoms of the genre of Windows online help, a genre to which this particular help system belongs.

Like the notion of rules explored in activity theory, the notion of genre presupposes that practices are not separable from the artifact that is governed by them. The rule-tool relationship is explored as a unit. However, unlike activity theory’s account of the artifact, the Bakhtinian account investigates how rules (habits, clustered in a genre) cohere to each other as well as the artifact. Furthermore, the Bakhtinian account makes the question of semiosis central in its account of artifacts.

In this section, I draw on the work of M.M. Bakhtin, P.N. Medvedev, and V.N. Voloshinov as I discuss genre theory’s ability to describe how practices cohere and how artifacts acquire a semiotic dimension.

**The Forming of Coherent Collections of Practices**

Genres are conceived as coherent collections of practices that form and develop within spheres of human activity. The notion of a sphere of human activity is somewhat similar to that of an activity network, although not developed to the same degree. Human beings engage in various activities, and Bakhtin implies that those activities spark the development of genres. That is, people develop genres so that they can *do* things — so that they can accomplish their activities. As those activities change, the genres also change (Bakhtin 1981, 1986). It is for that reason that genre is considered a *relatively stable* type of utterance. Gary Saul Morson and Caryl Emerson characterize genre as "the
residue of past behavior, an accretion that shapes, guides, and constrains future behavior" (1990, p. 290). The surface features that might populate a genre, then, are not conveyors of meaning, as formalists would have it; they are not what defines the genre. Bakhtin is resolutely opposed to such a form-content distinction (Bakhtin, 1990; Medvedev, 1978). Rather, they are symptoms of the habits that people in a particular social sphere have developed to deal with certain needs. But those habits also affect how their users view the world.

Genres are doubly oriented: they are oriented towards history and addressivity. In regard to history, genre is a collection of habits or strategies that participants in a given social sphere have developed to deal with particular activities within that sphere. These habits do not spring up ex nihilo for each speaker: each speaker is in some degree a respondent, not the "first speaker" (Bakhtin, 1981, p. 69). Rather, genres are the result of an ongoing dialogue among speakers in a particular sphere of activity, and the past dialogue of those speakers imposes itself upon present speakers in ways that they might not even recognize:

A genre is always the same and yet not the same, always old and new simultaneously. ... A genre lives in the present, but always remembers its past, its beginning. Genre is a representative of creative memory in the process of literary development. Precisely for this reason genre is capable of generating the unity and uninterrupted continuity of this development. (Bakhtin 1984, p. 87)

Genre is thus a sort of social memory that its practitioners accept without their explicit recognition that they are doing so. Such genre habits are extremely powerful because they provide us with ready-made strategies for interpreting not just discourse in a genre, but the world as seen through the "eyes" of that genre (Medvedev, 1978, p. 133-135). Bakhtin clarifies this idea further:

Cultural and literary traditions (including the most ancient) are preserved and continue to live not in the individual subjective memory of a single individual and not in some collective "psyche," but rather in the objective forms that culture itself assumes (including the forms of language and social speech), and in this sense they are inter-subjective and inter-individual (and consequently social); from there they enter literary
works, sometimes almost completely bypassing the subjective individual memory of their creators. (Bakhtin, 1981, p. 249)

The genre embodies a galaxy of assumptions, strategies and ideological orientations that the individual speaker may not recognize. It represents others' "thinking out" of problems whose dialogue has been preserved in the genre's habits (Examples can be found in Bazerman 1988, Berkenkotter and Huckin 1994, Yates 1989).

Genres, although temporarily stabilized social constructs, are also dynamic and reshappable by any speaker for her or his specific utterance. Utterances are unique and unrepeatable as a function of the speech situation in which they are uttered; by the same token, genres are mutable in that they are made by their speakers to address specific speech situations. This addressivity can manifest itself in something so minor as a change in tone or a use of irony, and as significant as the very selection of genre, a blurring among genres, or an introduction of new elements to an existing genre that help it to perform its functions better. One example of the latter is Charles Bazerman's (1988) study of the genre of the experimental article, in which he chronicles how this genre gradually crystallized, in part to substitute for the lack of eyewitnesses (who once were customarily present at experiments).

In the concept of genre we find a strong account of how practices cohere to each other: genre is a collection of practices that finds its nexus not in the artifact (as computational psychology does) but in the recurrent, dynamic activities in which users engage. Furthermore, genre provides a strong account for nonsequential, nonhierarchical collections of practices, an account that activity theory does not provide.

The Role of Semiosis

A genre functions as a way to interpret others' artifacts and construct artifacts meaningful to others. As we have seen, genres are developed in a particular sphere of activity to accomplish certain things within that sphere. Genres accomplish these things by making an artifact meaningful; they invest the artifact with a semiotic dimension.
For instance, let us again consider the Windows online help system. The system has certain features that the user has grown to expect from online help systems: it has a series of buttons across the top of the window; words that are hyperlinks are displayed in green; its introductory screen has a list of major topics that are hyperlinks leading to fuller discussions of those topics. These features — the presence and placement of buttons, the formatting of words, the arrangement of words — *in themselves* do not carry meaning. However, these features have been assembled according to a particular set of habits that meet a certain purpose within a particular sphere of activity: in a word, a genre. If the user recognizes the genre in this particular artifact, she can interpret the artifact in more or less the same way that the designer intended. She immediately knows, for instance, what the buttons are for and how to identify hyperlinks. On the other hand, if the user fails to recognize the genre, she has a much harder time using the artifact.

The significance of these features is not a function of the features themselves. Rather, it emerges from the interaction of the artifact, the individual, and the sphere of activity in which the individual encounters and uses the artifact.

Genre theory could provide a solution to the problem of accounting for the semiotic treatment of artifacts. So far, however, few have attempted to do so. Some have made references to genre, especially when discussing interfaces similar to text, such as online help (Carroll, 1991a, Oren, 1990, Shirk, 1988). But the most ambitious and broad approach to applying genre theory to human-computer interaction has been that of Brown and Duguid (1992). In this article, Brown and Duguid argue that genres are "stable conventions" that "grow up around" artifacts. These genres help their users to make use of "border resources," or characteristics of the artifact-in-use. For instance, an answering machine signals to the caller that it is an answering machine through such border resources as the recorded quality of the voice — a factor that is interpretable through the genre, the cluster of interpretive practices, that has co-developed with the technology of the answering machine. Changes in the technology (e.g. better voice quality) will change the border resources and necessitate changes in our genres.
Brown and Duguid's analysis is thorough, timely, well-reasoned, and full of good examples. Yet their treatment is not activity-centered, it is community-centered. Thus, they find that their definitions of community of practice and border are inherently circular (p. 20, footnote 19). They also conceptualize the artifact-in-itself, separate from its use, its border resources, and the genres that make those resources accessible. That is, they maintain a clear, Cartesian distinction between the mediational means (tool or artifact) and the rules that govern it. Although Brown and Duguid have done important work on applying genre theory to artifacts, much is left to be done, and Bakhtinian genre theory seems to be a good basis for that future work.

Thus far I have discussed genre as a dynamic accretion of habits shared by a set of communicants involved in a shared activity. These habits, we must remember, surround and are inseparable from material, concrete artifacts: texts, paintings, buildings, and even user interfaces. As Medvedev points out, "meanings and values are embodied in material things and actions. They cannot be realized outside of some developed material" (1978, p. 7). Form embraces content (Bakhtin 1990, p. 282). Genre thus collapses the tool-rule distinction by recognizing that these "socio-ideological things" are not things-in-themselves: they are material invested with practical meaning within a specific cultural-historical milieu, engaged in specific activities.

For instance, an online help system is a material artifact: it exists on the user's screen and the user's hard drive. It has a physical manifestation. But a user can make sense of that physical matter only if it is used in a definite activity. In fact, the online help window can be recognized as different from other segments of the screen only because it is used for certain activities, and consequently, genres have evolved for producing, identifying, and using online help windows within those activities.

Bakhtinian genre theory, then, provides a well-developed account of the semiotic dimension of artifacts. Whereas formalists tend to consider the artifact as a code or conduit that unproblematically conveys meaning, genre theory considers the artifact to be
made meaningful by the sphere of activity in which it is encountered and used. An artifact, then, is the union of material and the practices associated with it.

In the next section, I pull together insights from activity theory and genre theory to produce a theory of artifacts.

**Toward a Theory of Artifacts**

The theory of artifacts I have developed provides an underpinning for the methodology that I describe, refine, and evaluate throughout the rest of this dissertation. This theory of artifacts has five main points.

**Activities are Cyclical and Developing**

Artifacts derive their meaning and shape from the activities in which they are used. Since these activities develop and change over time, so do artifacts. Yet artifacts and the practices that infuse them are temporarily stabilized: they develop slowly enough that their use becomes generalizable in the short term. The theory of artifacts needs to provide a cultural-historical account of how activities and their artifact co-develop. Below, I sketch out such an account, based on Yrjö Engeström’s (1990, 1992) discussions of contradictions, discoordinations, and breakdowns. This account is fleshed out and demonstrated in chapters 4-6.

**Contradictions**

Artifacts are used to mediate activities. Yet activities are complex, unstable, changing things. One type of destabilizing factor in an activity network is a contradiction, a tension or imbalance between elements of the network — a tension or imbalance that often occurs because activity networks interpenetrate each other. For instance, the standards for HTML, the page description language for the World Wide Web, was originally developed to provide simple formatting for research scientists wanting to post text. Its formatting capacities are quite limited. But in 1999 websites tend to be developed by graphic designers, who have much more complex formatting needs. The activity networks of research scientists and graphic designers both
interpenetrate the activity network of webpage development. And since these
interpenetrating activity networks have different goals, values, and needs, tensions
appear; debates spring up about how HTML should be extended or replaced; graphic
designers co-opt the language's features to create work-arounds; and research scientists
become unhappy with the general changes in websites. Website designers disagree about
whether websites should include few graphics (so that they can download faster,
delivering information more quickly) or more graphics (so that they can provide
designers with more control over the look of the site, allowing them to make the site
more attractive and familiar to target users).

Contradictions, then, are macro-level differences among activity networks that
interpenetrate the activity network being analyzed. These contradictions tend to manifest
themselves as discoordinations.

Discoordinations
Contradictions, these fundamental tensions and imbalances within an activity
network, can manifest themselves as discoordinations, or difficulties that users face as
they interpret and use artifacts and genres within the network. Whereas contradictions
are analyzable at the macro-level of activity, discoordinations are analyzable at the meso
(middle) level of recurrent actions. Discoordinations involve (1) perceiving and
interpreting artifacts, and (2) managing those artifacts' relationships with other artifacts
used in the activity.

Perceiving and interpreting artifacts. Discoordinations can involve perceiving and
interpreting artifacts. For instance, a graphic designer who creates a website may
encounter discoordinations between his or her tool (an HTML editor with sharply limited
layout abilities) and his or her training (which emphasizes flexible, complex layouts).
The graphic designer's training may lead him or her to interpret a webpage as a canvas on
which one might inscribe free-form designs — but the page refuses to act like a canvas.

Managing artifacts' relationships. Discoordinations can also involve managing
artifacts' relationships. Graphic designers often find themselves enlisting other artifacts
to use in conjunction with HTML, especially artifacts that provide ways to get around HTML's limitations. For instance, most graphic designers use illustration programs and photograph editors to create graphics which they can then include in their webpages. Yet managing these artifacts with HTML is sometimes a challenge. Unlike other computer display formats, HTML does not guarantee that graphics will be displayed at a specific level of detail (it depends on the computer that views the webpage). Similarly, although HTML allows the designer to pick a certain color for text, that color also varies from display to display, making color-matching between graphics and text almost impossible. Finally, the strictures of the World Wide Web call for graphics to be as small as possible for quick download time — and creating small graphics can be a challenge when one uses a graphics program that was built to produce detailed graphics.

Because of the underlying contradiction between graphic designers' activities and research scientists' activities, then, discoordinations tend to crop up throughout the graphic designers' work, in their recurrent actions. These meso-level discoordinations lead to breakdowns.

**Breakdowns**

Breakdowns are points at which a user finds her or his present interpretation of an artifact to be inadequate for the task at hand. These are symptoms of deeper discoordinations and the contradictions that underlie them. Like most symptoms, these are often mistaken for the disease; they tend to be the focus of much usability research. Breakdowns are analyzable at the micro-level, that is, by examining moment-by-moment actions and operations.

I use the term *breakdown* as it is used by Bødker (1991), who draws it from Winograd and Flores (1986) and Engeström (1990). According to Bødker, a breakdown happens when an artifact that has become operationalized ("ready-to-hand") suddenly does not behave as expected. At this point, the user becomes conscious of the artifact and her unconscious operations on the artifact consequently become conscious actions. The artifact, which was used to mediate the user's actions on an object, now itself becomes
"unready-to-hand" — it is now the object of the user's actions. The user must reinterpret the artifact if she is to continue to use it to mediate her activities.

For instance, a graphic designer has learned certain actions to perform when designing publications using page design software, such as Adobe PageMaker or Quark XPress. These actions include using a text tool to set columns of text. And the actions involved in using the tool — clicking the tool icon with the mouse, clicking the cursor at one point on the screen, and dragging it to another point to delineate a text column — quickly become automatic. The designer no longer has to consciously think about performing these actions; they have been operationalized.

Yet when the designer turns to webpage design, he finds that these automatic operations no longer help him. Unlike page design software, webpages do not feature columns of text. And this difference, this discoordination between the page design tool and the capabilities of HTML, can lead to breakdowns. For instance, when using a webpage design tool such as Adobe PageMill (which looks quite similar to PageMaker), the graphic designer may automatically try to create a text column. When he realizes that the text tool is not available, he becomes conscious of his (heretofore unconscious) actions. He reinterprets the PageMill interface as a web design tool rather than a page design tool.

Breakdowns are not necessarily undesirable for users. They can be opportunities for learning and a basis for reconceptualizing (Bødker, 1991). Breakdowns provide users with clues for applying different rules, and sometimes different clusters of rules (genres), to the artifact. And they can also be an interesting way of understanding where a user's expectations mismatch the mediating artifact she is attempting to use — that is, when the user has applied an inappropriate genre to the artifact. A micro-level analysis of users' moment-by-moment actions can turn up breakdowns and provide important starting points for examining discoordinations and contradictions.

Indeed, this is where the rubber meets the road. Breakdowns are symptoms of these underlying discoordinations and contradictions. The methodology that I develop in this
dissertation involves studying these symptoms to understand, diagnose, and treat the discoordinations and contradictions they indicate.

In chapters 4 and 5 I discuss breakdowns in more detail, providing micro-level analyses that I relate to meso-level analyses of discoordinations and macro-level analyses of contradictions.

*Artifacts are Interpreted Segments of the Material Environment*

Artifacts are the confluence of the material environment and the cultural-historical rules developed to deal with that environment within certain activities. That is, artifacts are not separable from the cultural-historical milieus in which they are encountered. Rather, the very act of recognizing an artifact necessitates that the user be involved with an activity; know about certain artifacts involved in that activity (as either mediational means or objects); identify a segment of the material environment that can be isolated and used as an artifact; and apply the rules of the expected sort of artifact to the new artifact. Recognizing an artifact entails recognizing that it belongs in one's lifeworld.

For instance, I have already pointed out that an online help window appears to the user as merely a specific segment of the screen. That segment is recognizable as different from the rest of the screen only because the user has encountered similar artifacts (similar application windows, perhaps) in the past, and uses those experiences to isolate the help window from the rest of the screen. She then invests the window with the rules that she has applied to similar windows: perhaps she clicks on the buttons along the top or moves the scrollbar.

Artifacts, then, are interpreted according to past experiences with the genres of which the artifacts are perceived to be instantiations. When the user interprets a segment of the environment as an artifact, she applies a relatively stable cluster of rules (a genre) to the segment; an artifact is created through the investing of the environment with rules. And as the above example shows, some interpretations — and some genres — lead to fewer breakdowns than others.
Artifacts’ Rules are Complex, Relatively Stable, Co-Developing, and Socially Distributed

As I have argued, the coherent sets of rules that cluster around artifacts are complex, co-developing, and socially distributed. In the past they have been analyzed differently because their nexus has been taken to be the artifact itself rather than the activities in which the artifact are used. These clusters originate in activities and develop as a coherent unit to address similar activities. Collections of rules thus provide a sort of social memory, a way for subjects within activity networks to share and stabilize the strategies they have developed for dealing with artifacts.

Genre provides a unit of analysis for the coherent collections of rules surrounding an artifact. Since rules are related to other rules in relatively stable collections, breakdowns must be considered in terms of the other rules, that is, in terms of the genre(s) surrounding the artifact.

Artifacts are Instantiations of Cultural-Historical Traditions

Just as a word is always "half someone else's" (Bakhtin, 1981), so is a genre. A genre provides a relatively stable, historically developed set of habits that is easy for authors/developers to produce and readers/users to interpret. To take a simple example, a webpage designer tends to automatically structure pages so that first-level headings represent main points, second-level headings represent subpoints, and so forth. Readers familiar with webpages tend to interpret the headings in the same way. The relationships between levels of headings exist not in the code itself but in the common genre that designer and user share.

Since any given artifact is the confluence of the material environment and the rules that a user applies to the environment, genres (which are relatively stable, historically developed, socially distributed collections of rules) are especially important for designing and using artifacts. Genres provide traditions of a sort: common cultural-historical resources on which designers and users can draw.
Genres can be studied at the levels of the institution and the individual. A focus on institutional history and addressivity means that we can talk about and track artifacts' development in institutional terms. And a focus on individual history and addressivity means that we can talk about and track artifacts' development in individual terms.

Artifacts Occupy Ecologies

We sometimes tend to think of activities as mediated by a single artifact. For instance, we may be tempted that a web designer's work is mediated solely by her webpage editor. But as we investigate the designer's activity, we may find that she uses several artifacts to jointly mediate the activity: besides the editor itself, she may use one or more web browsers to examine her completed webpages; notes and sketches to help her as she plans and executes the work; a host of graphics programs to create graphics for the website; printouts of the webpages that she uses to mark changes and make annotations; a Pantone color wheel for helping her to pick colors; software documentation; web design texts; others' webpages, which provide tips and tricks or serve as models. We can imagine dozens more tools that the designer might use, especially if we explore how she collaborates with others in her activity network: programmers, database administrators, marketing personnel, executives, and so forth.

We can think of these tools as forming an ecology of tools (see Hutchins, 1995): a relatively stable set of tools that are used in concert to mediate complex activities. I explore the notion of ecology more thoroughly in Chapter 3.

Conclusion: Where the Theory Leads

The theory of artifacts I have articulated in this chapter provides a basis for a cultural-historical research methodology, one that can guide research into the design and evaluation of information systems. It has five main tenets:

(1) Activities are cyclical and developing

(2) Artifacts are the meeting of environment and rules
(3) Artifacts' rules are complex, relatively stable, co-developing, and socially distributed

(4) Artifacts are instantiations of cultural-historical traditions

(5) Artifacts occupy ecologies

I continue to develop the methodology throughout the rest of the dissertation. In the next chapter, I describe methods that I used to conduct five empirical studies, studies through which I test, extend, and refine the methodology.
CHAPTER 2
METHODS

In the remainder of the dissertation, the theory of artifacts described in Chapter 1 becomes the basis for the research methodology I am attempting to develop. In its broad outlines, the methodology consists of the five tenets of the theory of artifacts applied to existing research methods. By applying these tenets to methods used in empirical studies, I develop parts of the methodology in more detail, and I test, extend, and refine the parts of the methodology.

Given the cultural-historical theory of artifacts that is my starting point, I use methods that are designed to explore the mediatory roles of artifacts and their genres in various activity networks. Below, I describe the methods that I use to gather and analyze data from a pilot study I conducted at Schlumberger Well Services and the five interrelated studies I conducted in conjunction with the Iowa Department of Transportation and the Center for Transportation Research and Education. These five studies explore four questions:

• Where is "usability" located? That is, working within a cultural-historical framework, how can we locate and conceptualize the difficulties that people encounter as they attempt to use information systems?
• How do participants in an activity come to perceive and manage the various on- and off-screen tools used in an activity to mediate their work?
• How do participants come to encounter difficulties in perceiving and managing these on- and off-screen tools?
• If information system designers import some of the many off-screen tools into the interface — transforming them into on-screen tools — how do these changes affect how users perceive and manage the tools?

Below, I discuss the methods I used to conduct the four studies.
Study 1: The Pilot Study (Chapter 3)

The pilot study attempts to answer the question: Where is "usability" located? That is, working within a cultural-historical framework, how can we locate and conceptualize the difficulties that people encounter as they attempt to use information systems?

The study, conducted over 10 weeks during the summer of 1997, involved observations and interviews of 20 software developers and interviews of two additional developers. The developers worked at three business groups of Schlumberger, a multinational corporation that provides information tools for the oil industry, including software dedicated to analyzing and interpreting seismic and drilling data. Schlumberger encourages developers to reuse existing code from the corporation's code libraries — collections of commonly useful code features that are accessible to all developers working on a given project. Yet developers sometimes have a difficult time finding appropriate code in the libraries; consequently, they "reinvent the wheel," writing new code that is not always as efficient or well-tested as the existing code in the libraries.

Originally, I conducted this pilot study with the following research questions:

• How are developers at Schlumberger Well Services currently finding appropriate library code to use? That is, how do developers find out about the code features (datatypes and routines) that currently exist in Schlumberger's libraries?

• How do the developers use mediational means (such as search tools, features of the code, communication media, and manuals) when comprehending and producing code?

The answers to these questions helped Schlumberger's User Services for Engineering Research (USER) team to map out the ecology of tools (Hutchins, 1995) in use at the research sites. The result was insight into how the tools related and how information systems could be designed to fit into this ecology. But the study, analyzed properly, can help to answer this dissertation's first research question: Where is "usability" located? Below, I discuss the steps I took to investigate the ecology; the research sites I visited; and the participants I observed and interviewed.
Investigating the Ecology

I visited 22 developers at work. For 20 of them, I used the following methods:

1. **Opening interview.** I asked each developer a series of questions meant to explore their background with software development, Schlumberger, and the code with which they were working. These questions are reproduced in Appendix A. Interviews were audiotaped. These questions were designed to draw out information about the developer’s background and development, as well as the object of his or her work, the community that participated in the activity, and the division of labor within that community.

2. **Observation.** After completing the opening interview, I silently observed the developer as he or she worked, taking field notes. I particularly focused on mediational means that developers used as they attempted to find and use existing code. Observations lasted 30 minutes to an hour, but averaged about 45 minutes.

   The observations allowed me to compile a list of some of the mediational means used in the subject’s labor and the rules that governed them. The observations also gave me insight into the object of the labor (the code), the community, and the division of labor.

3. **Closing interview.** At the end of the observation period, I conducted a stimulated recall interview in which I asked the developer about (a) the mediational means that I had seen him or her using and (b) mediational means that he or she did not use in that session. Interviews were audiotaped.

   The closing interview allowed me to round out the list of mediational means, as well as discuss the other parts of the activity network in which each subject labored.

   I was not able to observe the other two developers at work, but did conduct the opening interview and also asked them about mediational means they had used when finding information.

   After conducting the interviews and observations, I did the following:
• Categorized each developer's work as developing, maintaining/debugging, or documenting.
• Listed the mediational means I had observed them using, and categorized the means as genres used to address and mediate the developers' labor.
• Listed questions that developers used each mediational means to answer, based on the interview responses.

Beyond helping me to think through the nature of usability, the information gathered during this pilot study helped me to develop the notion of an ecology of genres. It also allowed me to investigate mediational means in terms of genre. Finally, it underscored the need for a systematic exploration of breakdowns. These are all dealt with in the following four interrelated studies.

**Background of the Four Interrelated Studies: The ALAS Lifeworld**

In the four interrelated studies described in the rest of the dissertation, I examine how users interact with a family of products developed by the state of Iowa to locate and analyze vehicle crashes on Iowa roads. The current system, a DOS-based product entitled the PC-based Accident Location and Analysis System (PC-ALAS), is an aging program that is distributed to various local and state agencies. If all goes to plan, it will be supplanted by a Microsoft Access-based system (Access-ALAS) and eventually by two geographic information systems (Explorer-ALAS and GIS-ALAS). At the time of this project (1998), PC-ALAS is used by various users such as public works staff, traffic engineers, engineering assistants, state patrol personnel, city police, county sheriffs, and personnel from law enforcement agencies. Although the ALAS family of products is examined in chapters 4, 5, and 6, I give a brief overview of PC-ALAS and GIS-ALAS here.

PC-ALAS is used to obtain accident statistics for locations during certain time periods by searching "nodes" of information. These nodes can be searched by accident characteristics and/or driver characteristics. PC-ALAS can display the results on screen, send them to a file, or print them using predefined formats.
However, users have informally asserted that PC-ALAS is difficult to use. One problem that the designers have identified is that users need to use PC-ALAS in conjunction with cumbersome node tables or paper maps; nodes are not shown graphically. Another problem is that the existence of a node does not necessarily indicate the existence of accident data — a distinction that tends to confuse users.

In response, CTRE is currently developing GIS-ALAS, a complex set of macros that transforms an existing product (ArcView) into an accident location and analysis system that has PC-ALAS' capabilities and general menu structure. GIS-ALAS displays maps on-screen, eliminating the need for paper or CAD maps. It also allows users to specify accident locations spatially or through queries, eliminating the need for specifying nodes or using node tables. Finally, query results can be displayed on a map or in a table, creating varied displays of inputs (queries) and outputs (statistics).

I conducted four studies of information systems in the ALAS lifeworld.

**Study 2: The Historical Study of the ALAS Activity Network (Chapter 4)**

In Study 2, I conducted a historical analysis of accident location and analysis in Iowa. This analysis answered the research question: How do participants in an activity come to perceive and manage the various on- and off-screen tools used in an activity to mediate their work? I obtained data through conducting interviews and examining historical documents.

1. **Interviews** were conducted with five personnel who have been involved in Iowa's accident location and in the development of ALAS tools.
2. **Historical documents** included ALAS manuals, articles from specialty journals, proposals, and reports.

After gathering the interviews and historical documents, I analyzed them in terms of the theory of artifacts outlined in Chapter 1. The combined activity theory-genre framework allowed me to conceptualize and analyze changes within the activity network over three decades. I analyzed the data using a genre analysis and an activity theory analysis.
Genre Analysis

The genre analysis involved identifying artifacts, understanding them in terms of genre, and tracing these genres across the four decades covered by the study. I identified genres by examining, in Bakhtinian terms, the historical, addressive, and formal relationships among artifacts, both within and without the computer interface.

For instance, as I discuss in Chapter 4, the genre of the map is central in all stages of development of the ALAS activity network.

**Historical relationships.** All of the maps that show up in the ALAS activity network — standard city and county maps, node maps, GIS-based map windows — are historically related: node maps are a transformation of the older city/county maps, while GIS-ALAS’ online maps are a transformation of the node map. They are all part of the same genre, the map. But they are also members of subgenres, i.e. the Story County node map for 1996 is a member of the genre of node maps, and can itself be studied as a historical moment in the development of the node map genre. I investigated historical relationships by examining documents and interviews for clues about how various artifacts originated.

**Addressive relationships.** Genres, in the Bakhtinian conception, are more than historically related. They are also related in terms of addressivity. What activities does a given genre mediate in a given activity network? In the ALAS activity network, for instance, maps primarily are used to represent and visualize accident locations. And this use holds true across all of the map genres. I investigated addressivity by examining similarities in use among artifacts in different points of the ALAS history.

**Formal relationships.** Although Bakhtinian genres are not conceived as formalist, formal characteristics can be understood as possible indicators of genre relationships. For instance, all map genres used in the ALAS activity network share formal characteristics such as graphic depictions of primary, secondary, and municipal roads. After finding historical and addressive relationships among artifacts, I looked for formal relationships as confirmatory evidence of artifacts’ generic commonalities.
Activity Theory Analysis

The activity theory analysis involved using the data to sketch detailed activity networks at pivotal stages in ALAS development. I categorized data and identified contradictions.

Categorizing data. I categorized the interview and documentary data according to the activity network categories discussed in Chapter 1: subject, mediational means, object, outcome, rules, community, and division of labor. I used these categories as a heuristic for organizing data and conceptualizing relationships among the categories during certain periods in the ALAS activity network: just before and just after the introduction of various ALAS tools. I then depicted these stages of the activity network in triangle diagrams similar to those seen in Chapter 1.

Identifying contradictions. Once I had identified and pictorially depicted the ALAS activity network, I looked for contradictions (Engeström, 1990, 1992) in the activity network that were eased by the introduction of new ALAS tools. I identified contradictions from the interviews and documentary data, where they were often conceptualized as "problems" or indicated by the description of new features in subsequently developed ALAS tools.

The information I gathered during this stage helped me to understand the experiences that participants (and the activity network in general) had had with previously learned genres. It provided me with vital background information as I proceeded to Study 3.

Study 3: Discoordinations and Breakdowns that Participants Encounter when Using PC-ALAS (Chapter 5)

In the study of PC-ALAS use, I asked the question: How do participants come to encounter difficulties in perceiving and managing on- and off-screen tools? To answer the question, I continued to explore the experiences that PC-ALAS users have with genres. In addition, I observed and interviewed participants to determine what
breakdowns they experience when using PC-ALAS, and what roles discoordinations played in these breakdowns.

I collected all data from 26 participants at various state and local agencies such as the Iowa Department of Transportation, the Federal Highway Authority, the Governor's Traffic Safety Board, and city and county engineers' departments, sheriffs' offices, and police departments.

Ten participants completed questionnaires, which then became the basis for interviews exploring their experiences with the ALAS family.

1. **Questionnaires** (Appendix B) collected data about the participants' experience with their activities, the ALAS family of software, and other types of online products with which they were familiar.

2. **Stimulated recall interviews** were usually conducted over the telephone. They involved clarifying and amplifying the participants' answers to the questionnaires. Interviews were audiotaped.

Five participants demonstrated activities ancillary to PC-ALAS use, including the entry of crash reports into the PC-ALAS database, the entry of crash information into the MARS database, the use of node maps, and the use of PC-ALAS data to write an environmental impact report for a government agency.

1. **Questionnaires** collected data about the participants' experience with their activities, the ALAS family (if applicable), and other types of online products with which they were familiar.

2. **Observations** were recorded in field notes as well as on videotape.

3. **Stimulated recall interviews** were conducted after the observations and involved clarifying the participants' answers to the questionnaires. Interviews were audiotaped.

---

7 One participant took part in the round of questionnaires and interviews, then later took part in the round of questionnaires, observations, and interviews. All other participants took part in only one round of research.
Twelve participants completed questionnaires and were observed as they either completed actual work with PC-ALAS or walked through data similar to that which they would normally process with PC-ALAS. (One of these participants also took part in a telephone interview.) After the observation, I interviewed participants about their general experiences with PC-ALAS and their specific experiences during this observation.

1. **Questionnaires** collected data about the participants' experience with their activities, the ALAS family, and other types of online products with which they were familiar. I distributed and collected the questionnaires before each participant's session.

2. **Observations** were recorded in field notes as well as on videotape, and later coded. In six cases, observations were of participants' actual work; however, as I discuss in Chapter 5, most participants used PC-ALAS rather infrequently — one to twelve times a year — and therefore naturalistic observations were difficult to schedule. In the remaining six cases, participants walked through tasks similar to those they would normally conduct with PC-ALAS, using actual data and often actual problems that they had encountered recently.

3. **Stimulated recall interviews** were conducted immediately after the observations and involved reviewing the participants' work during the observation as well as in general. I gave special attention to breakdowns encountered during the session. Interviews were audiotaped.

Once I had collected the data, I analyzed them on three levels. Through a macrolevel analysis, **participant profiling**, I drew on all three types of data to construct a detailed picture of the participants and the local activity networks in which the PC-ALAS participants worked. Through a microlevel analysis, **video coding**, I categorized the artifacts, actions, operations, and breakdowns of the 12 observed PC-ALAS participants. And I constructed mesoanalyses, **chronological accounts**, through a qualitative examination of the 12 participants' sessions in terms of their local activity networks and the genres they used. I describe each of the three analyses in more detail below.
Participant Profiling

Although the historical analysis in Chapter 4 gave a general description of the ALAS activity network, local agencies such as sheriff’s offices, county engineers’ offices, and traffic safety organizations have their own local activity networks as well. They participate in the activity of locating and analyzing accidents, but in terms of their own desired outcomes (law enforcement, engineering, traffic safety). I attempt to learn more about some of these local organizations through participant profiling.

Participant profiling, in this study, involved the following steps:

1. **Categorize local objects and outcomes.** First, I examined questionnaire and interview data in terms of the participants’ activities. In particular, I studied their organizations’ desired outcomes and how the participants transformed the object of their labor (PC-ALAS output) to produce those outcomes, which indicate interpenetrating activity networks.

2. **Summarize demographics.** Next, I examined questionnaire and interview data in terms of participants’ demographics, including age, sex, years using PC-ALAS, frequency of PC-ALAS use, and years using various operating systems. I categorized these by the categorized outcomes.

3. **Transcribe usage comments.** I reviewed interview data and field notes, transcribing portions relating directly to the following issues: a) what tasks the participants were trying to accomplish; b) how participants coordinated PC-ALAS with other artifacts; c) how participants used PC-ALAS reports in their work; and d) what complaints the participants had about PC-ALAS. These transcribed portions served to more fully profile participants.

I stored these data in a database, which I could easily arrange, sort, and explore.

**Video Coding**

Although macroanalyses are important for a general understanding of artifact use, a specific understanding requires more detailed observation. To provide this level of detail,
I coded the 12 videotaped PC-ALAS observations in terms of interface artifacts used by the participant and their genres (including menus, dialog boxes, and the like); actions performed on the artifacts; operations used to perform the actions; and breakdowns encountered by the participant.

This video coding scheme is based on Susanne Bødker's (1996). But whereas Bødker coded events by breakdowns, I code events by actions. Consequently, I can compare successful and unsuccessful actions and associate them with the genres and artifacts within which they occur.

Table 2.1 describes the coding scheme and shows a sample episode that has been coded. Episodes are events in which a participant performs an action on an artifact. In this sample episode, the action is the entering of a county number into a field of an artifact, the City/County Request Dialog Box (an instance of the genre of the data entry dialog box). The action of entering the number could have been performed with a variety of different operations; here, the participant clicks on the field and types. But the participant encounters a breakdown: the number is not entered into the field. This particular breakdown occurs frequently in the data and is often due to the field being improperly selected.

Table 2.1 Video coding categories. Detailed coding was used for precise accounting of artifacts and actions. Simplified coding was used for measuring inter-rater reliability.

<table>
<thead>
<tr>
<th>Coding Category</th>
<th>Artifact Description</th>
<th>Action</th>
<th>Operation(s)</th>
<th>Breakdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Detailed</td>
<td>City/County Request Dialog Box</td>
<td>Enter county number</td>
<td>Click on field, type</td>
<td>Input not entered</td>
</tr>
<tr>
<td>Example: Simplified</td>
<td>Data entry dialog box</td>
<td>Enter number</td>
<td>Click on field, type</td>
<td>Input not entered</td>
</tr>
</tbody>
</table>

Table 2.1 Video coding categories. Detailed coding was used for precise accounting of artifacts and actions. Simplified coding was used for measuring inter-rater reliability.
Figure 2.1 shows a series of episodes executed by one of the participants. As Figure 2.1 demonstrates, each artifact could have multiple actions performed on it. For instance, the participant performed seven separate actions on the Mile Pointed Request dialog box—entering data in various fields and accepting the settings—and these actions are represented in seven different entries.

Each action can be carried out using one or more operations. For instance, in Figure 2.1, some numbers are entered by (1) clicking in a field and (2) typing. Other numbers are entered by (1) tabbing into a field and (2) typing.

Finally, an episode might involve a breakdown. In Figure 2.1, two breakdowns occur. In one, the participant attempts to enter a number into a number field, but the field has not been properly selected. In the other, the system displays a red Warning message box which reports an error in the participant’s input.

I coded both simplified and detailed information. The detailed information included names of specific artifacts and precise descriptions of actions performed on those artifacts; coding this detailed information required an intimate knowledge of PC-ALAS. The simplified information included artifact genres (types of artifacts) and generic actions; coding this simplified information required only a general knowledge of PC-ALAS, so it was suitable for checking inter-rater reliability.

I tested inter-rater reliability using the instructions in Appendix C. The reliability test determined whether the coding scheme was systematized enough to be teachable to other raters. One rater was trained during an hour-long session by (a) viewing and coding two five-minute segments of an observation; (b) viewing and coding another five-minute segment; and then (c) comparing the result with my coding. The subsequent training sessions helped me to refine my coding scheme.

After the training, the rater coded the first eight minutes of each work session, or if the session was shorter than eight minutes, the entire session (Table 2.3). (Work sessions were used because they provided a closer representation of workplace use than walkthroughs and because they generally involved more interaction with PC-ALAS.)
Figure 2.1 A portion of detailed video coded data.

Table 2.2 Cohen’s kappa for the four categories.

<table>
<thead>
<tr>
<th>Artifact</th>
<th>Action</th>
<th>Operation</th>
<th>Breakdown</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.935</td>
<td>0.927</td>
<td>0.919</td>
</tr>
<tr>
<td></td>
<td>(93.5%)</td>
<td>(92.7%)</td>
<td>(91.9%)</td>
</tr>
</tbody>
</table>

The rater coded 21.1% of the total video data. Table 2.2 indicates that the rater reliably applied the coding scheme.

Once I had coded the video data, I entered it into a database. I then generated various reports and graphs to help me visualize relationships among the coded elements and with the participant profiles.

Case Studies

After conducting the macro- and micro-level analyses, I used them to construct case studies (mesoanalyses) of all users’ sessions.
Table 2.3  Observed PC-ALAS participants. Italicized participants were coded by raters for the indicated duration. "Total duration" shows the time of each user's entire PC-ALAS interaction. Participants are arranged by interpenetrating activity networks.

<table>
<thead>
<tr>
<th>User</th>
<th>Total duration</th>
<th>Duration of rater coding</th>
<th>Type of session</th>
<th>Interpenetrating AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>P13</td>
<td>11:48</td>
<td>--</td>
<td>Walkthrough</td>
<td>Traffic safety</td>
</tr>
<tr>
<td>P16</td>
<td>57:38</td>
<td>8:00</td>
<td>Work</td>
<td>Traffic safety</td>
</tr>
<tr>
<td>P17</td>
<td>25:40</td>
<td>--</td>
<td>Walkthrough</td>
<td>Traffic safety</td>
</tr>
<tr>
<td>P19</td>
<td>28:08</td>
<td>8:00</td>
<td>Work</td>
<td>Traffic safety</td>
</tr>
<tr>
<td>P4</td>
<td>11:46</td>
<td>8:00</td>
<td>Work</td>
<td>City engineering</td>
</tr>
<tr>
<td>P18</td>
<td>1:11</td>
<td>--</td>
<td>Walkthrough</td>
<td>County engineering</td>
</tr>
<tr>
<td>P22</td>
<td>2:24</td>
<td>2:24</td>
<td>Work</td>
<td>County engineering</td>
</tr>
<tr>
<td>(with P21) (entire session)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P23</td>
<td>12:05</td>
<td>--</td>
<td>Walkthrough</td>
<td>County engineering</td>
</tr>
<tr>
<td>P25</td>
<td>3:57</td>
<td>3:57</td>
<td>Work</td>
<td>City engineering</td>
</tr>
<tr>
<td>(entire session)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P27</td>
<td>12:23</td>
<td>8:00</td>
<td>Work</td>
<td>County engineering</td>
</tr>
<tr>
<td>(with P26)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P24</td>
<td>1:54</td>
<td>--</td>
<td>Walkthrough</td>
<td>Law enforcement</td>
</tr>
<tr>
<td>P32</td>
<td>12:47</td>
<td>--</td>
<td>Walkthrough</td>
<td>Law enforcement</td>
</tr>
<tr>
<td>Total across all users</td>
<td>3:01:41</td>
<td>38:21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. **Action sequence graphs.** Using the videocoding data, I generated graphs showing participants' action sequences (requesting data, searching data, generating reports) in the order that they occurred within the sessions, along with breakdowns that occurred within those action sequences. These data allowed me to visualize participants' typical action sequences and clusters of breakdowns, meaning that I could associate breakdowns with parts of the sequences as well as with individual artifacts or genres. For instance, P24 encountered multiple breakdowns with vertical menus while reporting, but not during other action sequences. This unusual clustering led me to examine the video data and the stimulated recall interview (see below); both suggested that these breakdowns
occurred because P24 could not remember how to print, not because the menus were difficult to use.

2. **Examination of participants' breakdowns.** Once I had spotted clusters of breakdowns in the graphs, I investigated the clusters by examining the video data where the breakdowns occurred, as well as stimulated recall interviews discussing these or similar breakdowns. These data provided evidence that I used to determine which breakdowns could be attributed to genre expectations. For instance, P22 and P24 both examined PC-ALAS' File menu before moving on to other menus. Both indicated in stimulated recall interviews that the desired act — printing — was usually handled in the File menu of Windows-based programs; they expected the File menu to handle such functions likewise, even though they had used PC-ALAS in the past and knew that the functions were handled elsewhere.

3. **Comparison of action sequences and artifact use across participants.** Finally, I categorized participants by data in their profiles: object, outcome, and demographic data (such as frequency of PC-ALAS use and number of years of Microsoft Windows experience). Then I compared participants' action sequences and genre-related breakdowns across categories. Doing so allowed me to examine users' recurrent actions in terms of the genres, objects, and outcomes that were involved; this mesoanalysis gave me added insight into the discoordinations that users faced as they perceived and managed genres.

Once I had conducted these analyses, I had the background to evaluate how participants used two GIS-ALAS prototypes.

**Study 4: Discoordinations and Breakdowns that Four PC-ALAS Participants Encounter when Using an Undeveloped GIS-ALAS Prototype (Chapter 6)**

In this and the following study, I investigated the last research question: If information system designers import some of the many off-screen tools into the interface — transforming them into on-screen tools — how do these changes affect how users
perceive and manage the tools? To investigate this question, I studied how four participants from the ALAS activity network conducted actual work using an early prototype of a geographic information system (Prototype A).

The four workers completed questionnaires and were observed in their normal work using Prototype A. After the observation, workers were interviewed about their use of the prototype.

1. **Questionnaires** (Appendix B) were used to explore the participants' experience with their activities, the ALAS family, and other types of online products with which they were familiar. The questionnaires allowed me to compare the workers in this study with those in Study 3, allowing me to confirm that both sets of workers had generally similar backgrounds and experiences and engaged in generally similar activities. I distributed and collected the questionnaires before each participant's session.

2. **Observations** were recorded in field notes as well as on videotape, and later coded. Participants used Prototype A to conduct work that they would normally conduct with PC-ALAS. These coded observations allowed me to examine the data systematically. In particular, they helped me to find patterns of tool use; identify discoordinations; and observe which genres users adapted into the genre ecology.

3. **Stimulated recall interviews** were conducted immediately after the observations and involved reviewing the participants' work during the observation as well as in general. Special attention was given to breakdowns encountered during the session. Participants were also asked to compare the interface with those of PC-ALAS and related information systems. Interviews were audiotaped. I transcribed selected portions to "profile" participant activities and the use of Prototype A during those activities.

I analyzed these data through participant profiling (a macro-level analysis) and case studies (a meso-level analysis).
**Participant Profiling**

I profiled participants in much the same way that I did the PC-ALAS participants in Study 3. Participant profiling, in this study, involved the following steps:

1. **Characterize local objects and outcomes.** First, I examined questionnaire and interview data in terms of the participants’ activities. In particular, I studied their organizations’ desired outcomes and how the participants transformed the object of their labor (Prototype A output) to produce those outcomes.

2. **Summarize demographics.** Next, I examined questionnaire and interview data in terms of participants’ demographics, including age, sex, years using PC-ALAS, frequency of PC-ALAS use, and years using various operating systems. These data helped me to understand the broad activities in which users were engaged.

3. **Transcribe usage comments.** I reviewed interview data and field notes, transcribing portions relating directly to the following issues: (a) what tasks the participants were trying to accomplish; (b) how participants coordinated Prototype A with other artifacts; (c) how participants used Prototype A in their work; and (d) what complaints the participants had about Prototype A. These transcribed portions served to more fully profile participants.

I stored these data in a database, where I could easily arrange, sort, and explore them.

**Case Studies**

After constructing the participant profiles, I used them to guide my exploration of the data.

1. **Categorize other genres.** First, I examined the videotape to identify other genres that participants adapted as they used Prototype A. This analysis allowed me to map out the genre ecology and hypothesize how it might develop over time.

2. **Examine points of voiced confusion.** Next, I identified and transcribed points at which participants voiced confusion about Prototype A. I examined participants'
comments and behavior in terms of the genres they had trouble perceiving and coordinating, as well as the specific breakdowns they encountered. This analysis allowed me to examine breakdowns as symptoms of deeper discoordinations and contradictions.

3. **Examined genres that they had described as problematic in the stimulated recall interviews.** Finally, I examined the videotape for points at which participants used particular genres that they had described as problematic in the stimulated recall interviews. This analysis provided another view of breakdowns and discoordinations, since it allowed me to examine users' successful as well as unsuccessful encounters with difficult genres.

Once I had conducted these analyses, I was able to assess how experienced PC-ALAS participants might react to Prototype A.

**Study 5: Discoordinations and Breakdowns that Students Encounter when Using a Developed GIS-ALAS Prototype (Chapter 6)**

As in Study 4, here I investigated the last research question: If information system designers import some of the many off-screen tools into the interface — transforming them into on-screen tools — how do these changes affect how users perceive and manage the tools? To investigate this question, I studied a group of 13 students using a second, more customized prototype of GIS-ALAS (Prototype B). The students were seniors and graduate students in a Community and Regional Planning course on GISes, taught at a large midwestern state university (Table 2.4).

This investigation allowed me to gain a broader perspective on GIS-ALAS design because it allowed me to compare participants with different types of expertise using similar artifacts. Recall that participants in Study 3 were familiar with the activity of locating and analyzing accidents, but were unfamiliar with GISes. Participants in Study 4 were familiar with both. In Study 5, the 13 students were familiar with GISes, but were unfamiliar with accident location and analysis.
Table 2.4  Student participants.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Sex</th>
<th>Age</th>
<th>Degree</th>
<th>Major</th>
<th>Experience (in years)</th>
<th>ArcView</th>
<th>MapInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>F</td>
<td>22</td>
<td>Bachelor's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>G2</td>
<td>M</td>
<td>24</td>
<td>Bachelor's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>G3</td>
<td>M</td>
<td>27</td>
<td>Master's</td>
<td>Public Administration</td>
<td></td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>G4</td>
<td>M</td>
<td>22</td>
<td>Bachelor's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>G5</td>
<td>M</td>
<td>21</td>
<td>Bachelor's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>G6</td>
<td>F</td>
<td>23</td>
<td>Master's</td>
<td>Water Resources</td>
<td></td>
<td></td>
<td>0.25</td>
</tr>
<tr>
<td>G7</td>
<td>F</td>
<td>20</td>
<td>Bachelor's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>G8</td>
<td>F</td>
<td>39</td>
<td>Master's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>G9</td>
<td>F</td>
<td>28</td>
<td>Master's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>G10</td>
<td>M</td>
<td>38</td>
<td>Master's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>G11</td>
<td>M</td>
<td></td>
<td>Master's</td>
<td>Landscape Architecture</td>
<td></td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>G12</td>
<td>M</td>
<td>21</td>
<td>Bachelor's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>G13</td>
<td>M</td>
<td>24</td>
<td>Bachelor's</td>
<td>Community and Regional Planning</td>
<td></td>
<td>0.25</td>
<td>0.33</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25.75</td>
<td>0.48</td>
</tr>
</tbody>
</table>

The 13 students completed questionnaires. I then gave them a set of instructions (Appendix I) and observed them as they used Prototype B. Afterwards, I interviewed them about their use of Prototype B and the breakdowns they had encountered during the session.

1. **Questionnaires** (Appendix F) were distributed and collected before the observation. These provided the same information as the questionnaires from the previous stages, and therefore allowed me to compare the groups. Using this information, I was better able to account for similarities and differences among the groups of participants.

2. **Observations** involved having the participants respond to a set of instructions that resembled their class assignments. The instructions required them to find and analyze certain accidents. I videotaped observations and recorded them in field notes; I later coded the observations. I gave special attention to breakdowns. These coded observations allowed me to examine the data systematically. In particular, they helped me to find patterns of tool use; identify dis coordinations; and observe how users dealt with the various genres in the ecology.
3. **Stimulated recall interviews**, conducted immediately after the observations, explored breakdowns that participants had encountered as they used Prototype B. Participants were also asked to compare Prototype B with other GISes they had used. I audiotaped these interviews, then transcribed selected portions to help me interpret video data and analyze breakdowns.

Once I had collected the data for this group, I analyzed them using modified versions of the methods used in Study 3.

**Participant Profiling**

Participant profiling, in this study, involved the following steps:

1. **Summarize demographics.** First, I examined questionnaire and interview data in terms of participants' demographics, including age, sex, years of experience with GISes, and years using various operating systems. These data helped me to understand how the students' backgrounds differed from those of the GIS-ALAS users in Study 4.

2. **Transcribe usage comments.** I reviewed interview data, transcribing portions relating directly to the following issues: (a) how participants perceived and managed Prototype B genres; (b) points at which they had trouble perceiving and managing those genres; and (c) complaints and suggestions the participants had about Prototype B. These transcribed portions served to more fully profile participants.

   I then stored these data in a database, where I could easily arrange, sort, and explore them.

**Video Coding**

Although I used a method for gathering and coding data that was quite similar to that of Study 3, in this study I concentrated on coding very different sorts of breakdowns. Study 3, an *in situ* study of experienced participants at work, allowed me to use detailed breakdown categories based on participants' interaction with the system. It assumes that
the participants are to some degree familiar with the system. But Study 5 does not make that assumption — *none* of the student participants are familiar with Prototype B, so they naturally encountered what were considered breakdowns under the old system: hesitations, lingering over menu choices, and so forth. These are to be expected from new participants.

For this study, I was more interested in breakdowns that halt or seriously impede participants' sessions. To identify such breakdowns, I invited participants to voice questions during the session. Then I coded their questions as breakdowns (Appendix G). I associated these breakdowns with the actions that students were performing as they asked the questions; like the video coding in Study 3, this video coding thus allowed me to compare successful and unsuccessful actions.

I tested inter-rater reliability using the instructions in Appendix G. The reliability test determined whether the coding scheme was systematized enough to be teachable to other raters. One rater was trained during an hour-long session by viewing and coding one observation, then comparing the result with my coding. After the training, the rater coded three sessions, each of which was randomly selected from one of three groups: (a) male native English speakers, (b) female native English speakers, and (c) non-native English speakers of both sexes. The rater coded 24.2% of the total video data. Table 2.5 indicates that the rater reliably applied the coding scheme.

**Case Studies**

After conducting the macro- and micro-level analyses, I used them to construct case studies (mesoanalyses) of all users' sessions.
1. **Tracing of breakdowns to discoordinations.** Once I had coded breakdowns, I investigated breakdowns by examining the video data where the breakdowns occurred, as well as stimulated recall interviews discussing these or similar breakdowns. These data gave me evidence to determine which breakdowns could be attributed to discoordinations related to genre perception and genre management.

2. **Tracing of discoordinations to contradictions.** Finally, I compared the identified discoordinations with the contradictions I identified in Study 2. These data allowed me to make connections among the micro, meso, and macro level analyses.

After concluding Study 5, I was able to draw conclusions about the four research questions that I state at the beginning of this chapter.

**Conclusion**

This chapter has outlined the methods that I used when conducting the five studies in this dissertation. The methods are based on the nascent methodology implied by the theory of artifacts. In the remaining chapters, I test, extend, and refine the parts of the methodology through analyses of the studies.
CHAPTER 3

GENRE ECOLOGIES AND THE LOCATION OF USABILITY:
A STUDY OF SOFTWARE DEVELOPMENT AS
MEDIATED ACTIVITY

Where is "usability" located? That is, working within a cultural-historical framework, how can we locate and conceptualize the difficulties that people encounter as they attempt to use information systems?

In this chapter, I attempt to answer that question. The question is important to information systems researchers, who often see usability as their primary goal, yet often do not examine the concept of usability in depth. In answering the question of where usability is located, I further develop a theory and methodology that can guide research into the design and evaluation of information systems. Such a methodology needs to propose a clear understanding of what usability entails; thus, this chapter provides a detailed examination of usability.

I begin by proposing the concept of genre ecologies, or interconnected and dynamic sets of genres that jointly mediate activities. Next, I use this concept to analyze a study of 22 software developers. At the end, I conclude that usability is a quality of the entire activity network rather than located in a particular artifact or user.

The Theory of Artifacts Applied to Compound Mediation

In Chapter 1, I forward a concept of artifacts as parts of the material environment that become meaningful to us as they are used in our various activities. In this notion, artifacts do not exist as things-in-themselves, but neither are they simply perceptual categories. Rather, they are defined by the confluence of material conditions and rules in a particular activity.

Such a conception is a starting point for the methodology I am attempting to construct. Yet to make the conception useful, we must be able to look beyond the individual artifact or even the historical development of a single artifact. As Chapter 1
briefly argues, our activities are mediated, not by a single tool, but by a range of different tools (e.g. Cole, 1996; Orlikowski and Yates, 1994; Russell, 1997). Thus we need a way to talk about *groups of artifacts that jointly mediate activities*.

But if that is so, then, how can we locate "usability"? In which tool? Or should we locate it in the interconnections between those tools — and if so, what do these interconnections mean for the evaluation of information systems?

In this chapter, I argue that genres can be conceived as belonging to ecologies of tools. These ecologies can serve as compound mediational means. I call these relatively stable configurations of genres *genre ecologies*. This concept of genre ecology allows comparisons between ecologies and within ecologies (i.e. the same genre used in different ecologies; different genres used in the same ecology). Given the interconnectedness of genres in an ecology and the possibility of a genre being a member of multiple ecologies, *usability is a quality that must be seen as distributed across an activity network*.

Below, I discuss genre ecologies in theoretical terms, then use an empirical study to demonstrate their utility for information systems design and evaluation.

**Ecologies of Genres**

The notion of artifact is a powerful concept for analyzing information systems. But if we stop there, we again run the risk of too narrow a perspective: we may be led to focus on a single artifact, and in doing so, disregard its multiple relationships with other artifacts in the same activity. We could also lose sight of the complexity of mediation.

Below, I use Susanne Bødker's developing conception of artifact to frame the need for a more ecological study of artifacts. Then I draw on Edwin Hutchins' notion of tool ecologies to discuss how *groups* of artifacts can jointly mediate activities. I conclude this section by discussing implications of an ecological perspective for artifact comparisons and for the concept of usability itself.
**Bødker's Developing Conception of Artifacts**

Bødker's work is the basis for much of mine: like her, I am interested in how users employ computer-based artifacts in their work and how they encounter breakdowns when using those artifacts. But in Bødker's early work the concept of *artifacts* is limited. Below, I use that early concept to point to the need for a more developed concept — one that I supply, based partially on Bødker's later work.

In Bødker's early work, "artifacts are things that mediate the actions of a human being toward another subject or towards an object" (1991, p. 34). That is, artifacts are conceived simply as *mediational means*. And it's not always clear what qualifies as a mediational means. For instance, Bødker says that "the basic role of the user interface is to support the user in acting on an object or with a subject through the artifact" (p. 77, emphasis mine). The interface, then, is not itself seen as an artifact. Neither is it seen as being a part of the artifact. Rather, it "supports" the user as sort of a layer between subject and object. Bødker does not make clear how this support is different from mediation.

Indeed, Bødker does not make clear in her early analyses exactly what she considers an artifact. Is it the physical computer itself? The software application? Are aspects of the interface (e.g. dialog boxes) artifacts in themselves? Clearly these aspects figure into the analysis (see pp. 88-89), but they are discussed as being "within" the artifact. They are also parts of the interface, which, as we have seen, Bødker describes as being separate from the artifact. It appears that Bødker's early notion of artifact is not scalable: an artifact does not appear to be composed of other artifacts.

Finally, Bødker's early idea of an artifact seems to emphasize the materiality of the artifact. In her discussion of artifact evolution, for instance, she discusses that evolution strictly in terms of the artifact's *design*. Yet as I argue in Chapter 1, artifacts' properties are as much a function of their users' practices as they are of the artifacts' materiality.

In her later work, Bødker develops a more complex notion of artifacts. She portrays artifacts as not simply mediational means, but means that reflect the state of praxis
Artifacts crystallize knowledge so that operations which are developed in the use of one generation of technology may later be incorporated into the artifact itself in the next" (1997, p. 150). These artifacts relate to one another, being situated in a "web of artifacts" (1996, p. 161). And they are scalable — a computer application, for instance, may contain several artifacts that jointly mediate activity:

An activity does not make use of a single artifact — rather, a number of artifacts are juxtaposed in their mediation of a particular activity. In other words, one computer application may contain several instruments, each making it possible to work on a certain object in the web of activities. (1997, p. 150)

Bødker chooses to call these groups of artifacts "juxtaposed" artifacts. Others have advanced similar concepts of compound mediation using different terminology. Some, like Bødker, focus on artifacts: Cole (1996) discusses "aggregations of artifacts" that collectively mediate activities, for instance, while Hutchins (1995) describes "tool ecologies" that perform computations. Others focus on genres to which artifacts belong: Freedman and Smart (1994) discuss "interweaving webs of genre," Bazerman (1994) describes "genre systems," and Orlikowski and Yates (1995) discuss "genre repertoires," all of which involve orchestrating textual genres to collectively mediate activities. All of these terms describe a group of artifacts that acts as a compound mediator of activities. It is this quality that I explore throughout the rest of this chapter.

**Genre Ecologies**

To discuss how such groups of artifacts mediate activities, I turn to Hutchins' "ecologies of tools."

**Ecologies**

Hutchins' concept of "ecologies" seems to me to be more developed than Bødker's or Cole's accounts of joint mediation, and more distributed than Bazerman's essentially
dialogic model of genre systems. Furthermore, it leaves room for non-textual artifacts in a way that Orlikowski and Yates and Freedman and Smart do not.

The artifacts that Hutchins investigates during his stay on a naval vessel, in fact, are dynamic and meaningful without being strictly textual. Astrolabes, compasses, sextants, and calculators are arrayed and employed by multiple users to transform data. Hutchins sees these multiple transformations themselves as part of a larger computation:

> It is important to consider the whole suite of instruments that are used together in doing the task. The tools of navigation share with one another a rich network of mutual computational and representational dependencies. Each plays a role in the computational environments of the others, providing the raw materials of computation or consuming the products of it. In the ecology of tools, based on the flow of computational products, each tool creates the environment of the others. (p. 114)

Although tools are portrayed as "providing" each others' inputs and "consuming" each others' outputs, Hutchins' tool ecologies are not simply stepwise arrangements of tools. Rather, the tools are connected in multiple, complex, and often nonsequential ways. They co-evolve: changes in one lead to changes in others. Functions sometimes move around from astrolabe to quadrant to cross staff to sextant (p. 113-114); such movement is made possible by the tools' interconnections in the ecology. The ecology itself — not its individual tools — is the mediator of the activity.

Yet for my purposes, Hutchins' analysis is limited by his focus on tools. Such a focus works well when discussing tool types that vary little from iteration to iteration. (One imagines that one Navy-issue pelorus is more or less identical to another.) But the artifacts in which I am interested (which include texts and non-texts) tend to vary quite a bit. For instance, in this chapter's empirical study, developers use program examples, embedded comments, and online reports in their activities. All of these vary from iteration to iteration in daily use; one might imagine one of Hutchins' sailors using a substantially different pelorus each time he has to take a measurement. Following Freedman and Smart's (1997) lead, then, I extend Hutchins' work by considering genres rather than tools.
Genres

Artifacts such as program examples, embedded comments, and online reports can be understood as genre instances and studied accordingly. In any given activity network, artifacts become familiar to users over time, so much so that users begin to interpret artifact types as instantiations of genres. These genres of artifacts collectively mediate the developers' activities, and in doing so they become interconnected with each other in mediational relationships. Such interconnected genres can be considered genre ecologies (cf. Russell, 1997).

The concept of genre is quite amenable to Hutchins' ecologies. Genre's focus on historical change can be extended to help understand the historical changes of an entire ecology; its focus on addressivity can be extended to understand how "each [genre] creates the environment of the others" (Hutchins, 1997, p. 114) in the ecology. At the same time, genre accounts for coherent collections of practices and semiosis (see Chapter 1), and those qualities themselves can be extended to the ecology.

The notion of genre ecology, then, can be a useful way of thinking about aggregations of artifacts. Particularly useful is its ability to foster complex analyses through inter- and intra-ecological comparisons.

Inter- and Intra-Ecological Comparisons

Once we explore an activity network in terms of genre ecologies, we find that all sorts of comparisons become possible. To demonstrate, let's return to Bødker's early work for a moment.

In Bødker's early conception, a software application is an artifact, and when she makes comparisons between artifacts, it's usually at this broad level. For instance, she compares two word processors (1991). And although the comparison gets more specific (she talks about interface features such as footnotes), it is limited in two ways. (1) Bødker does not have a basis for defining these features in more than a purely formal
way. (2) She does not attempt to make comparisons among interface features within a particular application.

To overcome limitations such as these, I propose to extend Bødker's early work, but I discuss compound mediation using the highly articulated framework of genre ecologies. Reinterpreting Bødker's example of word processors in terms of genre ecologies, for example, can enable us to make both inter-ecological and intra-ecological comparisons.

**Inter-ecological comparisons.** If we interpret interface features as genres, we are able to compare genres between the two interfaces (ecologies). For instance, we could compare two footnote functions, two dialog boxes, or two writing spaces between the two word processors (or among several word processors), and we could do so in terms of history and addressivity. Such an analysis could allow us to compare two rival applications in systematic yet nonformalist terms. Or it could allow us to systematically compare two or more generations of the same application (see Chapter 4).

**Intra-ecological comparisons.** Furthermore, interpreting interface features as genres allows us to examine the connections among genres in a particular interface. How do footnote functions, dialog boxes, and writing spaces relate to each other? How are they interconnected? What different operations do they require users to learn? Do the different genres require different assumptions or operations? Are they designed in response to different activities?

But the notion of genre ecology also encourages us to push beyond the confines of the interface. After all, users of word processors do not engage in unmediated communion with the machine. As I type these words, I am working from handwritten notes on a previous draft; consulting various articles and books into which I have inserted notes; and sometimes checking the word processor's manuals and online documentation. Some of these are on-screen genres (displayed on the computer screen), whereas some are off-screen genres (separate from the computer), but all of these artifacts mediate my writing; they are all part of a genre ecology.
Genre ecologies, then, highlight the interconnectedness of artifacts, both genetically (through the concept of genre) and addressively (through the concept of ecology). It is that interconnectedness that problematizes the notion of usability.

*Usability in Terms of Genre Ecologies*

Now that I have discussed genre ecologies, I turn to the question of where usability is located. In North America, usability is typically understood as a quality of an individual artifact: usability specialists test how usable an artifact is, often through controlled experiments; identify usability problems; and suggest ways to redesign the artifact to avoid such problems in the future. Suggestions typically amount to "mere simplification at the level of operations and tasks," simplifications that can lead to the de-skilling of workers, "imperiling the opportunity and challenge that should inhere in the design of work" (Carroll, 1996; see also Adler and Winograd, 1992).

If we are to take the interconnectedness of artifacts seriously, we must realize that usability cannot be the quality of a particular artifact because that artifact only makes sense in terms of its genre ecology and the activity network in which that ecology is employed. *Usability is a quality of the entire network, and it is distributed across the network.* Simplifying a given artifact cannot *in itself* lead to usability — that is, it cannot *in itself* provide the artifact's users with more freedom, competence, confidence, and ability to complete the tasks they wish to complete — because the quality of usability is not that of a particular tool but of the multitude of connections among tools, rules, users, communities, divisions of labor, objects, and outcomes (cf. Mirel, 1988).

Below, I study three related workplaces in terms of genre ecologies. I compare genres between and within the genre ecologies of these three workplaces. Throughout, I explore and problematize the concept of usability.

---

8For instance, Dumas and Redish (1993) state the "usability is *an attribute* of every product" (p. 4, emphasis mine).
Overview of the Study

I conducted this study at Schlumberger Oilfield Services as a member of the Usability Services for Engineering Research (USER) team. The study, conducted over 10 weeks during the summer of 1997, involved observations and interviews of 20 software developers and interviews of two additional developers. I collected these observations and interviews to inform the design of an information system meant to provide quick access to information on library routines and datatypes, the "building blocks" of Schlumberger's code. The proposed system was conceived as a usability-oriented solution, since it responded to perceived difficulties users had with comprehending and producing code.

In the study, I concentrated on the mediational means that developers used as they comprehended and produced code. As I argue below, the developers' comprehension and production of code was mediated by the ecologies of genre that surrounded their work.

Background: Questioning the Usability of Code Libraries

Schlumberger is a multinational corporation that provides information tools for the oil industry. Currently, Schlumberger's oilfield computer systems run more than 30 million lines of code dedicated to analyzing and interpreting the seismic and drilling data that are collected by Schlumberger equipment at drilling sites. This code is developed in-house, and software developers continue to generate more code as they maintain and develop products. Naturally, Schlumberger encourages developers to reuse existing code from the corporation's code libraries — collections of commonly useful datatypes and routines (explained below) that are accessible to all developers working on a given project. Yet developers sometimes have a difficult time finding appropriate code in the libraries; consequently, they "reinvent the wheel," writing new code that is not always as efficient or well-tested as the existing code in the libraries. Since the code libraries can change daily, printed reference guides and even static online guides quickly become out

---

9For another, shorter account of this study, see McLellan, Roesler, Tempest, and Spinuzzi (1998).
of date. In-house quantitative studies on code-sharing suggest that developers are not availing themselves of the routines in the code libraries (McLellan, Roesler, Fei, Chandran, and Spinuzzi, 1998).

Schlumberger's library code contains two related types of code: datatypes and routines. Datatypes are units for storing information; these units can contain complex combinations of integers, floating-point numbers, characters, and other kinds of data. These datatypes are manipulated through routines: chunks of instructions that involve transforming the data stored in datatypes. Developers can use a library's datatypes and routines within a given code library in any program that explicitly refers to that library.

The Study: Investigating Usability

The USER team interpreted the low levels of code-sharing as a possible indicator of a usability issue. As a result, I conducted a study meant to investigate the tools that developers use as they write code, with a particular interest in how the USER team could modify or introduce tools to better support program comprehension and production.

I originally conducted this study with the following research questions:

• How are developers currently finding appropriate library code to use? That is, how do users find out about the datatypes and routines that currently exist in Schlumberger's libraries?

• How do the developers use mediational means (such as search tools, features of the code, communication media, and manuals) when comprehending and producing code?

These questions brought me to ask the research question I attempt to answer in this chapter: Where is usability located? Below, I discuss the research sites I visited and the participants I observed and interviewed.

Research Sites

Schlumberger consists of a number of related large business groups. I visited three of these groups during the course of my research, groups that use similar but different,
interrelated genre ecologies. (In a closer analysis, we might study each site as a separate activity network that shares strong commonalities with the others because they all engage in the more general activity of Schlumberger software development.) These three business groups, originally separate organizations, were acquired by Schlumberger at different times during the last decade. Much of their code dates from that pre-Schlumberger time. This code is called "legacy code."

At each site, I observed and interviewed a group of software developers roughly proportional in size and gender makeup to the total population of the site. Interview questions and observations focused on the mediational means that they used as they produced and comprehended code.

Site 1 maintains a compilation of seismic processing systems that has been unified into a seismic data interpretation system. This system is mainly written in FORTRAN, although some new additions are being written in C. Developers work on a UNIX platform. I observed two developers at work and interviewed two others. These developers, like most at Site 1, are oriented towards internal users: that is, they write code that is primarily used by Schlumberger employees and have contact primarily with Schlumberger engineers.

Site 2 maintains a variety of data management products used to interpret oilfield data. This system is mainly written in C and C++, although two of the developers I interviewed were porting (i.e. translating) an existing product to Java. Like the developers at Site 1, these developers work on a UNIX platform. I observed eleven developers at work here. These developers, like most at Site 2, are more oriented to outside customers: they tend to review the system's user manuals to find out how their code is "supposed" to work (from the users' standpoint) and seem to spend more time on details like the user interface.

Site 3 maintains products for extracting, processing, and interpreting data on location, as it is collected at the drilling site. This system is written primarily in C and C++. Unlike the other two locations, Site 3 is a "Windows shop": developers work on
Table 3.1  Participants in the Schlumberger study.

<table>
<thead>
<tr>
<th>SLB</th>
<th>Loc</th>
<th>Cat</th>
<th>Language</th>
<th>SLB Time (Years)</th>
<th>Project Time (Years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Site1</td>
<td>Dev</td>
<td>C/C++</td>
<td>5</td>
<td>.8</td>
</tr>
<tr>
<td>S2</td>
<td>Site1</td>
<td>M/D</td>
<td>FORTRAN</td>
<td>8.0</td>
<td>8</td>
</tr>
<tr>
<td>S3</td>
<td>Site1</td>
<td>M/D</td>
<td>FORTRAN</td>
<td>.4</td>
<td>.4</td>
</tr>
<tr>
<td>S4</td>
<td>Site1</td>
<td>Dev</td>
<td>FORTRAN</td>
<td>11</td>
<td>.5</td>
</tr>
<tr>
<td>S5</td>
<td>Site2</td>
<td>M/D</td>
<td>C++</td>
<td>3.5</td>
<td>1.2</td>
</tr>
<tr>
<td>S6</td>
<td>Site2</td>
<td>M/D</td>
<td>C</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>S7</td>
<td>Site2</td>
<td>Doc</td>
<td>C++</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>S8</td>
<td>Site2</td>
<td>Doc</td>
<td>C++</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>S9</td>
<td>Site2</td>
<td>M/D</td>
<td>C++</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>S10</td>
<td>Site2</td>
<td>M/D</td>
<td>C/C++</td>
<td>7</td>
<td>.8</td>
</tr>
<tr>
<td>S11</td>
<td>Site2</td>
<td>Doc</td>
<td>C++</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>S12</td>
<td>Site2</td>
<td>M/D</td>
<td>C</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>S13</td>
<td>Site2</td>
<td>Dev</td>
<td>Java</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>S14</td>
<td>Site2</td>
<td>Dev</td>
<td>Java</td>
<td>1.5</td>
<td>5</td>
</tr>
<tr>
<td>S15</td>
<td>Site2</td>
<td>M/D</td>
<td>C</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>S16</td>
<td>Site3</td>
<td>M/D</td>
<td>C</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>S17</td>
<td>Site3</td>
<td>M/D</td>
<td>C</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>S18</td>
<td>Site3</td>
<td>Dev</td>
<td>C++</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>S19</td>
<td>Site3</td>
<td>M/D</td>
<td>C</td>
<td>8</td>
<td>9.5</td>
</tr>
<tr>
<td>S20</td>
<td>Site3</td>
<td>M/D</td>
<td>C</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>S21</td>
<td>Site3</td>
<td>M/D</td>
<td>C</td>
<td>8</td>
<td>.5</td>
</tr>
<tr>
<td>S22</td>
<td>Site3</td>
<td>Dev</td>
<td>C++</td>
<td>7</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Microsoft Windows NT workstations rather than UNIX workstations. I observed seven developers at work here. These developers, like those at Site 1, are oriented toward Schlumberger employees who use their software in the fields.

**Participants**

Participants drawn from the three sites have a wide variety of experience and constitute a complex division of labor, categorized by phase in the production cycle: development (Dev), maintenance/debugging (M/D), and documentation (Doc).

**Development (Dev)** involves writing new code that extends the functionality of the existing code. Development usually occurs near the beginning of the production cycle. Developers are typically given requirements and specifications documents that spell out what functions are needed, what they should do, and how they should interact with existing code. Developers in this part of the cycle need to understand library code so that
they do not "reinvent the wheel" by writing stretches of code that already exist in the library.

**Maintenance/debugging (M/D)** involves updating or repairing existing code. Maintenance and debugging typically occur near the end of the development cycle; more and more developers are given maintenance and debugging duties as the code's release date approaches. Developers in this part of the cycle need to understand library code so that they can better comprehend the existing code, make it more efficient, and substitute library calls for existing, redundant code introduced during the development phase of the production cycle.
Documentation (Doc), in this category scheme, refers to "process documents" written by developers during the course of their work cycles (such as design documents, requirement specifications documents, dataitem reports, and release letters). Developers usually write documentation near the beginning of the cycle, before development starts (design and specification documents) or near the end of the cycle, just before the code is released (dataitem reports, release letters). This documentation frequently details new items to be added or that have just been added to the library.

Table 3.1 shows the participants, locations, job categories (on the day I interviewed each participant), programming language primarily used, years that they have used that language, years they have worked at Schlumberger, and time that the participants have worked on their present project. The italicized participants were interviewed but not observed.

As Table 3.1 shows, the interviewed developers have a wide variety of experience. Nevertheless, developers tend to have strong commonalities within and across the three sites. As I argue below, the developers' work was mediated by the genre ecologies at the three sites as they attempted to find suitable code and to comprehend and produce code.

Analysis: Ecologies of Genre and the Nature of Usability

Using the data from the interviews and observations, I constructed Figure 3.1, which shows the overall activity network of the developers. Bear in mind that this is a general activity network that extends across sites. In the following analysis I will explore the more specific activity networks of the individual sites, based on differences in objects, communities, and so forth, drawing out some of their differences. Of the most interest to us, though, is the abundance of genres in the ecology and their complex interconnections.

As Figure 3.1 shows, the developers operated within a complex activity network with a variety of mediational means, a system of rules governing the use of those means, and a complicated division of labor. In the analysis below, I focus on these three elements. But the different sites are also activity networks in their own right: their activities differed significantly in mediational means, rules, and objects. Some of these differences are
Figure 3.2  The genre ecology of the developers' activity network. Numbers in parentheses mean that the indicated genres are only used in the ecologies of the enumerated sites. Lines indicate mediational relationships among genres.

The genres (here, depicted as the confluence of mediational means and rules) have complex relations with each other, forming an ecology of genres that developers employ to find specific types of information about the code that is the object of their work. As Figure 3.2 depicts, these genres are connected in a variety of different ways.

It's important to note here that at all three sites, the elements of the activity network (Figure 3.1) are substantially the same. The mediational means, for the most part, are materially the same, available across all sites, and organized in generally the same genre ecology. The object (the code) is more or less the same — in fact, much of it is shared among sites. The division of labor is at least formally the same across sites. Yet within
these different sites (communities), the unwritten rules governing mediational means are sometimes quite different. Consequently, developers sometimes perceive the mediational means governed by those rules as having different uses, possibilities, and levels of importance at different sites. These genres are constrained by the characteristics of the artifacts and the code, as well as the historical development of practices and the interrelationships of genres at the various sites.

**Genre Ecologies at the Three Sites**

Figure 3.2 shows the genre ecologies at the three sites. The three ecologies are similar — thus the single diagram. Numbers in parentheses mean that the indicated genres are only used in the ecologies of the enumerated sites. For instance, developers use grep (a software tool used to find strings of characters in text files) at all sites, but they use grep scripts only at site 2.

The lines indicate mediational relationships among genres. That is, if two genres are connected by a line, I observed those two genres being used in conjunction: one mediates actions performed on the other. For instance, developers at all sites interpret project code by consulting third-party programmer manuals, code examples, and comments, so in Figure 3.2 these genres are all linked to the source code by lines. On the other hand, developers do not use Schlumberger's user manuals to help them interpret comments, so these genres are not linked.

Figure 3.2 gives us a partial idea of what genre ecologies look like at the three sites. However, it is just a partial idea; a more extended study might turn up dozens or hundreds of genres and might make it practical to subdivide genres (such as online texts).

In addition, the diagram gives us an informal method of identifying genres that are densely connected. For instance, the object of the developers' work is the project code, which is connected to nearly every other genre in the ecology. On the other hand, Schlumberger user manuals were used for only one purpose, to understand the finished software.
Finally, the diagram helps us to understand how usability is problematized by an ecological perspective. If a user has difficulty using one genre (e.g. a code library), is the difficulty located "in" the library? Or in one or more of the many genres that have mediatory relationships with it? Or in the relationships themselves? Or the rules applied in this particular activity network?

Below, I compare some of these genres between ecologies and within ecologies. In doing so, I explore the distributed nature of usability.

**The Generic Nature of Artifacts: Inter-Ecology Comparisons**

In Chapter 1 I argued that artifacts can be seen as the confluence of rules and the material environment. In this analysis I discuss how the three sites, although using the same materials, used them in quite different ways. I analyze these artifacts in terms of genre. In doing so, I attempt to locate and conceptualize usability issues.

Figure 3.2 depicts the three ecologies "overlapped" as it were. Most of the genres are in all three ecologies and connected to other genres in similar ways. But there are a few exceptions, and those exceptions are telling ones: they suggest that similar genres are perceived as having different affordances, different uses, and different degrees of usefulness at the three sites, partially because of how these genres interact with other genres in the ecologies. Below, I discuss two examples — the grep utility and comments embedded in the code — and use them to demonstrate that usability is distributed across the activity network.

*Grep and Grep Scripts*

The grep utility is a program that searches for specified strings of text within sets of files. This program is a utility that was originally developed for the UNIX platform (used by developers at sites 1 and 2), but has also been ported to the Windows NT platform (used by developers at Site 3). Some developers at Site 2 have augmented grep's capabilities by assembling *scripts*, or collections of commands that invoke grep for specific files in specific directories. Users could run a script by typing its name at the
command line. So, for instance, a developer might assemble a script of grep commands for a certain project and call the script "pgrep." Then, to search the appropriate files for a certain string of letters — say, the routine name "reversestr" — the developer could type `pgrep reversestr` at the command line. Figure 3.3 shows a fictional grep script similar to those written by developers.

Six developers were observed using grep; in the interviews, 12 discussed using grep for finding information in the libraries.

The grep utility is used repeatedly and cyclically by developers to accomplish repeated actions within their activities. And the repeated use gives rise to stabilized-for-now rules (habits) for operating it. *These rules are not simply embedded in the software:* developers used grep with strikingly different rules at the three locations. Rather, the rules are demonstrably affected by other available genres, the search environment, and the community's distributed knowledge. Use — and usability — differs from site to site, although the grep program itself does not.

**Grep use at Site 2.** For instance, developers at Site 2 tend to create grep scripts that search specific files stored in specific paths\(^\text{10}\) related to the developer's current project. The institutional practice of storing projects' files in baselines, and the industry-wide practice of affixing standard suffixes to file types, contribute to the utility of this practice. These scripts are not simply passed from one developer to another — as one developer told me, each developer knows enough about grep and various scripting languages to create these scripts for themselves, and in any event each developer searches different baselines, so the scripts must be different for each developer. And, the developer pointed out, a grep script is an *obvious solution* to the limitations of grep. Yet the practice of creating these scripts is widespread at Site 2 and unheard-of at the other two sites. Although developers at each site had access to the same tools (grep and some scripting language), grep is a *qualitatively different artifact* at the three sites — it has different uses and meaningfulness — because the rule of scripting has only developed at

\(^{10}\)A *path* is a list of specific subdirectories.
grep $1 /home/jones/project/fileio/*.c
grep $1 /home/jones/project/screenio/*.c
grep $1 /home/jones/project/fileio/*.h
grep $1 /home/jones/project/screenio/*.h

Figure 3.3  A (fictional) sample grep script, similar to those used by developers at Site 2. In each invocation, $1 is automatically replaced with the string to be found.

Site 2.\textsuperscript{11} (Unfortunately, this ten-week study did not allow me enough time to collect historical data that might suggest why the sites developed different rules.)

To illustrate the qualitative differences of grep across sites, I contrast Site 2's practice of scripting with that of information searching at Site 3 and Site 1.

**Grep use at Site 3.** Site 3's developers use grep extensively, but only for searching files in a particular directory. Developers conduct searches across paths using a separate tool, Microsoft Developer. Many of Site 3's developers preferred grep because it is faster than MS Developer, but none have assembled grep scripts — although those scripts, like individual invocations of grep, would theoretically be faster than MS Developer’s search feature. When I asked some developers at Site 3 about this difference, they indicated that the idea of grep scripts simply hadn't occurred to them — and that they probably would not write grep scripts in the future. The initial, intensive labor involved in writing the script sounded like too much trouble compared with the less intensive but repeated labor of using MS Developer.

The usability of grep, then, is conceived differently at Sites 2 and 3. At Site 2, usability entails the automation of repeated tasks such as searching multiple baselines. This automation requires initial investment, but minimal interaction each time the baselines are searched. That is, the Site 2 developers were willing to devote substantial time to devising grep scripts, believing that the initial effort would pay off in easier searches. Site 2's developers have introduced a new genre into the ecology, that of grep scripts, and in consequence grep has become more useful to them.

\textsuperscript{11}In the terminology of Norman (1988), grep affords searches across entire paths at Site 2, but does not afford that capability at the other two sites.
At Site 3, on the other hand, usability entails little initial investment and moderate to heavy interaction each time the baselines are searched. Site 3 developers devoted no time to devising scripts; rather, they used a tool that required more interaction each time it was used. Rather than introducing another genre into the ecology, Site 3's developers have distributed the task of searching between grep and another existing genre, the text editor.

Grep use at Site 1. At Site 1, developers use grep to search specific directories, but do not develop grep scripts either. In fact, practices of baseline-wide searching did not become apparent in either the observations or the interviews, except in one case: one developer did demonstrate an analogous script for searching a database of code, using database commands rather than grep scripts. This script allows him to find the sought code within the code archives rather than actual paths. The database script is simple, but this developer must type it each time, since he has not created a permanent electronic version. To aid him, he printed the script and taped it to the desk beside his keyboard.

At Site 1, then, this developer conceives of search usability in a third way. Usability entails moderate initial investment (typing out and printing a script) and moderate interaction (using the script to guide the developer's own actions each time). By introducing a new genre into the ecology, the script, Site 1's developers have found a third way to mediate their searches.

Grep use and text editors. At all three locations, developers use text editor-based searches rather heavily. Site 3 developers use MS Developer to conduct searches across paths as well as through individual files; developers at other sites do not, although that capability exists in some of the editors they use. In general, developers use grep and grep scripts to identify files with the needed information, then open individual files in a text editor and use its search function to find spots of interest within those files. The developers then glean information by viewing the found line of code as it functions in the surrounding code.
The data above suggest two things. One is that, to paraphrase Hutchins (1995), genres embody distributed cognition. That is, a genre — such as, say, a grep script — is a material solution to a problem once faced by its originator, a solution that was successful enough to be used repeatedly by others. When users apply a genre to a problem, in a sense the problem is being partially solved by those who developed the genre. Over time, the genre becomes familiar enough to its users that it is perceived as obvious (and in some cases trivial). That effect is difficult to see when one is embedded in the community using the genre. The developers at Site 2, for instance, were familiar enough with grep scripts to consider them an "obvious" solution, not an innovative answer to a difficult problem. And this is why comparative studies of genre use (such as this one) can be valuable: by comparing genre use at different sites and in different genre ecologies, we can reexamine artifacts as meaningful and culturally-historically developed solutions rather than things-in-themselves.

But rules are not "designed into" these artifacts, as a simplistic reading of Bakhtin, Leont'ev, or Hutchins might suggest. Tools that are materially identical, such as the grep program, are paired with different rules at each site, resulting in qualitatively different genres with divergent jobs and different perceived abilities. That is, the tools' affordances have as much to do with the sites' cultural-historical practices as they do with the material tools themselves. Use and usability extend beyond the boundaries of any given artifact; they describe how the artifact interacts within its genre(s), genre ecologies, and activity networks. At the three sites, many of the material tools did not vary, but the rules did, and that variation resulted in artifacts that were qualitatively different, that is, perceived as having quite different uses and possibilities.

Finally, interactions among genres can influence genre use. For instance, all three sites face the same problem: how to find code across multiple paths. Although all three have the same basic tools (grep, text editors, databases), each site has adapted a different tool to occupy this ecological niche. In doing so, they have made compromises. Site 2

\textsuperscript{12}cf. Bakhtin's theory of genre, especially Bakhtin (1986).
developers use the genre of the grep script, a genre that requires some initial time investment to write but that is rewarding for those who spend a great deal of time working with the same paths. Site 3 developers use MS Developer, which takes time to start and search, but does not require an initial time investment and provides an easy-to-use Windows interface. And one Site 1 developer opted for a database script, which is shorter than a grep script but must be typed each time.

Comments

Another genre that illustrates the distributed nature of usability is that of comments. Comments are messages that are embedded in both library and project code by developers. (In this study, developers only used comments embedded in the project code.) Comments do not have any effect on how the computer runs the program: they are not included when the program is compiled (i.e. turned into machine code that can be processed by the computer). Comments are generally assumed to be notes that developers include to help later developers interpret the code (see Takang et al., 1996, Tenney, 1988), although in this research I found many comments that had other uses. Figure 3.4 shows C-style comments embedded in the code between the character combinations /* and */. When asked about how they comprehend routines and datatypes in the libraries, eleven developers mentioned looking at the comments to aid comprehension — although, as I explain below, developers at the three sites used comments in quite different ways.

Comment use, like grep use, differs by site. For instance, at Site 1 and Site 2, developers tend to read, write, and maintain multiline comments as a rule. Developers use these comments as notes for interpreting code, but also to signal plans for maintenance, as in the comment shown in Figure 3.4. (The product name has been changed to protect confidentiality.)
Such comments are common in the Site 1 and Site 2 code, and provide important resources for interpreting the code in relation to future changes, as well as coordinating work. Developers at these sites often remarked on the utility of these comments both for sharing information with others and for reminding themselves of changes. These developers are scrupulous about documenting code with comments.

On the other hand, developers at Site 3 are far less likely to write or read multiline comments. At this site, developers have continually maintained and updated legacy code dating from the early 1980s, but they have not updated comments dating from that time, subverting the comments' usability as an interpretive tool. One developer's explanation is typical:

CS: Do you find these comments to be very helpful?

S16: Not in the slightest.

CS: Really? Why not?

S16: Well, people put in comments and make changes and they never keep up with the comments. The comments are — I don't know. That's about like if you're gonna drive down the road, and you're wantin' to get out on the street, and you see someone comin' down the road, they've got their blinker on, okay, are you going to trust that that car is going to turn?

The attitude expressed by this participant was pervasive throughout Site 3. Rather than being helpful aids to interpretation, comments were perceived as actually misleading. Some developers actually regarded multiline comments as evidence of poor coding rather than tools for interpreting code or coordinating work:
CS: ... you said that ... you try and make [your code] very clear for the other people who are going to be maintaining it after you. Do you put in a lot of inline comments in your code?

S18: If it's necessary. Not too many comments. Because, uh, to me, if you look at the program that has lots of comments, it means the program wasn't structured right. If you have meaningful variable names and meaningful data structures set up, organization in your program, you don't need to have that much comments. You need to have enough comment not to clutter the code. Because you can have so much comment to confuse the user. You know, have more comments than code.

At Site 3, criteria for "good code" — which is to say, usable code, interpretable and structured code — go far beyond explicit criteria such as robustness and efficiency.

This low regard for comments at Site 3 gives rise to interesting uses of mediational means, uses that are not found at other sites. For instance, one developer sets his text editor to change the color of comments in the code he maintains:

S20: But [the editor's feature of displaying text in different colors is] nice especially in things like the comments. Now for me, it, it's probably not, useful in the way that it was intended, see I use it to almost get rid of the comments, I make them a light grey color so they don't stand out, so it's easy to glance and see where the code is, and not be distracted trying to, you know, with something that's a comment.

Another developer used comments primarily as landmarks for navigating through the code, rarely if ever reading them to understand the code's workings:

CS: I noticed there weren't a lot of comments inline in the code. Do you find those very helpful?

S21: Inline comments? Yes, absolutely. It helps to separate the code, instead of having one big chunk of commands. With a small header for each section, it's a lot easier to get to it when you're looking for a part of the code.

CS: When you're looking at comments, are you looking for, basically, your place in the code, or are you looking for information on specific variables and functions, or are you looking at kind of an overall view of a, of a function or a method or an object?
S21: Well, when it's your own code, and you see it every day for hours, you're not reading comments anymore. You try to type them in — you try to keep them short, first of all, not more than one line. And then make sure that you accurately describe what happens in those few lines of code, but eventually that code changes, so the comments may remain, sometimes they get outdated, most of the time they are. But I use them for locating my code, really. If I have 50 lines of code without a comment I get lost. It takes me a while to actually read the code and find out what it's doing. But if I have comments I can separate it into sections, and if I know it's the second section in the function, I can go right to it. But, I don't read the comments more than once or twice after I type them in. They're just markers for sections of code. [Emphasis mine]

Like grep, then, comments were different genres at the three sites — not because of inherent characteristics of the comments themselves, but because the production, comprehension, and use of comments depend on their history within the three activity networks and the three genre ecologies that mediate them. The genres developed to support different kinds of work, that is, to support different sorts of usability by interconnecting with different practices at the three sites. At two of the sites, comments have been historically valued and maintained, and consequently developers see the value in reading, believing, and producing them. Comments had become central to some of the actions carried out at those sites. But at Site 3 comments have been historically disregarded, and consequently developers find ways not to read them, or to read them in different ways, as placemarkers and as indicators of code quality.

Furthermore, these developers avoid maintaining and producing multiline comments, ensuring that future developers at this site will continue not to use such comments to comprehend or produce code. So comments at Site 3 constitute a quite different genre from those at Sites 1 and 2. This has implications for usability. What is a "usable" comment? One that provides resources for interpreting and maintaining code? Or one that provides code separations and allows developers to evaluate code quality? The answer depends more on one's activity network than on the material qualities of the comment itself. Usability, then, is not located in the comment, or even in the developers' practices; it is more usefully conceived as distributed across the entire activity network.
Up to now I've selected single genres and shown how their uses differ from activity network to activity network. At this point, I turn my attention to inter-ecological comparisons, that is, how different genres in the same ecology relate to each other. My analysis of how genres relate within an ecology suggests that usability is not merely distributed between a genre and an activity network: it is distributed among genres as well. That is, a genre's usability depends partially on the other genres with which it is used.

**Ecological Relations Among Genres**

In this section I focus on how genres interrelate with each other in the same ecology. I explore how developers relate code examples to other genres in the ecology. Next, I explore how they use three genres of direct communication. Finally, I analyze the repercussions of these ecological relations for the concept of usability: that usability must be seen as distributed across interrelated genres as well as the activities they support.

**Code Examples**

Much research into computer program production and comprehension suggests that programmers approach their programs in a way that can be characterized as *generic*. Over the last two decades a set of theoretical models have been developed that view program comprehension and production as an interrelated set of habits and hypotheses that programmers form. These models of programming tend to use terms such as "schema" (Rist, 1989; Rist, 1991), "beacons" (Brooks, 1983; Wiedenbeck, 1986; Wiedenbeck, 1991), and "plans" (Soloway and Ehrlich, 1984; Boehm-Davis, Fox, and Philips, 1996; Pennington, Lee, and Reder, 1995). These models tend to portray programmers as sectioning code into interpretive (rather than structural) units and examining or producing those units cyclically rather than linearly. Although this is not the place to go into a detailed analysis of such models, they describe programming as an

---

13 For a brief discussion of some of these articles in terms of code examples, see McLellan, Roesler, Tempest, and Spinuzzi (1998).
activity that involves the development and refining of stabilized-for-now interpretive strategies for understanding code — that is, generic strategies (e.g. Schryer, 1993; Bakhtin, 1986; Bazerman, 1994; Russell, 1997).

Thinking of code as generic helps us to understand a common activity of software developers: examining one piece of code to understand another. The project code becomes an indirect mode of communication for the developers, a sort of narrative that developers can use to find useful library code and understand how it is used.

Using code examples with libraries. Looking at the library code allowed developers at all three sites to learn certain (abstract, definitional) things about the routines and datatypes — for instance, the arguments to be passed to the routines, the makeup of the datatypes, and the specifics of inner workings such as memory usage. These are all data that developers tend to use when they (a) know the exact name of the routine or datatype to examine, (b) are just learning the inputs of the routines, or (c) need to know specific information about the inner workings of the code.

But library code gives little information about the uses to which the routines and datatypes could be put. And workers at all three sites found that they needed contextualization of the code far more often than they needed specific information. This quote, from a developer at Site 2, is typical:

CS: The examples are fairly important to you?

S14: Yeah. Examples are important because basically they're, they're kind of a parallel proof of concept. Or they show what the boundary conditions, what you can do and can't do. Because just the definitions are not appropriate alone. When you want a certain function to do something, definitions are great. If you're scanning through this here, you want to know the constraints of it. But if you want to know its usage, the usage scenarios, you gotta have examples.

Here, S14 discusses using code examples — examples in living code as well as examples in a book — in conjunction with other genres such as code libraries and programmers' references (which provide "definitions" and "constraints" of functions). The various genres are different resources with different roles in the ecology.
Using code examples with search genres. In addition, looking at project code provided developers with a sort of reverse search mechanism. Recall that if developers are to use search genres such as grep scripts, text editor searches, or database searches to locate code, they need to know the exact name of a routine or datatype when searching for it in the library code. But if developers do not know the exact name — a situation that happens more often than not — they tend to look in the project code for a stretch of code that does something similar to the task they want to accomplish. They find examples of procedures in the code, procedures that include calls to library functions that they find useful in their own code. Code examples complement the other search genres: rather than looking up a function to find scenarios in which it is used, they look up scenarios to find functions they might use.

That is, genres of text searching (i.e. text editors) must be coordinated with the genre of the example if developers are to find the proper information. The labor of finding information is complex. Developers may have to "work both ways," from examples to formal definitions and from formal definitions to examples:

(A) Examples to formal definitions. An interesting routine in example code might lead the developer to search the project code and libraries for a formal definition of the routine.

(B) Formal definitions to examples. A definition in a code library, manual, or technical specifications might lead the developer to search the project code for examples.

If only one genre were available, it would be less usable, because developers use the genres together to comprehend the code. And these genres work in concert with other genres in the genre ecology: manuals and technical specifications.

In fact, part of the reason for Schlumberger's "code bloat" (McLellan, Roesler, Fei, Chandran, and Spinuzzi, 1998) was that this synergy is not always sustained. Developers rely on examples which do not always fit their tasks, and as a result they do not find the more appropriate routines and datatypes. Furthermore, developers sometimes use these examples to avoid writing code: they copy an example, paste it in the spot where they
want new code, then modify the example to meet their needs. Such code sometimes contains references to routines and datatypes in the code libraries. The developers don't necessarily need to know what routines and datatypes originate in the library, and in fact, they often seemed to be unclear as to exactly where some routines and datatypes originated. One developer at Site 3, for instance, who had been on the job for only a few months, revealed to me that he does not always understand the code he produces. Rather, he programs as a bricoleur: he finds similar stretches of code, pastes them into the proper spot, and tweaks them until they work. This nonhierarchical programming style has been observed by other researchers into program production and comprehension — and it can be a very successful strategy (e.g. Lange and Moher, 1989, Rosson and Carroll, 1996; see Lay, 1996 for a feminist perspective). Indeed, this bricoleur strategy should sound familiar if we think about the code in terms of genre. Students learning to write memoranda, for instance, often start out by simply copying an example's memo heading.

Project code, then, is often approached as narrative descriptions of routines and datatypes. By examining project code, developers at all three sites gain practical (although not always detailed) knowledge of relevant library code. Code is used for other things, of course — such as building programs. But these developers have developed certain ways to use project code and comments to mediate their various activities related to searching, comprehending, and producing code.

Developers mostly work with the project code and the comments embedded within it. For the most part, they are not interested in the libraries' abstract descriptions of datatypes and routines. Rather, they want to see how the routines and datatypes had been used in concrete ways, for concrete purposes. Once they have seen routines and datatypes used in one or more examples in the project code — in familiar contexts, dealing with familiar problems — they generally do not need more abstract description. To use a literary analogy, developers prefer to learn vocabulary from the code's many narratives rather than from the dictionary definitions supplied in the libraries. In fact, one
participant described the reading of code as "just like an English statement, like reading an English book."

The usability of code examples, I suggest, is best described as located in the relationships of the many genres used in the activity. Code examples are themselves used as a genre, one that is coordinated with other genres in the ecology such as libraries and search mechanisms. Although code examples help developers by providing narratives of code-in-use, they can also inhibit code development in the long term: Copying and modifying in this manner can propagate mistakes and misunderstandings in the code, and contributes to "code bloat" (McLellan, Roesler, Tempest, and Spinuzzi, 1998). Again, usability is best conceived as distributed across the genre ecology: code examples support the use of libraries (and vice versa), yet if the examples are copied, they subvert the very purpose of libraries by introducing repeated, unneeded code.

Conclusions

At the beginning of this chapter, I asked the question: Where is usability located? That is, working within a cultural-historical framework, how can we locate and conceptualize the difficulties that people encounter as they attempt to use information systems?

Although usability is customarily seen as a property of a single artifact, this study suggests that usability is distributed across the entire activity network. That is, it is located within the entire activity of people and their tools, not just the individual tools that mediate that activity or even the individual practices taught to the users. Usability issues cannot simply be addressed by redesigning an artifact or retraining an individual; they must be addressed through a more holistic perspective.

In fact, activities are not mediated by single tools so much as they are by ecologies of tools, including those tools I call genres. Just as individual genres develop to address activities, whole ecologies of genres develop as relatively stable configurations that jointly mediate activities. If we are to talk about the difficulties of computer users, then,
one way is to locate difficulties (a) between genres in these ecologies and (b) between the ecologies and the other elements of the activity.

These points are demonstrated in this chapter's study. The mediational means can be seen as artifacts that are the meeting of material tools and local rules. Artifacts have become familiar to users, so much so that developers interpret them as instantiations of genres. These genres of artifacts collectively mediate the developers' activities, and in doing so they become interconnected with each other in mediational relationships, relationships that I conceptualize as genre ecologies.

This study deals with three interrelated ecologies. These ecologies differ in a variety of ways, most notably in that some ecologies lack genres that are important to other ecologies. Consequently, different sites develop artifacts and genres that, although they are materially the same, are qualitatively different.

This study underlines the difficulties of design and evaluation: it implies that information systems designers at best can only contribute to the final artifact, not guarantee its success in any given activity, since usability is a quality distributed across the entire network. In this particular case, I have tried to understand some small part of the developers' lifeworlds, including the genres they regularly use to accomplish their activities. This understanding of the genre ecologies at the three sites can be applied to the design of new artifacts which draw on the most successful genres in the broader activity network of Schlumberger developers, as well as facilitate the most often performed actions within site-specific activity networks.

**Coda: Schlumberger's Response to the Study**

Ecological studies, which emphasize the cultural-historical and distributed nature of usability, provide guidance for developing future resources. Whereas traditional human-computer interaction studies and computer documentation studies might involve usability tests of individuals interacting with a single closely bounded artifact, this approach emphasized artifacts as (1) belonging to, perceived as, and used as genres, and
(2) interconnected in complex, developing ecologies that are compound mediators, i.e. artifacts working together in relatively stable formations.

As I mention above, this view implies that designers can only contribute to the final artifact and its success. What designers can do, however, is to design information systems to work within existing ecologies, that is, to fit ecological niches. Schlumberger's Usability Services for Engineering Research (USER) team has used this ecological approach to design three such information systems.

An automated, webbed data reference will soon reside on the corporation's intranet. USER recognizes that traditional printed documentation is often inadequate for developers' work because it becomes out of date more quickly than it can be written.

The automated reference will include complete references for all library datatypes and routines, formatted in a way similar to other references (e.g. manuals, online specifications) with which Schlumberger's developers are familiar. But unlike standard static references, this information system will be dynamic: it will be automatically recompiled from the library code each night, ensuring that the online reference will always be up to date. And the use of a familiar interface genre (the web browser) could ease the transition to this new form of documentation (McLellan, Roesler, Tempest, and Spinuzzi, 1998).

A set of complex, well-commented example programs will be distributed with the code. These programs will be more static than the webbed information system, but will provide a sort of detailed narrative illustrating the use of key datatypes and routines (McLellan, Roesler, Tempest, and Spinuzzi, 1998). These examples are an adaptation of the examples that are already used. But unlike the examples in the living code (which sometimes lead to inconsistent code sharing practices), this code will be designed to exhibit consistent and preferred code sharing practices. The features that make examples usable at these sites are preserved (e.g. the ability to coordinate the examples with formal definitions), but features that have been found to inhibit usability at these sites (e.g. inconsistencies in programming style) are muted or eliminated.
A software mining tool is in development (McLellan, Roesler, Fei, Chandran, and Spinuzzi, 1998). This tool draws on the success of genres such as automated searches and online internal documents, as well as software metrics tools and visualization tools. The software mining tool automatically scans a product’s entire baseline and creates a database containing quantitative and qualitative data about the code. Developers and project managers can then query the database to get a different view of the code: rather than looking at ground-level code or the one-line slices returned by grep queries, these users can produce a variety of reports and graphs detailing how various aspects of the code are used in the existing code. And these reports and graphs are rendered rapidly, meaning that users can form and test hypotheses about the larger system on the fly. The software mining tool combines contextualization (a feature of the example code) with text searches (a feature of grep and text editors). And it combines these with other features to provide levels of detail that were not available to developers before. Thus developers can be led to specific code without getting lost in the details.

Schlumberger is planning to continue ecological investigations of these three systems as they are being developed.

Implications for Methodology
In this chapter, I develop the concept of genre ecologies as compound and developmental. Through a study of software developers at work, I demonstrate some ways to analyze genre ecologies in cultural-historical terms and show some of the benefits of such an analysis. Although I used certain methods to investigate these ecologies — interviews, naturalistic observations, and genre analyses — I could have used a variety of other methods driven by the same methodological principles of compound mediation and ecological development.

Compound mediation. Genre ecologies jointly mediate activities; the usefulness of a particular genre in an activity primarily depends on how users perceive it and connect it with other genres in the ecology. Thus, I argue, the methodology I am constructing must take compound mediation into account. In this study I used naturalistic observations and
interviews to identify genres in the ecology and trace genres' interconnections. Other methods can also be brought to bear on this task, such as protocol analysis and ethnography.

**Ecological development.** Genre ecologies develop jointly. Changes in one genre can ripple through the entire ecology, affecting other genres. For instance, developers at Site 2 developed grep scripts, a genre that allowed them to automate code-searching and thus search code (a second genre) more easily. This innovation made it possible for them to avoid using other, more labor-intensive genres for searching code. In comparison, developers at sites 1 and 3 did not develop the genre of grep scripts, so they used other, more labor-intensive genres — text editors and database commands — to perform similar code searches.

The methodology I am constructing, then, must take the ecological development of genres into account. In this study I examined such developments through genre analysis. Other methods can also be brought to bear on this task, such as historical analysis, conversation and discourse analysis, protocol analysis, cooperative prototyping, and participatory design.

In this chapter I discussed and analyzed three interrelated genre ecologies, composed of on-screen genres (text editors, online help) as well as off-screen genres (such as manuals and a printed database script). This work leads us to the second research question: How do participants in an activity come to perceive and manage the various on- and off-screen tools used in an activity to mediate their work? I tackle this question in the next chapter. Along the way, I abandon the conventional notion of usability (as the quality of a single artifact) for a more complex tripartite classification used by activity theorists (Engeström, 1990, 1992; Nardi, 1996). And I study a genre ecology over time, using the study to think through the implications of the long-term development of genre ecologies.
CHAPTER 4
THE ALAS ACTIVITY NETWORK

As I argue in Chapter 3, the methodology I am constructing should afford a way to conceive of artifacts as members of an *ecology of genres* — an interrelated, changing group of genres that co-mediate their users' work in a shifting variety of ways. Yet Chapter 3 only brings us part way. It provides a static snapshot of three related genre ecologies, but it does not thoroughly explore the formation and development of these ecologies.

This present chapter aims to answer the question: How do participants in an activity come to perceive and manage the various on- and off-screen tools used in an activity to mediate their work?

The question is an important one for the methodology I am constructing. The methodology's purpose is to guide research into the design and evaluation of information systems. Yet one traditional way of conducting such research — usability testing, which tends to assume that usability is the property of a *particular artifact* — has been brought into question by Chapter 3, which locates usability as distributed across the *entire activity network*. This realization, if taken seriously, broadens the researcher's focus to an uncomfortable extent. Rather than studying individual artifacts, researchers must lend attention to a broad, changing array of artifacts being used in concert. Rather than studying individual users, researchers must lend attention to groups of users adapting, developing, and sharing interpretive strategies. Such concerns must be addressed by the methodology. This chapter takes up the challenge of developing a theoretical apparatus for performing such analyses.

In this chapter I explore the activity of accident location and analysis as it has developed in Iowa, and how it has shaped and been shaped by three generations of Accident Location and Analysis System (ALAS) software. Data in this chapter came from interviews, documents, and examinations of computer interfaces (see Chapter 2).
In this chapter, I argue that contradictions — tensions and imbalances — often appear among elements of an activity network. Activity networks are complex and only temporarily stable: elements of the activity network (e.g. actors, goals, communities, rules, divisions of labor) change continually, and not always in the same ways. Sometimes, as I show in this chapter, contradictions arise among the elements. Contradictions manifest themselves as discoordinations, difficulties in perceiving and managing genres in the ecology. Discoordinations in turn give rise to breakdowns, points at which users find that they must reinterpret the genres they are using.

To ease these contradictions, actors attempt to reorganize the activity network in various ways: by redividing the labor, adding more actors, adapting more genres to bring into the genre ecology, and fundamentally changing tools such as information systems. Of special interest is the point that a given information system is always supplemented by other genres. As users adapt other genres to supplement the information system, they import those genres into the ecology surrounding their work.

When contradictions become severe, software developers are brought in to develop a new information system or to revise an old one. Developers tend to attempt to purify the human-computer relationship (that is, to reduce users' need for off-screen genres) by reproducing genres within the interface (on-screen genres). Genres that have been adapted by users are often imported into the computer screen, where they are combined with established screen genres to produce hybrid genres. But as off-screen genres are imported into the interface, other off-screen genres proliferate.

Hybridized genres, the result of combining off-screen and on-screen genres, tend to retain their history, their addressivity, and their relationship to other genres in the ecology. They continue to be used more or less as their forebears were.

In the next section, I discuss how genre ecologies develop, change, and form contradictions. Then, in the remainder of the chapter, I describe how contradictions arise in a particular activity network over the course of three decades. I describe how genres have been imported into, and temporarily stabilized within, an ecology of genres used to
mediate this activity network. And I discuss how the ecology extends across both sides of the computer screen.

**Studying Genre Ecologies in Cultural-Historical Terms**

James Wertsch illustrates the development of mediated action by describing the changes in airplane design from the 1960s to the present.

In the 1960s, the design of a new airplane might have involved dozens of draftsmen working for months or years with slide rules, drafting equipment, and other such cultural tools. Today, the same task might be done in a much shorter time by a single computer operator using the complex hardware and software that makes computer imaging possible. The relevant issue to address in such cases is, "What happened?"

It is fairly obvious that an explanation of the increased productivity cannot be grounded solely in an account of increased intelligence or skill on the part of the individuals involved. Indeed, some might be tempted to argue that the single computer operator today needs less intelligence or skill than what was required of the engineers using slide rules, complex mathematical formulas, and other instruments several decades ago. What the illustration does suggest is that the intelligence involved is an attribute of the system created by the irreducible tension between agent and mediational means. (Wertsch 1998, p.35)

Many authors have argued, as Wertsch does, that cognition can be distributed among people and artifacts (e.g. Hutchins, 1995; Salomon, 1993). But what is intriguing about Wertsch's illustration is that it depicts an entire ecology of tools, including slide rules, drafting equipment, mathematical formulas, and so forth, being replaced by a computer. The ecology's functionality remains, but it is transferred somehow to the computer. How? Are slide rules, drafting tables, and mathematical formulas simply abandoned, replaced by an entirely different computer system? Are they swallowed whole by the computer, where they and their intricate ecology somehow reside — albeit in altered forms — within the interface? Wertsch does not elaborate. It seems to me that this is a highly interesting question for information system designers and evaluators: if genres are indeed imported into the computer interface (as I argue that they are), we may benefit greatly by understanding the genres' ecological relationships and how they have been
altered. One way to explore the issue is to conduct a study — not merely a snapshot of a
genre ecology at one point in time, but a cultural-historical study of a genre ecology as it
develops over a period of months or years.14

Such a study can provide a developmental view, one that can help us to fully
understand the operations that the current system requires of users. As Susanne Bødker
argues, "Artifacts crystallize knowledge so that operations which are developed in the
use of one generation of technology may later be incorporated into the artifact itself in
the next" (1997, p. 150). To put it in terms of Wertsch's illustration: the mathematical
formulas that the airplane designers used in the 1960s have become embedded in the
computer systems used by airplane designers in the 1990s.

But these computer systems have incorporated more than just mathematical
formulas. As Bødker argues, "one computer application may contain several
instruments," instruments that "are juxtaposed in their mediation of a particular activity"
(1997, p. 150). That is, the genres that jointly mediated airplane design in the 1960s
(mathematical formulas, slide rules, drafting equipment) are not simply replaced by a
computer system, they are to some extent reproduced, operationalized, and "juxtaposed"
in it. The 1960s-era genre ecology has migrated into the 1990s-era computer system,
albeit in drastically altered form.

**The Development of Genre Ecologies**

Such migrations do not happen en masse and for only one time; genre ecologies are
not entirely stable or static. An activity continually changes as elements of the activity
network change. Thus the ecology of genres mediating the activity must also change if
the activity is to continue.

One way to change a genre ecology is to introduce new genres to better mediate
activities (e.g. in the Chapter 3 study, software developers introduced online process
documents and grep scripts to help them better comprehend programming code). Yet in

---

14Such studies of computer systems are not unknown, but they tend to be limited in detail (Bødker, 1994;
Bødker and Gronbæk 1996; Yates and Orlikowski, 1994; Zuboff, 1988) or scope (Haas, 1996; Orlikowski
and Yates, 1994).
entering an ecology, those new genres are themselves changed. Bakhtin describes such a metamorphosis of genres:

Secondary (complex) speech genres — novels, dramas, all kinds of scientific research, major genres of commentary, and so forth — arise in more complex and comparatively highly developed and organized cultural communication (primarily written) that is artistic, scientific, sociopolitical, and so on. During the process of their formation, they absorb and digest various primary (simple) genres that have taken form in unmediated speech communion. These primary genres are altered and assume a special character when they enter into complex ones. (1986, p. 62)

Similarly, Charles Bazerman describes genres being "drawn into" documents:

In constructing the text, the writer makes visible for the readers some components that went into it, represented in generically appropriate ways and put in relation to other visible elements. Tables, charts, descriptions, and references to other documents are more obvious modes of drawing outside artifacts into documents, but more subtle are the passing mention of a government office or piece of legislation, the name of a form, or a list of addressees. (1994, p. 20)

Bakhtin and Bazerman see genres as entering a larger genre, yet retaining something of their own history, addressivity, and distinctiveness, as well as their interrelations with other genres. When we encounter the genre of the personal letter in Dostoyevsky, or the genre of the table in an annual report, we expect them to be quite similar to the same genres outside the genre ecology — although perhaps metamorphosed to fit a niche in the ecology (e.g. tables in reports have table numbers).

Yet neither Bakhtin nor Bazerman deals with how genres enter and are changed by the genre ecology of the computer interface. If a table genre were to enter a computer interface, it would have to be displayed using some sort of interface genre — as a graphic in a window or as a collection of spreadsheet cells, for instance. The genre that migrates into the ecology of an interface must be combined with (i.e. become represented or recast in) an existing interface genre, and their union produces what I term a hybrid genre: a genre that involves the history, addressivity, and distinctiveness of its parents; that retains the interrelationships with other genres that its parents enjoyed; that users
perceive as being more or less the "same" genre as its parents, and can apply habits they have developed for dealing with its parents.

I discuss such hybrid genres throughout this chapter, and in more detail in chapters 5 and 6. In the latter two chapters I discuss how hybrid genres relate to breakdowns and discoordinations; in this chapter I focus on how they relate to contradictions.

**Contradictions, Discoordinations, and Breakdowns**

All is not always harmonious in a genre ecology. Genre ecologies mediate activity networks, which are constantly destabilized as they are interpenetrated by other activities (Engeström, 1992) and as their elements change. As Marcie Tyre and Wanda Orlikowski describe it:

> the introduction of a new technology into an operating environment triggers an initial burst of adaptive activity, as users explore the new technology and resolve unexpected problems. However, this activity is often short-lived, with effort and attention declining dramatically after the first few months of use. In effect, the technology, as well as the habits and assumptions surrounding it, tends to become taken for granted and built into standard operating procedures. This initiates a period of stability in which users focus attention more on regular production tasks than on further adaptation. Later on, users often refocus their attention on unresolved problems or new challenges, creating additional spurts of adaptive activity. In many cases, this episodic pattern continues over time, with brief periods of adaptation followed by longer periods of relatively stable use. (1994, pp. 12-13)

Although Tyre and Orlikowski focus on how technological change leads to this "lumpy" or episodic pattern of adaptation, other changes — changes in organizations or goals, for instance — can lead to destabilizations as well. Such destabilizations, or *contradictions*, are the impetus behind changes in the genre ecology — changes that involve the adaptation, migration, and hybridization of genres. Contradictions manifest themselves as *discoordinations*, difficulties in perceiving and managing genres in the ecology. Contradictions in turn give rise to *breakdowns*, points at which users find that they must reinterpret the genres they are using. To understand changes in the genre ecology, then, we need to understand contradictions, discoordinations, and breakdowns.
Contradictions

"Between the components of the [activity] system," Engeström tells us, "there are continuous transformations. The activity system incessantly reconstructs itself" (1992, p. 12). This continual reconstruction is fueled by contradictions, which are tensions or imbalances in the activity network. As contradictions occur at the level of activity, they call for a macro-level analysis of the activity network and its components: actors, tools, objects, outcomes, rules, communities, and divisions of labor.

Engeström discusses two types of contradictions. Primary contradictions are fundamental tensions in the socioeconomic system; for instance, in capitalism, Engeström points to "the dual nature of commodities, the tension between the use value and the exchange value" (1990, p. 84). Primary contradictions reside in each component of the activity network: for instance, Engeström points out, doctors in a capitalist system are simultaneously expected to be (a) "gatekeepers and cost-efficient producers" and (b) "healers or consultants." The same tensions reside in the institutional rules that govern these doctors: health maintenance organizations pressure doctors in their employ to dispense with "unnecessary" tests, reflecting the tension between the institution's economic interests and the patient's health needs.

Primary contradictions are beyond the scope of this study. Instead, I concentrate on secondary contradictions, that is, tensions between components of the activity network. Engeström explains that "when a strong novel factor [is] 'injected' into one of the components [of the activity network] and it thus acquires a novel quality, pressing secondary contradictions appear between that component and some other components of the system" (1990, p. 85). For instance, when a new information system is introduced into an existing activity network, it is likely that a secondary contradiction will appear between it and the established rules that have until now governed the activity. In Engeström and Escalante's (1996) study of an automated kiosk, for example, the kiosk's users were able to use it to mediate a variety of activities, but to do so they had to learn new rules for operating the kiosk.
Of secondary contradictions, Engeström says:

These secondary contradictions of the activity are the moving force behind disturbances and innovations, and eventually behind the change and development of the system. They cannot be eliminated or fixed with simple remedies. They get aggravated over time and eventually tend to lead to an overall crisis of the activity system. (Engeström 1990, p. 84)

In the study later in this chapter, I discuss the history of the ALAS activity network, giving special attention to the secondary contradictions that arise and the genres that are imported into the genre ecology in attempts to ease those contradictions. I also point to how these contradictions lead to discoordinations and breakdowns (discussed in more detail in chapters 5 and 6).

Discoordinations

Discoordinations are difficulties in interpreting artifacts and managing the actions that those artifacts mediate. Since discoordinations occur at the level of action, they call for a meso-level analysis, that is, an analysis that connects the activity-level contradictions with the moment-by-moment breakdowns that occur at the level of operation.

Engeström contrasts discoordinations with coordinations, in which the normal, expected flow of interaction continues without a hitch. He describes coordination in two ways, which I attempt to unpack below in terms of genre perception and genre management.

Genre Perception. Genre perception can be conceived in Engeström's terms as the matching of "symbols" — that is, the understanding of an artifact in terms of a genre, and the application of genre habits (rules) to it. Engeström's comments on discoordinations in discourse, although having to do with spoken communication, are applicable to this sort of discoordination: "When both symbols are identical [e.g. when user and designer perceive the artifact as being of the same genre], we may say that discourse ... is coordinated" (Engeström, 1990, p. 87). In the terms of Chapter 1, the designer and user agree on the artifact's semiosis; they agree on its genre, and they understand its genre in a
similar way. On the other hand, if the user perceives the artifact as being of a different genre, the result is a discoordination: the user may attempt to use the artifact with the rules of the perceived genre, only to find that the artifact reacts unexpectedly. At that point the user may encounter a breakdown, a point at which she or he must consciously reinterpret the artifact in terms of another genre.

**Genre Management.** Genre management involves coordinating the interrelated genres in an ecology in such a way as to co-mediate the activity at hand. Users manage genres through genre rules applied within the ecology. For instance, in Chapter 3’s study, a software developer knows that programming code includes comments which can be used to interpret the code. So when the developer applies the genre rules of comments to a particular comment, the action only makes sense because she uses the result to interpret the surrounding code or to write new code. Thus the genres of the comment and the computer language are interconnected in a mediatory relationship.

When users successfully manage genres, they experience "the normal, scripted flow of interaction ... the script, coded in written rules and plans or tacitly assumed traditions [here, genre rules], coordinates [users'] actions as if from behind their backs, without being questioned or discussed" (Engeström, 1992, p. 66). Developers go about their work, interpreting and producing code. But often developers do not experience "the normal, scripted flow of interaction." Sometimes they have trouble managing genres: they attempt to apply genre rules, but the genres interact in unexpected ways. The genres are discoordinated. And the discoordination gives rise to a breakdown: rather than using the entire ecology to co-mediate their activity, users find themselves attempting to reinterpret the individual genres of the ecology.

**Breakdowns**

Points at which a user finds the present interpretation of an artifact to be inadequate for the task at hand are called *breakdowns*. Breakdowns cause the user to become aware of an action he has operationalized. Since breakdowns occur at the level of the operation,
they require a micro-level analysis of the user's moment-by-moment interactions with the system.

As I discuss above, breakdowns result from dis coordinations (Engeström, 1990, 1992). Breakdowns can lead the user to re-perceive and re-manage genres. For instance, imagine that a software developer attempts to use a code comment to understand the program he is examining. Although the developer does not realize it, the comment is out of date. Although the comment appears to describe the code, like others the developer has read, when he tries to use it to interpret the code he finds that it refers to code features that do not exist. The developer's perception of the comment is inadequate for supporting his work; he must reinterpret it if he is to continue to use it in his activities. For instance, he may conclude that the comment is out of date; he'll have to discuss it with the person who last worked on the program.

Breakdowns are relatively easy to locate because users are aware of them and typically call them mistakes. Designers can think of breakdowns as symptoms that indicate deeper dis coordinations and the contradictions that underlie those dis coordinations.

Of the three, contradictions are the deepest and most difficult to diagnose. Yet if researchers are to account for design flaws rather than their symptoms, the methodology must provide a way to diagnose these contradictions. In the remainder of this chapter I discuss the history of the ALAS activity network, giving special attention to the contradictions that arise and the genres that are imported into the genre ecology in attempts to ease those contradictions. Chapters 5 and 6 continue the work by studying dis coordinations and breakdowns at two points in the activity network's history.

**Overview of the ALAS Activity Network**

Like many states, Iowa keeps rather detailed statistics on the traffic accidents that occur on its roads. In fact, Iowa has one of the most extensive systems in the nation. The Accident Location and Analysis System (ALAS) contains data on crashes occurring on every interstate highway, every state road, every street, and even every gravel road in the
state. These data are the object of what I will term the ALAS activity network: the statewide activity network devoted to locating and analyzing accidents (Figure 4.1).

At this level of granularity, the subjects are governmental agencies that use various mediational means to transform data — for instance, to transform a sheaf of officers' accident reports into a set of statistics suitable for making decisions.

But, of course, this is not the whole story. If we increase the granularity — scale down the level of analysis to the activities of specific agencies — we begin to see that this activity network is the interpenetration of several activity networks distributed in time, space, and culture (Engeström 1992; Russell 1997; Bødker 1997; Bakhtin 1981; see also Chapter 5). For instance, individual agencies transform data in vastly different ways and for different purposes as they labor in their discrete activities. County and city
engineers use the transformed data to help them decide whether to regrade roads, raise bridges, and erect stoplights and signs, as well as to answer citizens who want to see those changes made. Police officers use the data to affect the rate of traffic offenses by deciding where to place officers or what sorts of violations to crack down on. Emergency response teams use the data to plan routes to and from high-risk areas. The Governor's Traffic Safety Board uses the data to monitor behavioral factors (such as alcohol use) in traffic accidents statewide. Legislators at the city, county, and state levels use the data to decide where they stand on issues such as road improvement and driving laws. And most of these entities use the data when applying for yearly grants from the state and federal governments. These entities all need what Johnson (1997) terms a "magic lens," a tool that enables them to browse through the copious data and select only the data they need to examine.

Just as looking at the different agencies synchronically complicates the picture, so does looking diachronically at different eras in the development of the ALAS activity network. Changes in the gathering, processing, and distributing of data have periodically rippled across the network, transforming the overall network itself as well as the activity networks of the individual agencies. These transformations include the data that the agencies seek, the ways they work, the mediational means they use, and even their fundamental divisions of labor.

Below, I discuss four eras in the ALAS activity network. These eras are defined by the introductions of ALAS products. At the end of each era, internal contradictions in the activity network made a new ALAS product possible, even necessary; at the inception of each new ALAS product, these contradictions were eased, the network was fundamentally changed, and new contradictions began to form. As they developed new ALAS products, software developers mingled the genres of the previous era with interface genres to produce hybrid genres suitable for mediating the reorganized activity network.15

15 The process of genre hybridization noted in this study is similar to Bakhtin's notion of genres becoming "partially digested" by the genre of the novel (1986).
Before 1974: Pre-Automation Accident Location and Analysis

The Ecology of Genres in the Pre-Automation Era

Before the early 1970s, the activity of accident location and analysis in Iowa was rudimentary — although even in this rudimentary state, the activity network involved a complex ecology of genres, including printed forms, various types of maps, summary sheets, a variety of reports and data displays, and eventually punch cards, magnetic tapes, and printouts. The ecology of genres was as widely dispersed as the division of labor in the activity network, as I describe below (Figure 4.2).

Note that in Figure 4.2, the outcome is not simply "output." Output genres are important as an aspect of the transformed data, but the data are transformed for a larger purpose: to provide effective solutions to vexing problems.

Before 1974, accidents were recorded — as they still are today — on two types of reports. If an accident involved fatalities, injuries, or damages over $100\textsuperscript{16}, drivers involved in the accident each had to file a driver's report (Figure 4.3), which is sent to the drivers' insurance companies and then to law enforcement. In addition, for accidents involving fatalities or injuries, police had to fill out officers' reports, which were more detailed versions of the driver's reports. Both types of reports included information such as location, conditions, number of vehicles involved, type of accident, and a diagram and narrative explaining the accident. Locational parameters were expressed in terms of address, intersection, or landmarks near which the accident took place (Wilbur Smith and Associates, 1972, p. 18-19). If the location was uninterpretable, the Iowa Department of Transportation (DOT) sent a map to the reporting party, who marked the proper location with an X and sent it back to the DOT.

The reports were stored at the local level: in the county sheriff's office or the city's police department. Copies of the reports were forwarded to the Iowa DOT in Ames. This

\textsuperscript{16}Later, the damage threshold went up to $400, then $500. In 1998, it was changed to $1000.
is essentially the same system of collecting data as exists today, although the system promises to change radically in the near future (as I explain later in this chapter).

For various reasons, agencies needed statistics on various types of accidents. For instance, summaries of statistics on deadly accidents were used by law enforcement agents, city and county engineers, and legislators, among others. These statistics were compiled and distributed by hand. They were obtained in two ways: at the local level and at the state level.

At the local level, officials would individually leaf through the reports, counting accidents on the rural secondary and municipal street systems that met certain criteria —
Figure 4.3  A completed driver's report (personal information blacked out). Notice the specialized diagram and the narrative: "Vehicle 1 was stopped behind 2 other vehicles at a red light. Vehicle 2 attempted to make left hand turn, lost control and hit left side of Vehicle 1." Officer's reports are more detailed.
e.g. how many accidents in the county involved alcohol in 1961, or how many deadly accidents occurred at a given intersection. This technique, which is still used by some local agencies, provided officials with rather crude statistics that were unsuitable for complex analyses. Yet many local jurisdictions did not even perform this level of analysis, and others performed it poorly because of inexperience, lack of training, or lack of detailed accident information (Wilbur Smith and Associates, 1972).

Local agencies visualized accident distribution by adapting a familiar genre, the map. According to a study conducted in 1971, 62% of cities and 49% of counties surveyed reported that they maintained spot maps or point maps\(^\text{17}\) to summarize accidents by location (Wilbur Smith and Associates, 1972).

At the state level, officials at the Driver's License Division of the Department of Public Safety (DPS) would match all reports covering the same accident and assign a number to the accident. In the early 1970s, forms with handwritten location codes were added to facilitate information storage for automated data processing (Wilbur Smith and Associates, 1972, p. 19-20). When analyzing accidents, personnel at the DOT would leaf through their copies of the reports, counting accidents across the entire state according to a wide range of criteria: fatal accidents involving pedestrians, trains, farming or construction equipment, and so forth. These data would be summarized and analyzed in annual reports from the DOT and other related agencies. The reports were then forwarded to various state agencies. By 1971, nine statistical summaries were regularly compiled by the Data Processing Division of the DPS (Wilbur Smith and Associates, 1972, p. 21-22). The reports included prose analyses, tables, bar graphs, pie charts, and a number of humorous cartoons depicting various types of fatal accidents (Figure 4.4).

Within each county, the DPS would identify segments of rural roads by two-character codes and indicate the segments ("control sections") on special county maps. Three clerks organized the reports by segment and attached special forms to them to indicate segment information.

\(^{17}\)Actors would mark accidents on these maps with pins or pens.
By the early 1970s, as part of the TRACIS initiative towards automation, the DPS was entering its accident data into its mainframe system via punch cards. The data were then stored on spools of magnetic tape and copies of these tapes were sent to the Iowa State Highway Commission. ISHC personnel generated printouts for the primary road system by category, then manually noted accidents on spot maps and Spot Location Accident Summary Sheets. The spot maps allowed users to identify concentrations of accidents, which could indicate trouble spots that could be mitigated through engineering (e.g. regrading a road, placing signage) or policy changes (e.g. changing the speed limit, focusing law enforcement efforts on particular areas).

Statistics provided by the DOT were more complex than those produced at the local level. But they were still crude by today's standards.

At both the local and state levels, the activity of accident location and analysis involved a transformation\(^{18}\) of data: from the actual event to the officer's or driver's reports, and again to a set of simple counts and categorizations. As has been discussed by others (Lave, 1988; Hutchins, 1995), such transformations are sometimes necessary for data to be used with existing tools. Statistical tools are not typically useful for parsing a narrative, but if the narrative can be transformed into a set of numbers and categories, statistical tools can be very useful indeed. In this case, the quantification of reports turned thousands of narrative and diagrammatic descriptions into visual data display genres such as pie charts, bar graphs, and tables, each of which could have more impact on public policy than individual narratives could.

As Figure 4.2 illustrates, actors at both levels used conceptual rules (e.g. report genres, localized rules for categorizing and counting elements within the narratives) and conceptual tools (e.g. mathematical and statistical formulae) to transform narratives into data reports. But these quantification tools and genres took quite a bit of work.

Quantification involved distilling copious narratives into minimal numbers, and thus

\(^{18}\)That is to say, one genre is translated into another. These transformations make it possible to analyze accident data quantitatively, but at the price of losing or compressing the individual narratives. In this case, the heteroglossic reports are monologized as they are quantified for the database. I do not go into the repercussions of such monologizing transformations here, but this point could warrant further study.
analysts had to examine file cabinets full of reports to get handfuls of numbers.
Narratives had to be individually parsed and transformed into quantitative representations; those representations had to be transformed by hand through statistical or mathematical calculations; and the results had to be transformed, again by hand, into reports couched in a particular genre. At every step of the process, users had to decide
what to include in the transformation (e.g. the sum of accidents occurring on rural roads) and what not to include (e.g. the sum of accidents at a particular intersection at a particular time of day).

Once these transformations were completed, however, users found the data far more versatile and began to see accidents in different ways. Users became like Eadweard Muybridge, whose photographic studies in the 1880s managed to dispel misconceptions about what everybody had always seen (how horses gallop, how people move) because he had subdivided the subjects' movements into a precise and lengthy enough sequence of shots ... (Sontag 1990)

That is, by viewing accident data in a new, systematized way, users found that they could use the transformed data to label dangerous stretches of road, dangerous behaviors, and dangerous classes of drivers. These labels became tremendously important for obtaining grant money, planning roads, establishing law enforcement strategies, and the like — and therefore transformed data became increasingly valued within the ALAS activity network.

Contradictions within the Pre-Automation Activity Network

But as systematized accident data became more widely used, users came to want and expect more from them. It's all very well to know general statistics about accidents at rural intersections throughout the state, but is a particular intersection more likely to have accidents than other intersections? What kinds of accidents? Under what conditions? Such questions could not be routinely answered under the labor-intensive pre-automation system. The need grew for more data — that is, the object of the activity system became more complex and the outcome more expansive (Figure 4.5) — and two contradictions also grew:

(A) The mediational means (counting by hand) was too labor-intensive for actors to make use of it often.

(B) Therefore, actors could not transform the object to a sufficient level of complexity, either at the local or the state level.
Engeström reminds us that such contradictions "are the moving force behind disturbances and innovations" (1990, p. 84). These contradictions began to cause problems that were eventually addressed through a radical reorganization of the activity network and its ecology of genres, a reorganization centering on automation.

The changes that rippled across the ALAS activity network at this point were attempts to ease the contradictions. We've seen how users adapted statistical and mathematical genres, report genres, and forms with location codes as they tried to ease the contradictions. But the expanding genre ecology provided more possibilities for
discoordinations among genres. Finally, the Iowa State Highway Commission hired a contractor to automate accident location and analysis.

1974: Mainframe-ALAS (IBM 3090 mainframe)

The Ecology of Genres in the Mainframe-ALAS Era

In the early 1970s, automation moved into the ALAS activity network and fundamentally changed it. Iowa's accident location and analysis became more centralized, its data more complex, and its ecology of genres more diversified (Figure 4.6). Automation eased contradictions, not by clearing out the existing genre ecology, but rather by importing some genres into the interface while retaining their history, addressivity, and relationships to other genres in the ecology. Genres such as spot maps and reports were hybridized with automation-era genres such as quantified representations of roadways and computer printouts to form a new genre ecology that existed partly within the computer.

This transformation was helped considerably by two developments: the node-link system and the data system that became known as mainframe-ALAS. Both were inventions of Wilbur Smith and Associates (WSA), a firm that was commissioned by the ISHC in 1971 to develop a system for accurately identifying accident locations on all primary, secondary, and municipal roadways in Iowa. WSA's recommendation report, submitted in June of 1972, recommended the node-link system for representing accident locations and a data system for manipulating the data.

To automate accident locations, WSA had to quantify the roadway system — turn it into a form that could be easily stored in and manipulated by a computer system. In addition, the system had to work with the existing data collection tools: the drivers' and officers' reports, which located accidents by street address, intersection, or distance from landmarks. Quantification, unfortunately, was a two-way street:

Ideally, accident locations would be reported in a format directly suited to automated data processing. However, a system which is readily usable to a computer may tend to be too abstract for the average motorist to
comprehend easily. This is particularly true when a series of special maps is the only source of the necessary identification numbers or coordinates.

However, even if field markers [such as mile point markers] were utilized, it must be admitted that it would seem more natural to identify an accident location by methods with which the motorist is already familiar, such as street name or route number. The most workable procedure, then, may entail conversion, by state coding personnel, from identifiers used on the accident report to a uniform numerical system suitable for computer processing. (Wilbur Smith and Associates, 1972, p. 35-36)

Indeed, this is the approach that WSA adopted for quantifying the roadway system. And the consequent effect on the nature of the data was the most striking change in the activity network. Whereas data from the officers' and drivers' reports had undergone limited quantification before — mainly involving the categorization and quantification

Figure 4.6  The mainframe-ALAS activity network. Straight arrows indicate the easing of contradictions from the previous era.
of accident factors — now the entire roadway system had to be quantified in order for mainframe-ALAS to process the data. WSA assigned each county and city a two-digit designation number, then mapped all counties with a system of nodes: coordinates marking all intersections, ramp terminals, railroad crossings, grade separation structures, bridges, road ends, 90-degree turns, county lines, major signalized commercial entrances, and interchanges and other multiple node intersections.

Now, crashes had to be located in relation to the nearest nodes. Crashes that occurred at a node (e.g. within an intersection) were referenced by the node itself. Crashes that occurred away from a node (e.g. 20 feet away from the intersection) were referenced in a link between nodes (e.g. recorded as 20 feet away from the intersection's node, in the direction of a second node). The node-link system, then, was a radical simplification of the accident's location performed to quantify it for further data analysis. And as the roadway system expanded, more nodes had to be identified, numbered, and charted. In 1972, there were about 180,000 nodes in Iowa; in 1998 the number was approximately 240,000.

The node-link system was conceived as a centralized system, tightly controlled at the state level:

This system should be used directly only at the state level, by accident-coding personnel. For this purpose, master maps would be printed in limited quantities, containing all node numbers. (Wilbur Smith and Associates, 1972, p. 41)

These "master maps" were the result of overlaying an old, familiar genre — the map — with the new node system. The result was a new hybrid genre: the node map (Figure 4.7).

Another new genre was the node table, a text-only list of nodes in each county along with a short literal description of each. Node tables were printed from the DOT's node database and bound in books. The books were then used by operators to identify nodes and (more often) strings of nodes. For instance, a user wanting to gather data on a stretch
Figure 4.7 A portion of an early node map of four townships in Story County. The diagonally angled numbers indicate nodes. These numbers are hard to read when densely clustered, and that problem was only exacerbated when more roads were built.

of Interstate 35 would look up the nodes in a county whose descriptions included the identifier "I-35."

The ALAS activity network required an automated accident location system that could more or less reproduce pre-ALAS work (locating, counting, and reporting
accidents) using the alphanumeric data of the node-link system. WSA recommended, and partially implemented, a cutting-edge system that became known as mainframe-ALAS, consisting of 32 programs and three permanent tape files (Wilbur Smith and Associates, 1974b). The programs were written in the programming language COBOL, but designers envisioned add-ons written in PL/I or FORTRAN. Mainframe-ALAS was seen as only the basis for a more sophisticated system. However, WSA was not retained for the second and third phases of development, so the system was not further developed.

Mainframe-ALAS could not take requests or updates in the same way that human operatives could (e.g. through handwritten reports, letters, and telephone calls), so requests and updates were input into mainframe-ALAS via punch cards. Designers adapted a variety of genres to translate users' oral and written requests to machine-readable format: a worksheet for generalized requests (Figure 4.8) as well as a variety of forms that guided users as they filled out punch cards for generalized requests, node string updates and requests, accident summaries, and so forth. These forms, along with the node maps, were the interface for the users: genres that addressed their frequent queries in terms of the node-link system, and in doing so standardized those queries.

Mainframe-ALAS' output drew on older, pre-ALAS report genres. Output was via printouts in one of two report formats. The High Accident Ranking Report, a "packaged" summary report (Wilbur Smith and Associates, 1974a) with a rigid format, ranked a group of locations by the number of accidents at each location. The difficult-to-read Generalized Request Report, a one-time request based on certain parameters, gave detailed information for each accident that matched the parameters. Although both reports were useful, the Generalized Request Report, with its ability to pinpoint specific locations and accident types, was to become the cornerstone of further ALAS.

19 "Users" were considered to be those making the requests, not those who actually operated the machine (Wilbur Smith and Associates, 1974a).
Figure 4.8 A worksheet for generalized requests. The Generalized Request Form helped users recast queries in terms of the node-link system.

development. Both reports were difficult to read, requiring the DOT to train special analysts to interpret them and thus widening the division of labor (as I discuss below).

These report genres were hybrids: combinations of older hand-compiled report genres, officers' and drivers' report forms, and standard computer printouts. The Generalized Request Report in particular resembled the officers' and drivers' reports: the location descriptions and the very narratives written on the reports appeared in the Generalized Request Report, albeit transformed to fit neatly into the ALAS database.

The older input and output genres (driver's and officer's reports, DPS reports) had migrated into the interface of the new system, mingled with the system's interface genres, and produced a hybrid genre that could be used to mediate new activities, just as the map genre and the node-link system had produced the hybrid node map.

Both the node-link system and mainframe-ALAS changed the activity network. Of particular interest is the increasingly complex division of labor. First, locators were hired
to transform the categorical and narrative data of the officer's and driver's reports into node data. Locators were DOT personnel whose sole job was to read the reports, pinpoint the accident location on a map, and figure out the node number(s) that best expressed the location. The results were passed on to data entry personnel, who would transform the reports' narrative accounts into database records (and occasionally reconcile conflicting narratives). Finally, analysts would fill out requests on preprinted forms, key them, and run them in batches. The resulting printouts were difficult to read, so specially trained analysts were assigned the job of interpreting them. Thus the task of accident analysis, once a relatively solitary affair, became piecework: different specialists had to work together to produce the analyses. And whereas accident analysis once involved fairly traditional tasks of writing and reading, in the mainframe-ALAS era these traditional tasks were disassembled and divided among workers: one translates the original reports into machine-readable code, one does the same for written or spoken requests, and a third transforms the machine-compiled results into more traditional written and spoken genres. See Figure 4.9.

But if the division of labor was more complex, so were the analyses that were possible from this new system. Access to complex accident data was quicker, easier, and cheaper than it had ever been before. Users could more easily access statewide accident records. Analyses were more feasible and results were more consistent. The centralized database reduced errors in accident reports and counts. The DOT produced node maps for all cities and counties to help local officials better understand reported data. Report formats became more standardized, and the number of reports and applications of accident data analysis increased (Pawlovich, 1996). By 1990 the DOT was regularly sending customized ALAS reports to all 99 counties (including data such as accident statistics on each intersection in the county), as well as filling specific requests from county engineering offices.
Contradictions within the Mainframe-ALAS Activity Network

Although mainframe-ALAS achieved a greater level of productivity and complexity than ever before, it formed contradictions as well, as illustrated in Figure 4.9.

(A) Data and analysis tools were deliberately centralized at the DOT, meaning that other agencies had to telephone or mail their query requests to the DOT, then wait days or weeks to have the resulting reports mailed back to them.

(B) The slow response time meant that it was impractical to make multiple subsequent queries: users could not easily test increasingly specific hypotheses or "snoop around" the data.

(C) Operating mainframe-ALAS was no picnic. There were problems with formulating and processing requests, due partly to the rigid nature of the query language. One character in the wrong column on a punch card could get the query rejected. And since queries were processed in batches, the unlucky operator would have to wait hours to discover errors. Output was in the form of two types of reports: one had an inflexible
format and the other had to be interpreted with reference sheets and tabulated by hand. Finally, mainframe-ALAS took weeks to learn how to operate (Moreland, 1992a).

Consequently, mainframe-ALAS was never used as widely as one might expect. It was used almost exclusively by the DOT and by some county engineers' offices. Its reports, in fact, were designed primarily for engineers (Iowa Department of Transportation, 1994). And other users, such as law enforcement officials and legislators, typically needed the data more quickly than the DOT could run the requests.

We've seen how WSA adapted a wide variety of genres, mingling some of them with interface genres to produce hybrids, in a quest to ease the pre-automation contradictions. Yet new contradictions formed (Figure 4.9). Despite these contradictions, mainframe-ALAS endured from 1974 until the late 1980s.

1989: PC-ALAS (DOS)

The Ecology of Genres in the PC-ALAS Era

The contradictions that bedeviled mainframe-ALAS led to the development of a new program, PC-ALAS, which again transformed the ALAS activity network in significant and far-reaching ways. Perhaps the most significant change was that the genre ecology expanded as mainframe-ALAS genres mingled with PC-based interface genres — menus, dialog boxes, message windows, and text windows — to form new hybrids. Automated report genres multiplied. At the same time, uses adapted other genres such as handwritten notes and manuals. (Chapter 5 discusses the PC-ALAS interface in more detail and provides an empirical evaluation based on participants' use.)

In 1989 Joyce Emery, who was running the DOT's Traffic Safety Office, hired Scott Moreland as an intern. Moreland, an 18-year-old computer engineering student at Iowa State University, was to serve as a data analyst, running queries and interpreting reports. At this point, mainframe-ALAS' shortcomings were becoming increasingly obvious to Emery and she was interested in a PC-based solution. Moreland's solution, which
eventually won him the Iowa Governor's State Top Achievement Recognition (STAR) Award, was PC-ALAS.

Moreland's idea was to create a PC-based ALAS that had the same capabilities as mainframe-ALAS: the ability to read the existing state crash data, run various queries, and generate reports. Whereas mainframe-ALAS had cost millions of dollars to develop and run, PC-ALAS was developed in Moreland's spare time and could be distributed freely to any and all who were interested.

Moreland developed PC-ALAS using Borland's version of the Pascal programming language, Turbo Pascal 5.0. The original PC-ALAS ran on MS-DOS 3.3 and was pilot-tested in late 1989 and 1990; other versions followed. In 1990 Moreland implemented a more user-friendly interface using Borland's TurboVision package. The new interface, although DOS-based, included Windows-like features such as pull-down menus, dialog boxes, and mouse support (Figure 4.10).

Moreland and others who later maintained PC-ALAS (John Barr and Chad Claire) also implemented a variety of other options, including canned queries and a report specifically for law enforcement agencies.

PC-ALAS was a revolution of sorts, a low-budget, high-tech assault on the mainframe-centered mentality that dominated the DOT at the time. In an article reporting Moreland's winning of the STAR award, one DOT publication enthusiastically hailed PC-ALAS as good news for democratic data processing:

Dyed-in-the-wool data processing people still say it's a mainframe world.
For the rest of us, namely the computer-illiterate population, personal computers (PC) may be our pathway into the electronic era. (Iowa Department of Transportation, 1990)

The mainframe-ALAS legacy was centralization. Under mainframe-ALAS, accidents were located, data were entered into the system, requests were received and processed, and results were issued and interpreted — all at the Iowa DOT. But PC-ALAS meant a certain degree of decentralization for many users: although data were still coded and consolidated at the DOT, the datafiles were then released — along with a copy of
Figure 4.10 A PC-ALAS screen, showing the Windows-like DOS interface.
Windows-like features include the horizontal menu; the dialog box with its checkboxes, radio buttons, and OK and Cancel buttons; and the status line at the bottom of the screen.

PC-ALAS — to all interested agencies. They were distributed through mail, the DOT electronic bulletin board system, and the DOT circuit rider (an individual commissioned to visit local agencies and disperse DOT materials). A single floppy disk could hold both the program and a county's datafile. As local agencies began processing their own requests, the burden on the DOT was initially lightened. Yet the ability of local agencies to process their own requests led to more complex analyses; it raised the bar on what could be considered a good analysis. Soon the DOT was awash in requests, this time from smaller agencies that had not felt the need for complex analyses before.
Figure 4.11 PC-ALAS activity network. Queries now go through local agencies.

Once again, the ALAS activity network was fundamentally changed. As Figure 4.11 indicates, PC-ALAS genres were still genetically connected to mainframe-ALAS. The node-link system was still used, and in fact became far more widely known and used than ever before. Local agencies began to see the road system in its terms. And the genres linking pre-automation and automation-era representations of the road system — node maps and node tables — were no longer centralized, but rather were widely dispersed and widely used by local agencies.

Furthermore, the genre of data entry forms used for querying mainframe-ALAS survived after a fashion: the forms migrated into the PC-ALAS interface, mingling with the genre of the dialog box to produce hybrids. For instance, the mainframe-ALAS Generalized Request Worksheet and Form both had selection types; these types are incorporated into PC-ALAS' Run Location dialog box. Similarly, the mainframe-ALAS Node String Request Form closely resembles PC-ALAS' String Request Parameters dialog box (Figure 4.12).
Figure 4.12 The mainframe-ALAS Node String Request form (top) mingles with the screen genre of the dialog box to produce a hybrid, the PC-ALAS String Request Parameters dialog box (bottom). Some features, such as the county specification and the columnar format for node entry, persist. Other features are distributed among other interface genres or are not handled by PC-ALAS (e.g. updating).
Table 4.1  Groupings of data and how they are acquired through the PC-ALAS menu system.

<table>
<thead>
<tr>
<th>Grouping</th>
<th>Description</th>
<th>Acquired through</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universe of data</td>
<td>All data stored in the PC-ALAS database</td>
<td></td>
</tr>
<tr>
<td>1. Requested data</td>
<td>Subset of the universe of data pertaining to a particular county, city, intersection, or other area</td>
<td>Request menu</td>
</tr>
<tr>
<td>2. Searched data</td>
<td>Subset of the base data meeting criteria such as accident conditions, vehicle characteristics, or driver characteristics</td>
<td>Search menu</td>
</tr>
<tr>
<td>(Optional)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Reported data</td>
<td>All found data, represented in a chosen report genre</td>
<td>Report menu</td>
</tr>
<tr>
<td></td>
<td>(If the user did not search for data, all base data are represented)</td>
<td></td>
</tr>
</tbody>
</table>

And this is in fact a rather important point. Recall that the mainframe-ALAS forms themselves were genres that addressed frequent queries in terms of the node-link system. These queries had migrated into the interface — first as paper mainframe-ALAS forms, then as PC-ALAS dialog boxes. It's important to understand that the PC-ALAS dialog boxes aren't simply metaphors for the "real" forms. Rather, they are qualitative transformations of the older genres, used in much the same way, designed with similar genre rules, and mediating the same activities. The older genres have mingled with interface genres to produce hybrid genres suitable for the new ecology: forms become dialog boxes, reports become scrollable text windows, and routine procedures are fixed within menus, just as routine queries were fixed within ALAS forms in 1974.

The last part bears closer examination. The PC-ALAS menu structure separates ALAS actions into requesting, searching, and reporting information, and in doing so, provides a way for users to conceive different groupings of ALAS data. This tripartite division was *implicit* in the separate boxes on mainframe-ALAS forms, but the division was *conceptually separated* in the PC-ALAS menu system (Table 4.1).
This tripartite separation serves to structure users' actions. Users begin at the left menu and work their way to the right (see Figure 4.12). Yet, as Chapter 5 discusses, users often find it difficult to remember which grouping of data they are using, a discoordination that contributes to numerous breakdowns.

Users could begin using PC-ALAS with minimal training (the standard training class lasted one day, not the several weeks required for mainframe-ALAS training). Workers in local agencies could now conduct searches. However, unlike in the pre-ALAS era, when workers conducted their own queries, in the PC-ALAS era agencies tended to locally centralize the job of searching: one person or a small group of people would be designated local "PC-ALAS experts" and would field search requests from others in the agency. PC-ALAS had made the ALAS activity network more decentralized, but not utterly so.

Using a node map or a node table, users at local agencies could formulate their own queries or sets of queries and receive the results immediately. Instant feedback made it practical to make multiple subsequent queries, allowing users to form and test increasingly specific hypotheses "on the fly." Quite specific types of accidents could be sifted out of the data: accidents involving people over 25, accidents involving dry conditions, accidents involving fixed objects. The Accident Location and Analysis System had suddenly become far more useful for most users.

The various types of reports also made PC-ALAS useful. In mainframe-ALAS, paper reports were the only form of output, and came in two genres: the High Accident Ranking Report and the Generalized Request Report. These were produced at the end of the process and regarded as end products. But with the advent of PC-ALAS, reports were fundamentally transformed in three ways.

(1) Reports mingled with an interface genre — the scrollable text window — to produce a hybrid genre, the on-screen report. Such reports could be generated, reviewed, and dismissed without ever being printed. And since generating these reports was swift
and relatively easy, users began to generate reports for preliminary results which could be used for further refining queries. The reports were no longer simply end products.

(2) Report genres began to multiply. In contrast to mainframe-ALAS, which was developed by outside consultants and code-frozen in 1974, PC-ALAS was developed internally and over a period of years. Consequently, the developers responded to users’ requests by developing several flexible, configurable report formats that were easier to interpret and presented information useful to a wider range of agencies. The Generalized Request Report split into two genres, the Engineering Report (which focused on the physical roadway system) and the Enforcement Report (which emphasized behavioral factors). The Ranking Report, while it remained a single genre, was given new options to make it more flexible. And other genres were developed: the Driver Matrix Report, the Injured Matrix Report, and a report that simply listed city and county numbers. Each served specific needs of various agencies and external clients.

(3) Part of the reports’ duties were redistributed to another interface genre, the message dialog box (Figure 4.13). Message dialog boxes simply present a message; in PC-ALAS, message dialog boxes inform the user (a) how many accidents are found in a given location and (b) how many are found when searching the location by specific characteristics. Both functions were once the domain of the reports, which were the primary output method for mainframe-ALAS. But by adding the message dialog boxes, Moreland made it possible for users to monitor their data on the fly. For instance, a user who wants to view all accidents at an intersection no longer has to run the request and print a report to discover whether such accidents exist; she finds out simply by running the request. As Chapter 5 argues, message dialog boxes become important resources for users who are trying to develop hypotheses about their data.

PC-ALAS became a success. Police departments, local legislators, and engineers began to use accident data with increasing frequency. And they could do so knowing that their work was consistent with the work of others, as Emery points out:

PC-ALAS is a tool for communication among all units of government. A problem location can be studied simultaneously by the city traffic
engineer and traffic law officer, by the Iowa DOT district local systems engineer and Safety Bureau, and soon by the FHWA [Federal Highway Authority] and Governor's Traffic Safety Bureau offices, each office knowing they are operating upon the same database with the same software. Each one may be using those tools differently, based upon the agency's expertise and purpose, while retaining the ability to reproduce each other's work.

Contradictions within the PC-ALAS Activity Network

Although PC-ALAS made for a far more widespread and decentralized use of accident data than mainframe-ALAS ever had, it still had some contradictions, contradictions that became obvious by the mid-1990s (Figure 4.14).
A. Attendant genres are physically and conceptually difficult to use with PC-ALAS. These genres include cumbersome node maps (created in CAD format by the DOT and distributed to local agencies) and node tables (printed in book form from the DOT's database). Users have to acquire these off-screen genres, learn how to interpret them, and learn how to use them in conjunction with PC-ALAS. Not only are these genres difficult to interpret and use with PC-ALAS, they are physically difficult to deal with, especially the maps. The node maps are four by four feet square, about the size of the top of an office desk when unrolled, and require about that much space (Figure 4.15). In addition, they tend to roll back up, necessitating the use of other artifacts to keep them flat (e.g. paperweights, coffee cups, small office equipment) or other strategies for representing the data (e.g. photocopying a frequently used part of the map onto a flat piece of paper).
Figure 4.15 A four-by-four-foot node map held down by arms, coffee cup and Dictaphone. The map covers an entire desk and tends to roll up. This participant uses a pencil to point out and mark nodes to enter into PC-ALAS.

In addition, node maps and node tables must be used with each other and with PC-ALAS' text-based interface, meaning that users have to convert between these off-and on-screen representations. Reports are also text-based and thus required further coordination with the node map and node table genres. Report genres, although far more flexible than their predecessors, do not address the activities of all users. And the mingling of older (off-screen) genres and interface (on-screen) genres is not always seamless: for instance, input entry dialog boxes look and act differently from the standard dialog boxes that users have encountered in Microsoft Windows.

(B) Queries are inflexible and difficult to formulate. Despite the changes in the user interface, PC-ALAS came to be seen as cumbersome and inflexible by many. The
queries are characterized as "tedious" (Pawlovich, 1996, p. 25). Part of this perception, undoubtedly, comes from users' experience with other, newer programs.

(C) Data are not up to date and are sometimes inaccurate. Errors in the data entry process are propagated in local databases. And since the same data are used to plot new nodes, node maps are not always accurate, and even in low-growth areas, tend to quickly become out of date. Databases are not usually up to date either: the PC-ALAS users that I observed in the spring of 1998 were working with data up to 1995 — a limitation that has important implications for how users perceive and employ PC-ALAS, as I discuss in Chapter 5.

(D) Data are difficult to maintain, just as they were in the mainframe-ALAS era. Officers, drivers, locators, and data entry personnel all have a hand in transforming accident reports into database records. This attenuated division of labor, and the intensive and complex nature of the labor, breed errors which are then propagated to the database and node maps.

Finally, PC-ALAS itself has become difficult to maintain. Although Turbo Pascal was a popular computer language in 1989, by the mid 1990s it was for all practical purposes a dead language, so updates to the original code became extremely difficult to make. Moreland and the other programmers have all gone on to other jobs. And DOS-based programs themselves have gone the way of the 5.25-inch floppy disk.

In the waning days of the PC-ALAS era, the ALAS activity network has begun to collapse under the weight of its own contradictions. Users are beginning to develop unofficial ways to deal with PC-ALAS' limitations, such as customizing geographic information systems to take over some PC-ALAS duties and using products with overlapping capabilities. But PC-ALAS' limitations are also being dealt with through the official development of the next generation of ALAS software: GIS-ALAS.
The Ecology of Genres in the GIS-ALAS Era

Like PC-ALAS, GIS-ALAS was developed by an innovative Iowa State student. In 1995, Michael Pawlovich, a master's student in Civil and Construction Engineering, was working at CTRE. He became involved with distributing PC-ALAS disks, training users in PC-ALAS, and extracting PC-ALAS data for a CTRE project. This project brought to Pawlovich's attention some of the problems with PC-ALAS:

So we were getting accident information out of [PC-]ALAS, which involves a long, arduous process, as most people know. And then I had to put all that information on a map, which made the process all that much more arduous, and it occurred to me that perhaps these [the program and the map] could be put together, and it would be a lot quicker.

Pawlovich did so using a geographic information system (GIS), a database that allows queries and displays results through an on-screen map. He developed a version of the product, dubbed GIS-ALAS, as his 1996 thesis project and as a project for CTRE. Using the programming language FORTRAN, Pawlovich extended the GIS' capabilities to read PC-ALAS crash data and perform standard PC-ALAS functions. The result was an on-screen map that could be configured to show accidents of various types.

CTRE obtained funding for the GIS-ALAS project in the fall of 1997. With the help of an undergraduate assistant, Pawlovich moved GIS-ALAS to another GIS, Arc View (Figure 4.16), and rewrote its routines in Arc View's macro language, Avenue. The Arc View version is currently undergoing testing and development, and should be completed by the end of 1999. Development is concurrently taking place on a Web-based version.

The map is a natural candidate for migration into the interface. Edward Tufte enthuses:

Only a picture [such as a map] could carry such a volume of data in such a small space. Furthermore, all that data, thanks to the [map], can be thought about in many different ways at many different levels of analysis.
Figure 4.16 The ArcView-based GIS-ALAS interface, including horizontal menu (top), map window (left), and control window (right). These genres are versions of older genres found in the PC-ALAS interface, which in turn are versions of still older genres. (See Chapter 6 for a more detailed analysis.)

— ranging from the contemplation of general overall patterns to the detection of very fine county-by-county detail (1983, p. 16).

This flexibility of analysis was quite desirable for ALAS users: the advantage of automation is that users could quickly form and test hypotheses, but the map had been a weak link because, until the GIS-ALAS era, the map could not be automated. Users could manually mark spot maps if they wanted to contemplate overall patterns or examine spatial relations in detail, but without automation the process was labor-intensive and involved converting from the node-link system to the map genre. With GIS-ALAS, Pawlovich successfully automated the map.
Pawlovich says that "the chief advantage of GIS [is] the ability to view and select nodes and links through a spatial graphical user interface" (Pawlovich, 1996, p. 5). But perhaps GIS-ALAS' chief advantage is that users can begin to forget about nodes and links altogether. Although users can still run queries based on node numbers, GIS-ALAS makes it easy for them to simply click on a node (represented as a glowing dot on a map) instead. Users can also select a group of nodes simply by clicking and dragging the mouse appropriately. Or, better yet, users can elect not to use nodes at all: they can select areas of interest on the map instead, such as actual accident locations. Finally, queries result in highlighted spots on the map (the genetic descendants of the pins and pens used to mark spot maps since the earliest days of the ALAS activity network) as well as tabular descriptions descended from the reports of mainframe-ALAS and PC-ALAS.

In fact, with GIS-ALAS, genres continue to be imported into the genre ecology. The map is the most obvious example, but the old PC-ALAS dialog boxes, queries, reports, and menu system are also hybridized. For instance, the node map combines with the online map genre common to GISes to form the GIS-ALAS map (Figure 4.16): points of interest show up as dots; map features such as roads, railroads, and bodies of water are
Figure 4.18 GIS-ALAS activity network. Notice the changes in genres available.

...conceived of as different "themes" that can be displayed or hidden; the map can be enlarged or shrunk. Reports (Figure 4.17) are now produced as series of discrete tables (displayed in their own windows) rather than complete documents. PC-ALAS menus are adopted in partial form, but coexist with ArcView menus.

GIS-ALAS promises to transform the ALAS activity network once again, and in significant ways. GIS-ALAS should reduce the need for users to know node numbers, and will visually display map features. And the rank-and-file users may have to transform data less: rather than looking up and entering node numbers to inspect certain intersections, for instance, they can simply select the appropriate intersections on the online map. In an interview, one of the developers expressed his hopes for GIS-ALAS data representation in this way (emphasis added):

[GIS-ALAS] will give you more of a way to use your data more efficiently, and maybe you can get a better sense of what the data is
actually representing instead of having to do some secondary analysis yourself, [using] the raw numbers.

GIS-ALAS makes use of a familiar genre that users have often and skillfully employed to transform their objectives. It minimizes the "secondary analysis" of translating report results back to the map genre. Furthermore, these maps could be updated more quickly: rather than printing maps on request and sending them to each of the 99 counties, the DOT can simply post the updated maps to its web site.

Contradictions within the GIS-ALAS Activity Network

Even in its prototype state, we can speculate how GIS-ALAS might form contradictions within the ALAS activity network (Figure 4.18). These contradictions are explored in Chapter 6.

(A) Contradictions exist in the ways that GIS-ALAS presents information in its hybridized genres. For instance, even though the node-link system is theoretically rendered obsolete by the map window, users still need to understand it if they are to understand the hybridized genres of the map window and the information window, whose data displays are still based on nodes (see Chapter 6).

(B) GIS-ALAS is more difficult to distribute because it represents substantial investments in money and training. Since it is based on ArcView, users must purchase ArcView to run GIS-ALAS. ArcView retails for over $1000, putting it out of the price range of many smaller agencies. In addition, many small agencies do not own computers that are fast enough or have enough memory to run GIS-ALAS, necessitating a substantial investment in hardware. Finally, ArcView is a complex program, so local agencies will likely need to train users thoroughly.

(C) Finally, the data are still not up to date. Although the MARS initiative should narrow the division of labor needed to enter data, shortening data turnaround and improving accuracy, it is only in the testing phase at this point. The slow updating and inaccuracies that plagued previous activity networks will continue in the GIS-ALAS activity network, at least in the short term.
Conclusions

How do participants in an activity come to perceive and manage the various on- and off-screen tools used in an activity to mediate their work?

The study above indicates that interrelationships among genres develop over time. As people use computers, they collectively develop ecologies of genres: they adapt existing genres to help mediate between other genres in the ecology. In doing so, they find new ways to perceive the adapted genre and they develop new habits for managing the relationship between that genre and the others in the genre ecology. That is, they learn a new way to interpret the genre, and that way involves using it to mediate between other genres.

People find that they must frequently import genres into their genre ecology. Such adaptations of existing genres are necessary to address deep contradictions among elements of their activity network — contradictions that manifest in discoordinations and breakdowns, as I demonstrate in chapters 5 and 6.

Sometimes new information systems are developed to help mediate an activity, and when that happens, the traditional off-screen genres often are imported to the computer screen, where they are combined with on-screen genres to produce hybrid genres. Hybridized genres, the result of mingling off-screen and on-screen genres, tend to retain their history, their addressivity, and their relationship to other genres in the ecology. But as off-screen genres are imported into the interface, users continue to adapt new off-screen genres.

These points are demonstrated in this chapter's study. Over the span of four decades, an ecology of genres has grown around the activity of accident location and analysis in the state of Iowa. This genre ecology serves to mediate the transformation of accident data into analyses, analyses that are then used in mediating other activities. Within the ecology, genres serve to mediate the use of other genres: mainframe-ALAS request forms and, later, PC-ALAS dialog boxes mediated between the node maps that workers
used and the printed reports they wanted to produce. The genre ecology constantly
develops as users adapt still other genres to mediate those that already occupy it.

At three points in the history of the activity network, secondary contradictions
became so severe that agencies developed information systems to re-mediate the activity.
In all three cases, the developers transformed genres of the existing genre ecology by
importing them into the computer interface, where they were combined with other genres
to form hybrids. Just as accident report forms were combined with mainframe query
forms to produce the hybridized mainframe-ALAS request forms, those request forms
were later combined with the screen genre of the dialog box to produce the hybridized
PC-ALAS dialog box. Yet despite these radical transformations, the genres retain their
history, addressivity, and relationships with other genres. They are perceived and
managed in substantially the same ways as their predecessors.

When these new information systems are developed, the developers tend to attempt
to import most or all of the genres from the ecology into the interface. This attempt is
most pronounced in the GIS-ALAS interface. Yet users continue to adapt off-screen
genres to supplement the on-screen genres; the ecology continues to develop.

**Implications for Methodology**

In this chapter, I explore the developmental aspect of genre ecologies, particularly in
terms of the hybridization of genres. I also adapt the framework of contradictions,
discoordinations, and breakdowns to examine the sorts of events that have typically been
interpreted as usability problems. In this four-decade historical study of accident location
and analysis, I demonstrate how to culturally and historically analyze an activity
network's developing genre ecology. Although I used certain methods to investigate this
network — interviews, document analysis, and genre analysis — I could have used a
variety of other methods driven by the same methodological principles of ecological
development, hybridization, and distributed usability.

**Ecological development.** As I argued in Chapter 3, genre ecologies develop jointly.
This chapter has continued to explore ecological development by examining how users
adapt new genres into the ecology over time. The methodology I am constructing, then, must take the ecological development of genres into account, particularly in terms of adaptation. In this study I examined such developments through historical analysis and genre analysis. Other methods can also be brought to bear on this task, such as conversation and discourse analysis, protocol analysis, cooperative prototyping, and participatory design.

**Hybridization.** When genres migrate to an ecology, particularly an ecology of interface genres, they tend to combine with the ecology's genres and produce hybrid genres bearing the characteristics — the history, addressivity, and genre relations — of their parents. For instance, the genre of the PC-ALAS dialog box derives from the genre of the mainframe-ALAS request form, but it also derives from its other parentage, the genre of the dialog box that had been established in computer interfaces by 1989 (see Chapter 5). Since hybrids evoke the longstanding interpretive habits and genre relations enjoyed by their parents, an understanding of hybrids seems vital to the methodology I am constructing. In this chapter, I examine hybrids through genre analysis, but other methods such as rhetorical analysis, conversation analysis, discourse analysis, and protocol analysis could also serve to examine such hybrids.

**Distributed Usability.** As I argue in Chapter 3, usability is best conceived as a quality distributed across the activity network. This chapter continues that work by replacing the conventional notion of usability with the more robust, complex tripartite division of contradictions, discoordinations, and breakdowns. The division provides for three levels of analysis so that broader cultural-historical activities can be connected with repeated actions and moment-by-moment operations. In this chapter, I have examined the first of the three classifications: I have conducted a macro-level analysis of contradictions in a developing activity system using methods of historical analysis and genre analysis. Other methods such as document analysis and cultural criticism could similarly be enlisted.
In this chapter I have provided a macro-level analysis of the ALAS activity network. In the next two chapters, I closely examine two of the information systems (PC-ALAS and GIS-ALAS) through empirical studies. In these studies I investigate two types of discordinations — involving genre perception and genre management — as they relate to breakdowns on one hand and contradictions on the other.
CHAPTER 5
CONTRADICTIONS, DISCOORDINATIONS, AND BREAKDOWNS: A STUDY OF PC-ALAS USE

In Chapter 4, I provided a macro-level analysis of the transformations that have taken place in the ALAS activity network over the past three decades. This chapter looks more closely at this activity network as it functions in the waning days of PC-ALAS. In this chapter, I draw from interviews and observations of PC-ALAS users to analyze some of the contradictions in the ALAS activity network. In particular, I investigate the third research question: How do participants come to encounter difficulties in perceiving and managing the various on and off-screen genres?

In answering this question, I develop another vital part of the research methodology with which this dissertation is concerned. Earlier I argued that the quality of usability must be conceived as a quality of the entire activity network rather than the property of a single artifact, and therefore, many of the obvious difficulties commonly called usability problems are symptoms of deeper contradictions in the activity. Chapter 4 provides a framework for talking about aspects of usability — the tripartite division of contradictions, discoordinations, and breakdowns often used by activity theorists — and provides a macro-level analysis of contradictions in an activity network. In this chapter I flesh out that tripartite framework by connecting the contradictions identified in Chapter 4 with discoordinations (explored through a meso-level analysis of repeated actions) and breakdowns (explored through a micro-level analysis of moment-by-moment operations).

Throughout this chapter, I make three claims:

(1) I argue that the breakdowns these participants encounter are related to discoordinations in how they perceive and manage genres, discoordinations that arise from deeper contradictions within the activity. The breakdowns and discoordinations, I
claim, are *symptoms* of the underlying contradictions; thus, to deal with the breakdowns and discoordinations, designers must address the contradictions.

(2) I argue that at least some of these contradictions are between three activity networks that interpenetrate the ALAS activity network in question (city/county engineering, law enforcement, and traffic safety). Each interpenetrating activity involves transforming the same object — the ALAS data — but in different ways, to achieve different outcomes. These differences underlie many of the discoordinations and breakdowns that users experienced.

(3) I argue that users avoided some potential discoordinations and breakdowns among genres by adapting additional off-screen genres, thus spreading the work of coordination interpersonally and temporally.

I support these claims with a study of PC-ALAS users. (See Chapter 2 for a more complete discussion of the study, and Appendix D for details on participants.)

The PC-ALAS Genre Ecology: Its Contradictions, Discoordinations, and Breakdowns

In Chapter 4, I introduced the terms *contradiction*, *discoordination*, and *breakdown*. Chapter 4 discusses contradictions more fully, providing illustrations from the ALAS activity network. These contradictions, I argue, manifest themselves as discoordinations, which in turn give rise to breakdowns.

In this section I begin by discussing the sorts of discoordinations that PC-ALAS users sometimes experience. I then analyze 12 participants' breakdowns and the discoordinations and contradictions that underlie them.

In 1998, the waning days of the PC-ALAS era, three contradictions exist in the ALAS activity network: (A) between the traditional (qualitative) representation of the roadway system and the node-link (quantitative) representation, (B) between PC-ALAS (mediational means) and Windows (genre rules), and among the various activity networks that interpenetrate the ALAS activity network (e.g. city/county engineering, law enforcement, and traffic safety). These contradictions underlie various
discoordinations; two types are those related to genre perception and genre management. Although I discuss these two types of discoordinations in Chapter 4, I recap them briefly below.

**Genre perception.** Users understand a given artifact in terms of genre and apply genre habits (rules) to it. For instance, when PC-ALAS was developed in 1989, those who had used the mainframe-ALAS request forms were able to understand some aspects of PC-ALAS because PC-ALAS dialog boxes closely resembled the earlier forms. Users perceived the forms and the dialog boxes as members of the same genre. Consequently, they could draw on an established set of genre rules for using the new interface.

**Genre management.** Users — and sometimes designers — coordinate interrelated genres in an ecology in such a way as to co-mediate the activity at hand. They manage genres through genre rules applied within the genre ecology (Figure 5.1). For instance, PC-ALAS users employ dialog boxes in conjunction with node maps: they use a node map to look up the intersection for which they wanted data, then type the node number into the dialog box. Such connections between genres spring up throughout the ecology. And perceiving an artifact as being the member of a certain genre entails managing those existing genre connections: the connection between the node map and the PC-ALAS dialog box echoes the much older connection between the node map and the mainframe-ALAS form.

Figure 5.1 depicts the PC-ALAS genre ecology. The lines indicate mediatory relationships among genres, that is, the coordinations among on-screen and off-screen genres that make them cohere as an ecology and allow them to co-mediate users' activities. In the terms I have discussed above, users perceive genres (interpret the genres themselves, represented by the genres' names in the figure) and manage genres (maintain the mediatory relationships among genres, represented by the lines in the figure). As we shall see, users sometimes encounter these types of discoordinations as they attempt to perceive and manage genres, discoordinations that result in breakdowns.
Off-screen genres

PC-ALAS manuals and training materials
Surfaces
Paperweights
Writing implements

On-screen genres

Horizontal menu
Vertical menus
Dialog boxes
On-screen reports

Figure 5.1 The ecology of PC-ALAS genres and artifacts.

Perceiving and Managing Genres

In this section we see 12 participants encountering breakdowns associated with three genres: horizontal menus, vertical menus, and dialog boxes. These breakdowns can be traced back to discoordinations in genre perception and management. The discoordinations in turn spring from two deeper contradictions. One contradiction is that of the mediational means (PC-ALAS) vs. genre rules (Windows). A second is that of the qualitative representation of the roadway system (the map) vs. the quantitative representation (the node-link system). A third is the contradiction among the different activities using PC-ALAS (city/county engineering, law enforcement, and traffic safety), a contradiction that I explore more thoroughly in the second half of this chapter. (See Chapter 2 for a more complete discussion of how I collected and analyzed these data. See Appendix E for a breakdown of the data.)
Horizontal and Vertical Menus

Participants encountered several breakdowns when using horizontal and vertical menus, primarily breakdowns in which users selected the wrong item. These breakdowns result from discoordinations: users perceived these genres as similar to Microsoft Windows' menu genres and managed them in ways that they might manage Windows menus, yet there are significant differences between Windows and PC-ALAS genres. Tracing the discoordinations further, I argue, we find that they spring from the contradiction between the mediational means (PC-ALAS) and the genre rules (from Windows) that participants used.

PC-ALAS is chiefly controlled through its horizontal menu and the vertical menus that drop down from it. Both horizontal and vertical menus are genres with variations in DOS, Microsoft's Windows, Apple's MacOS, and programs running on various other operating systems.

In Windows and MacOS, horizontal menus typically structure a user's activities: users tend to start and end their interactions with the program at the left, with the File menu; they perform the next most frequent actions in the next menu over, the Edit menu; and so forth. So it is with PC-ALAS. But, unlike horizontal menus, in Windows and MacOS vertical menus do not typically imply an order. In fact, this genre is typically used to imply exactly the opposite: co-equal choices. That is the case with most vertical menus in PC-ALAS.

The genres of the horizontal and vertical menu are important to PC-ALAS users because their chief purpose is to help participants coordinate their work with all of PC-ALAS' many artifacts. They establish the structure of the activity, and in doing so, guide participants' use.

Yet the basic contradiction between PC-ALAS and Windows gives rise to discoordinations related to how participants perceive the menu genres. These participants, who tend to have considerable Microsoft Windows experience and use Windows far more frequently than they do PC-ALAS, expected certain menus to do
certain things. For instance, in Windows, users open the File menu first because it allows them to load data (hence the menu item's name and placement). But in PC-ALAS, the File menu has nothing to do with that function — the Request menu does. Similarly, in Windows the File menu contains the Print command. In PC-ALAS, printing is done through the Report menu. This particular difference resulted in breakdowns for two of the participants. One described his experience in this way:

**P22:** A lot of times I hit the File menu almost automatically, cause that’s where -- you print out of there most of the time, and that’s just where I normally -- first place I go. It’s just habit, I guess, on that. It’s not -- I gotta do that in about all Windows-based application type things....

The other participant described her attempt to print through the File menu as "just going back to my Windows thought again."

Both participants were aware of their discoordinations with the PC-ALAS horizontal menu. Yet they go "back to [their] Windows thought again." Although they were aware of the difference, they had not used PC-ALAS with enough frequency to construct a (relatively) stable genre perception, one that would help them to interpret this menu in such a way as to avoid discoordinations and attendant breakdowns. And since the horizontal menu is a means for calling up other genres, discoordinations in genre perception led to discoordinations in genre management.

Participants relied on the menu system to help them manage (i.e. locate and string together) artifacts of different genres in a sensible way, especially when they didn't remember or know what artifacts they must deal with. For instance, a participant who knew that she wanted to request data selected the Request menu and chose an option — perhaps Run Location — that set her on a path in which one dialog box led to another dialog box, and then to a message window. Once the menu choice was made, the artifacts more or less coordinated themselves in a preprogrammed procession.

Yet the menu does not always give users enough assistance in that initial coordination, as Tables 5.1 and 5.2 suggest. Of the 12 participants I observed, 10 (83%) encountered breakdowns related to the menu system. Eight users (67%) opened vertical
menus only to close them because they were not relevant or desired — that is, the sought-for option was not accessible through them. Participants who selected options sometimes decided that the choices were wrong: as Table 5.1 shows, the Run Location dialog box was canceled a record 13 times. Why? Most users wanted the City/County Location dialog box because they were interested in area-wide accident analysis (Table 5.2). Yet "Run Location" is the first item in the Request menu, and the label seems to encompass all kinds of locations.

These discoordinations and breakdowns are rooted in the contradiction between the mediational means and the genre rules of the PC-ALAS activity network (Figure 4.14). Here, participants have learned a coherent set of genre rules by using Windows, and they attempted to apply those rules to the PC-ALAS interface. Yet in doing so, participants encountered discoordinations relating to genre perception and genre management. These discoordinations manifested themselves as the breakdowns I have described (Table 5.3).

I've argued here that participants' breakdowns with menus have deeper roots: they are related to participants' discoordinations in genre perception and genre management, discoordinations that spring from the contradiction between the mediational means (PC-ALAS) and genre rules (from Windows). As I argue below, this contradiction also affects how participants used dialog boxes.

**Dialog Boxes**

The on-screen genre of the dialog box (Figures 5.2 and 5.3) provides another instance of contradictions and discoordinations leading to breakdowns. Yet two participants avoided those breakdowns by coordinating genres in different ways: by perceiving and managing dialog boxes differently from Windows dialog boxes.

PC-ALAS dialog boxes are rather similar to Windows dialog boxes. Both variants of the genre provide more or less the same sorts of input devices: text/number fields for entering alphanumeric characters; radio buttons that allow one of a series to be selected; checkboxes allowing one or more of a series to be selected; and OK and Cancel buttons allowing users to accept or reject the dialog box.
Table 5.1 Selected dialog boxes accessed by menus, and their breakdowns.

<table>
<thead>
<tr>
<th>Dialog box</th>
<th>Breakdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dialog box not relevant/desired</td>
</tr>
<tr>
<td>City/County Location</td>
<td>2</td>
</tr>
<tr>
<td>Contributing Circumstances</td>
<td>1</td>
</tr>
<tr>
<td>Description</td>
<td>1</td>
</tr>
<tr>
<td>Enforcement Report Options</td>
<td>1</td>
</tr>
<tr>
<td>Engineering Report Options</td>
<td>2</td>
</tr>
<tr>
<td>Miscellaneous Options</td>
<td>5</td>
</tr>
<tr>
<td>Run Location</td>
<td>13</td>
</tr>
<tr>
<td>Type of Vehicle</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 5.2 Number of times two dialog boxes were invoked via menus and canceled or used.

<table>
<thead>
<tr>
<th>Dialog box</th>
<th>Invoked</th>
<th>Canceled</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>City/County Location</td>
<td>48</td>
<td>2</td>
<td>46</td>
</tr>
<tr>
<td>Run Location</td>
<td>46</td>
<td>13</td>
<td>30</td>
</tr>
</tbody>
</table>

Yet for all their similarities, the PC-ALAS and Windows variants of this genre differ in important ways, ways that lead to discoordinations involving genre perception. As Table 5.1 suggests, dialog boxes are the locus of many breakdowns — the most of all the genres. This abundance of breakdowns has much to do with the role dialog boxes play in the PC-ALAS genre ecology. Whereas menus provide a gross management of artifacts necessary to transform raw data into processed reports, dialog boxes provide a fine control over particular points in the transformation. To provide this fine control, dialog boxes sport a number of different input devices (text/number fields, radio buttons, checkboxes). These input devices allow participants to coordinate each dialog box with other artifacts: node maps, node books, notes, other dialog boxes, and reports that are eventually generated from input. It's not surprising, then, that of the 68 artifacts (i.e. dialog boxes, menus, reports) that participants encountered, 46 (68%) were dialog boxes.

Given the dialog box genre's central place in the PC-ALAS ecology, breakdowns associated with dialog boxes are especially troubling.
Table 5.3 A contradiction, discoordinations, and breakdowns associated with the vertical menu.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>Mediational means (PC-ALAS) vs. genre rules (Windows)</th>
</tr>
</thead>
</table>
| Discoordinations       | *Genre perception:* PC-ALAS genres perceived as Windows genres.  
                        | *Genre management:* PC-ALAS genres imply certain mediatory relations with other genres. |
| Breakdowns             | *Genre perception:* Vertical menu not relevant/desired.  
                        | *Genre management:* Dialog box not relevant/desired. |

Figure 5.2 The City/County Request Parameters dialog box has text/number input fields as well as OK and Cancel buttons.
Nine of the 12 participants (75%) encountered breakdowns when they unsuccessfully attempted to enter input into dialog boxes. Their attempts failed primarily because PC-ALAS does not give the user precise control over input: the mouse pointer is as large as a character, as are most of the input fields, so a successful attempt at input requires the user to click in exactly the right spot, with no leeway. In contrast, although Windows input fields tend to be smaller than PC-ALAS', Windows' mouse pointer is even smaller. Consequently, Windows users are able to be more precise in their input and at the same time do not have to make a "bull's eye" to activate an input field.

However, two participants avoided unsuccessful input attempts because they had operationalized their actions differently from the others. Whereas most participants clicked the mouse cursor on input fields, P17 and P23 do not. P17, who has 11 years of
PC-ALAS experience, clicked on field *labels* — a strategy that has the same effect as clicking the fields themselves, but affords a bigger target (fields tend to be one or two characters wide, while labels are several characters wide). P23, in contrast, did not click at all: he had no mouse, so instead he controlled his input entirely through the spacebar and the Tab, Enter, and Shift keys. Other participants who had learned these alternate operations, such as P19, tended to have relatively fewer breakdowns involving unentered input than those who always used mouse input.

One might be tempted to suspect that a larger repertoire of operations helps users: the larger the repertoire, the more flexible the data entry. Yet P17 and P23 did not appear to have broad repertoires at all. Rather, the opposite was the case: they each used only *one* way to enter input. Those ways worked because they took less fine motor control. And that brings us back to genre perception: all participants were steeped in the genre of Windows dialog boxes, in which fine motor control is facilitated by the interface so they can practically rely on entirely mouse-driven input. Most of the participants perceived PC-ALAS dialog boxes as variants of the Windows-based genre, and therefore best controlled (or *solely* controlled) by clicking the mouse on the input field. For some participants, alternate forms of control simply did not exist.

Participants encountered other breakdowns with dialog boxes as well, and these have to do largely with how dialog boxes are coordinated with other genres. For instance, 58% encountered a system error resulting from incorrect entry in a dialog box: e.g. they entered a wrong city or county code (resulting from discoordination with a node map or other mediating genre representing the road system), or a wrong month or year (resulting from discoordination with a report or note they used to guide their search). Additionally, a bug in PC-ALAS ensured another sort of breakdown: if a nonexistent city or county number is entered into the City/County Location dialog box (resulting from a discoordination with the node map), PC-ALAS crashes.

In summary, these breakdowns in data input can be traced to two discoordinations. One discoordination is the users' *perception* of PC-ALAS dialog boxes as Windows
Table 5.4 Contradictions, discoordinations, and breakdowns associated with the dialog box.

<table>
<thead>
<tr>
<th>Contradictions</th>
<th>Mediatinal means (PC-ALAS) vs. genre rules (Windows). Qualitative vs. quantitative representations of the roadway system (map vs. node-link system).</th>
</tr>
</thead>
</table>
| Discoordinations | *Genre perception:* PC-ALAS genres perceived as Windows genres.  
Genre management: Users must perform difficult conversions between map and PC-ALAS genres. |
| Breakdowns | *Genre perception:* Input not entered.  
Genre management: Entered wrong information (county, city, node numbers). |

dialog boxes, leading them to apply genre rules that do not have the expected effects. The second discoordination is the users' management of the dialog box with the node map, a tricky business that involves converting data between two very different representations. These discoordinations can be traced back to two contradictions (Table 5.4): the contradiction between the mediational means (PC-ALAS) and the genre rules on which users draw (those of Windows), and the contradiction between the qualitative and quantitative representations of the roadway system.

PC-ALAS menu and dialog box genres are implicated in many breakdowns. By studying their discoordinations, we can get a better sense of how such breakdowns occur. And by studying the contradictions that underlie them, we can begin to understand how designers might re-mediate those contradictions by redesigning the interface — or even, as I argue in Chapter 6, helping to redesign the activity itself.

**Summary: Contradictions, Discoordinations, and Breakdowns in Relation to On-Screen and Off-Screen Genres**

In this section, I have argued that the breakdowns these participants encountered are related to discoordinations in how they perceive and manage genres, discoordinations
that arise from deeper contradictions within the activity. Although designers often give
attention to breakdowns (commonly understood as usability problems), these
breakdowns are only symptoms of the deeper discoordinations and contradictions that
designers might address. By tracing breakdowns to their underlying discoordinations and
contradictions in the activity, then, designers can find ways to address usability at deeper
and perhaps more meaningful levels.

I have discussed two contradictions in this section — between PC-ALAS and
Windows, and between the quantitative and qualitative representations of the roadway
system. But contradictions can also arise from juxtapositions among the multiplicity of
activities that PC-ALAS mediates. The activity of accident location and analysis is more
complicated than simply finding accident data: the ALAS activity network is
interpenetrated with many other activity networks, all of which have ways of locating
accidents and different outcomes to be served by analyzing accidents. In the next section,
I argue that three such interpenetrations generate contradictions that underlie the
discoordinations and breakdowns of specific users.

Interpenetrating Activities Mediated by PC-ALAS Genres

Up to this point I have argued that contradictions lead to discoordinations, which
lead to breakdowns. In this section I expand that claim with two subclaims.

(A) At least some contradictions that participants encountered are between the
multiple activity networks that interpenetrate the ALAS activity network in question
(city/county engineering, law enforcement, and traffic safety). Each interpenetrating
activity involves transforming the same object — the ALAS data — but in different
ways, to achieve different outcomes. Thus, different participants sometimes encountered
different discoordinations and breakdowns, depending on the different activities in which
they were engaged.

(B) Participants sometimes mitigated discoordinations by adapting additional
off-screen genres, thus spreading the work of coordination interpersonally and
temporally. This move may have prevented further breakdowns.
I take up the first subclaim first, leaving the second subclaim for the case studies below. As Russell (1997) argues, a given activity network is interpenetrated with other activity networks. Users typically engage in a given activity not for its own sake, but because it relates closely to other activities in which they are also engaged. Furthermore, when the activity involves using an information system, users from different activity networks may use that system in very different ways — to meet contradictory goals — and thus the system may be more usable for one user than another.

The ALAS activity network is a good illustration of this point. Users engage in the activity of accident location and analysis so that they can use the object — the transformed data — to achieve an outcome. But what outcome? That depends on their other activities. Those who are engaged in law enforcement, for instance, might use an enforcement report (in conjunction with other genres) as mediational means to plan new enforcement strategies, resulting in more effective enforcement, fewer and less severe accidents, and shorter response times for accidents. To explore these interpenetrating networks, I interviewed ten PC-ALAS users about their PC-ALAS use as it related to their other activities. I also observed and interviewed 12 PC-ALAS users. (See Appendix D for more detailed information on these users.)

Participants engaged in three main activities — city/county engineering, law enforcement, and traffic safety — and transformed accident data to achieve outcomes that can be used in those activities. And within those three activities, participants transformed data in three ways: diagnostically (to help them find and analyze data with the intention of formulating solutions); rhetorically (to help them argue for or against certain actions); and presentationally (to generate data representations, usually reports, which they pass to others without comment). PC-ALAS must support each of these "complex tasks" (Mirel, 1998b), which all involve perceiving and coordinating genres.

What is the shape of these contradictions among interpenetrating activity networks, and how do they affect dis coordinations and breakdowns? As Table 5.5 suggests, the three activity networks have rather different objects. Those objects affect how users
locate accidents: the nature of the queries they run, the types of problems they attempt to solve, and the scope of those problems. Since they contradict each other, they call for very different ways to coordinate genres. For instance, P17's traffic safety work for the Governor's Traffic Safety Board requires him to run queries across all 99 of Iowa's counties, yet PC-ALAS gives him no way to run queries of that scope; he has to run subqueries for each county, then tally the results by hand. Similarly, P32's work as a sheriff's deputy requires him to compile certain statistics for grant proposals, but PC-ALAS' reports do not provide these particular statistics in the format he prefers, so he finds himself keeping track of these statistics using other means. A third example is that of P19, a traffic safety worker at the Iowa DOT who often has to run comparative statistics for legislators — for instance, the number of total accidents in a county versus the number of accidents involving drivers under 21. P19 could use a built-in function to calculate such differences.

In the three case studies below, I point out certain such contradictions — between PC-ALAS and the interpenetrating activity network's object, and (secondarily) between PC-ALAS and Windows — that give rise to discoordinations and breakdowns. Furthermore, I argue the second subclaim, that participants sometimes mitigated discoordinations by adapting additional off-screen genres.

**City/County Engineering**

In this case study of Mike (P4), a city engineer, his breakdowns and discoordinations arose from two contradictions: the contradiction between PC-ALAS and Windows, and more deeply, the contradiction between PC-ALAS and the object of city/county engineering. Twelve of the 22 participants took part in this activity, which involves modifying the physical structure of roadway systems to avoid accidents. Furthermore, despite the contradictions, Mike found a way to avoid other possible discoordinations and breakdowns: during Mike's session, he adapted the off-screen genre of the handwritten note to help him coordinate two genres, the node map and the dialog box.
Table 5.5 Interpenetrating activity networks, objects, and resulting characteristics.

<table>
<thead>
<tr>
<th>Interpenetrating AN</th>
<th>City/county engineering</th>
<th>Law enforcement</th>
<th>Traffic safety</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td>Roadway structure</td>
<td>Behavior of local motorists</td>
<td>Behavior of statewide motorists</td>
</tr>
<tr>
<td><strong>Queries</strong></td>
<td>Complex, one-time-only</td>
<td>Complex; repeated for different years or locations</td>
<td>Vary in complexity</td>
</tr>
<tr>
<td><strong>Problems</strong></td>
<td>Easements, signalization</td>
<td>Enforcement strategies, grants</td>
<td>Ecological impact, legislative impact, grants</td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td>One city or county</td>
<td>One city or county</td>
<td>Many (or all) cities and counties</td>
</tr>
</tbody>
</table>

Although another engineer in his office also uses PC-ALAS on occasion, Mike uses it the most, about 24 times a year; his supervisor calls him the "PC-ALAS expert." Mike has been using PC-ALAS for five years, primarily to help the city decide whether to install signage and signalization (stop signs, yield signs, traffic lights) at intersections.

At the beginning of the session, Mike adapted an off-screen genre to mediate between the node map and the dialog box. Before starting PC-ALAS, he reached for a node map (the four-by-four-foot kind) and looked up the intersection that he wanted to investigate. He copied down its node number and put the map away. This was his way of minimizing contact with the cumbersome map, as he clarified after the session:

**CS:** ... I noticed the first thing you did, and really the only time you did it, was at the beginning you brought out your node map, you looked up a node, and then you just put it away. Do you use the node map that much, or just at the beginning of the session?

**Mike:** Just at the beginning of the session. Unless I have a number of intersections to look at. Then what I’ll do is I’ll put the node map out and I’ll write the string of nodes down that I’m gonna do ...
By adapting this genre, Mike avoided the work of converting on the fly, work that might otherwise have led to discoordinations and breakdowns (as it did for some other participants).

After noting the nodes and putting the map away, Mike conducted a series of actions meant to find data for that one particular intersection (Figure 5.4). The figure graphs the actions that Mike took over time; each data point represents an action that Mike performed on an artifact. For instance, the first six actions that Mike took have to do with requesting information. Triangles represent breakdowns. (See Appendix C for breakdown categories.)

Mike's goal was twofold. First, he had to diagnose the problems with that intersection, specifically to determine whether those problems could be mitigated with new signage or traffic signalization. Second, he had to construct a rhetorical response to those who have complained about the intersection and suggested that it needed signage.

At the beginning of the observation, Mike requested data for the intersection node that he had written down. Next, he generated a report. Although he is an engineer, at this point Mike was interested in behavioral data: he wanted to find out whether past accidents are related to careless violations, in which case adding signage would be unproductive, or whether the accidents point to inadequate signage. To explore the behavioral data, Mike generated an enforcement report and sent it to the screen. Once the report appeared, Mike drew on his own knowledge of the roadway system and its characteristics to interpret it:

Mike: [Looks at report section: Time of Day/Day of Week] See, this is out in a school area. And what I try to do is I try to find out during different times of the day when school lets out, are there more accidents when school lets out or when school begins? And what day of week those, those happened to be.

[Scrolls down to report section: Contributing Circumstances]
As far as the traffic control, there's not much you can do. There's 14 people that run the light. And this one here's another common problem. [Points to FTYROW - Making Left Turn, which indicates 21 accidents.]

[Scrolls down to report section: Time of Day]

And that depicts that as, in that hour of the day. That's the highest, at 3:00 when school gets out.

Mike was fairly sure that "there's not much you can do" to correct the current accident level through traffic control measures such as signage. Currently, the intersection has a traffic light, but "there's 14 people that run the light." In addition, 21 accidents involved drivers who failed to yield the right of way (FTYROW) when making a left turn. To Mike's experienced eye the majority of these accidents were related to drivers' inexperience. Not only was Mike transforming the data diagnostically (by confirming his hypothesis), he was beginning to transform the data rhetorically (by constructing an argument that might eventually be presented to concerned citizens: "there's not much you can do"). These two transformations could eventually help him to meet his activity's objectives by guiding his work with the roadway system and helping him to rally support for that work.

At this point, Mike decided to investigate a subset of the intersection data: accidents involving pedestrians. Although he did not state why, it's possible that he thought engineering solutions such as walkways and pedestrian signals could reduce
pedestrian-related accidents. At any rate, he found only two pedestrian accidents and generated a report to examine them.

Up to this point Mike had only encountered one minor breakdown, a mistype. But as Mike began his next search, he made a mistake that led to multiple breakdowns. *He forgot to clear the previous search.* Consequently the second search was performed only on the two accidents found from the previous search (Figure 5.5).

Mike's mistake in clearing the search was rooted in two discoordinations. The first discoordination involved genre perception. Although he should have performed certain actions in a certain order — setting up the search, running the search, and clearing the search — the search process is embodied in the genre of the vertical menu, a genre that implies no order. We can trace this discoordination to the contradiction between PC-ALAS and Windows.

The second, less obvious discoordination involved genre management: PC-ALAS does not provide a genre to help Mike monitor the scope of the search, something that Mike needs if he is to perform the complex searches that are required in his job. This discoordination was rooted in the contradiction between PC-ALAS, which assumes a progressive narrowing of search data, and the object of city/county engineering, which requires engineers to view data in multiple and complex ways. Such contradictions are important for designers to notice: they indicate possibilities for redesign that may not be immediately obvious from the breakdowns.

At any rate, Mike conducted three more searches and each time was surprised by the results: the first search turned up only one match, and the next two turned up no matches at all. Since Mike had not cleared the previous search, each subsequent search was conducted using the narrower data of the last. Finally, Mike double-checked by generating a report, which confirmed that no accidents have been found. "Well," he said, "something happened. Let's go back."

Rather than discovering how to clear the search, Mike simply re-requested the intersection data (in effect, clearing the search by replacing the current narrowed data
Mike forgot to clear his first search, so his subsequent searches were performed on a progressively narrowing set of data.

with the original pool of requested data). This time he conducted a search — on drivers between the ages of 14 and 19 — without incident, then generated and interpreted another report.

To sum up, in this case study, the obvious contradiction in this session is between the mediational means (PC-ALAS) and the genre rules that Mike tries to apply to it (those of Windows). That contradiction gives rise to discoordinations involving genre perception and genre management. By perceiving the Search menu as an instance of the vertical menu genre, Mike forms an understanding of how to manage the genres of the vertical menu, the dialog box, and the report when conducting searches. But that understanding does not allow him to consistently predict the results of his actions. The discoordinations thus lead to breakdowns as Mike finds that he has to deal with unexpected results.
Table 5.6 Contradictions, discoordinations, and their relations to Mike's breakdown in interpreting search results.

| Contradictions | Between mediational means (PC-ALAS) and genre rules (Windows).  
|                | Between mediational means (PC-ALAS) and object (complex transformations required by city/county engineering) |
| Discoordinations | Genre perception: Windows vertical menu genre vs. PC-ALAS vertical menu genre; searches as nonsequential vs. sequential. |
|                | Genre management: Searches as separate vs. cumulative; reports as separate vs. cumulative; data as simple vs. complex. |
| Breakdown       | Mike has difficulty interpreting the search results in the report. |

Yet the breakdowns are also related to another discoordination: PC-ALAS does not provide a genre to monitor the status of searches. We can trace this discoordination to a second between PC-ALAS and the object of city/county engineering (Table 5.6).

Finally, Mike avoids other potential discoordinations and breakdowns: he adapts another off-screen genre, the handwritten note, to coordinate the genres of the node map and the dialog box. By recording nodes all at once, he coordinates on- and off-screen genres — the map, the handwritten list, and the dialog box — in such a way as to distribute the action of conversion across time and across artifacts.

**Law Enforcement**

In this case study of Barbara (P24), a police officer, her breakdowns and discoordinations arose from three contradictions. One is the contradiction between PC-ALAS and Windows. Two deeper contradictions are the contradiction between the activities of road and location studies, and the contradiction between PC-ALAS and law enforcement. Three of the 22 participants were involved in law enforcement, which entails modifying and regulating drivers’ behavior to avoid accidents. During Barbara's session, she, like Mike, adapted the genre of the handwritten note to help her coordinate the node map and the dialog box.
Diagnostically, law enforcement officials like Barbara transform the PC-ALAS data to reduce high-risk behavior. Whereas engineers see the data through the lens of physical road structure, law enforcement officials see accidents through the lens of behavior and devise law enforcement strategies to deal with it. For instance, one police officer interpreted a PC-ALAS enforcement report in terms of where to place officers, when to station them there, and what laws they should most energetically enforce.

The diagnostic activities of law enforcement tend to be fast-paced. Unlike engineering projects, which take months or years to implement, law enforcement strategies are implemented in a matter of days or weeks. (In Bakhtin's term, they occupy different chronotopes [1981; cf. Zinchenko, 1996].) Officers need faster turnaround of data to provide short-term evaluations of strategies. Currently, officers provide that turnaround by returning to the pre-automation strategy of looking up paper copies of officers' reports:

CS: ... You say that you haven't used this in, what, over a year?

Barbara: Yeah, probably.

CS: And you'd be more likely to use it if you had more up-to-date information?

Barbara: We'd probably use it weekly. Because, between our chief and what he's trying to accomplish with controlling of accidents and, versus writing speeding tickets in that area, we're usually doing it manually per month, and if we had a bigger database to search from, it'd be more accurate. ...

Law enforcement officers also transform PC-ALAS data rhetorically for use in grant proposals and court cases. For one participant, this is the primary use of PC-ALAS: He calls PC-ALAS "basically a tool to write grants with" and complains that he would like the report genre to provide data in the same order as the grants he writes.

Finally, law officers sometimes transform data presentationally. Often the transformation simply involves printing reports. For instance, in Barbara's small
community, the local police department sometimes generates reports on the request of the city engineer's office.

Barbara has been with this police department about two years, and has been using Windows and PC-ALAS about that same length of time. At the time of this observation, Barbara has not used PC-ALAS in over a year. When Barbara does use PC-ALAS, she looks for accident patterns that could be useful for devising enforcement strategies, but she also fills requests for the local engineer's office. That is the work she did in this case study.

Unlike Mike, Barbara minimized the work of genre management — and thus, possible dis coordinations and breakdowns — by adapting an off-screen genre. Before starting PC-ALAS, she removed a folder from a cabinet and opened it. Inside were old PC-ALAS reports and a Post-It™ note on which node numbers are written. She removed the Post-It™, placed it beside the computer screen, and used it as a reference during the session. In this way, she used an off-screen genre to temporally distribute the task of conversion among the node map, the note, and the dialog box into which she typed node numbers.

Barbara's session was rather brief compared to Mike's (Figure 5.6). Unlike Mike, Barbara did not perform a search. Mike transformed data diagnostically and rhetorically, so he needed to perform searches to dissect the data in ways that allowed him to interpret it. Barbara, on the other hand, needed to transform the data presentationally: she needed to request the proper data, then print it for others to interpret.

Barbara started by requesting data: she opened the Select Node dialog box and specified the county number and node number for the area she wanted to inspect. She typed the county number from memory but read the node number from the Post-it™. After a message window told her how many accidents occurred at this location, Barbara immediately went to the Report menu and selected the enforcement report. The Enforcement Report Options dialog box appeared with certain options already selected. Rather than changing the options, Barbara accepted them by clicking the OK button.
Unfortunately, although the dialog box does not signal this rule, one of the selected options (Accident Summary) should only be used with data on stretches of roadway. Barbara had requested data on an intersection. So when Barbara accepted these print options, PC-ALAS displayed a red message window warning that the Accident Summary could not be selected under these circumstances. Barbara was nonplussed as she clicked "OK" to clear the message window. The Enforcement Report Options dialog box reappeared, this time with the Accident Summary option unchecked.

At this point, Barbara was not sure what to do. The error message was not terribly clear, and at the time that she dismisses it, Barbara did not seem to realize that it referred to a specific option. After it was gone she realized that one of the options should not be used, but she did not remember which one, and she certainly did not seem to know that it had automatically been unchecked. She decided to be safe by deselecting all but one option — a choice that narrowed the scope of the report she eventually printed.

The breakdown that Barbara encountered here had to do with genre management. The message window had something important to say about the dialog box, but both windows could not appear at the same time. To get around the problem, the designer ensured that the offending choice was automatically deselected — but if PC-ALAS is smart enough to correct the error here, why not automatically print the report without the offending section? For that matter, why wasn't the choice simply deactivated in the first place, as it would have been in a Windows dialog box? The different genres are poorly
coordinated, and Barbara was too inexperienced with this particular contingency to know how to coordinate them in other ways.  

Here, Barbara's breakdown in using the dialog box came about because of two discoordinations. Barbara perceived the dialog box as equivalent to a Windows dialog box, but unlike Windows, PC-ALAS won't display both the error message and the dialog box at the same time, so Barbara could not compare the two. Furthermore, Barbara could not manage the two genres in such a way as to see both the error message and the dialog box, or to understand that PC-ALAS had automatically repaired the error. These discoordinations can be traced to the familiar contradiction between the mediational means (PC-ALAS) and the genre rules (Windows genres).

Underlying this contradiction is another contradiction between Barbara's activities in law enforcement and other activities that are performed in the ALAS activity network. The Accident Summary option only works with certain types of data — data culled from along a stretch of road. This is the sort of data useful to city engineers deciding whether to regrade a road, or traffic safety workers studying the ecological impact of traffic, or police officers deciding where to place highway patrols. But Barbara is doing none of these things: she is more interested in specific intersections. It is the tension between these two very different types of data — and the activities in which each is useful — that underlies the breakdown. This tension results in a discoordination in genre management: the dialog box, with its default settings, is discoordinated with the report that Barbara wants to produce (Table 5.7). Such contradictions can indicate possibilities for redesign that may not be immediately obvious from the breakdowns.

Since she wasn't sure which option to deselect, Barbara deselected several. Then Barbara accepted the dialog box, not realizing that she had not specified where to send the report.

PC-ALAS reports can be sent to the screen, to a printer, or to a text file. These choices are controlled with radio buttons, input devices that allow users to choose one

---

20In addition to Barbara, others accidentally included Accident Summary when it was not appropriate. P16, P22, and P25 ran into similar errors, and P27 remarked on it as well.
Table 5.7 Contradictions, discoordinations, and their relations to Barbara's breakdowns in interpreting an error message.

<table>
<thead>
<tr>
<th>Contradictions</th>
<th>Between mediational means (PC-ALAS) and genre rules (Windows). Between road studies and location studies.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discoordinations</td>
<td></td>
</tr>
</tbody>
</table>
| *Genre perception:* Windows message window genre vs. PC-ALAS message window genre; Windows dialog box genre vs. PC-ALAS dialog box genre; error messages as indicating needed corrections vs. effecting those corrections.  
| *Genre management:* Between dialog box settings and desired report. Between error message and dialog box. |
| Breakdowns | Barbara is surprised by the error message.  
| | Barbara has difficulty deciding which options to deselect. |

and only one of the items. In Windows dialog boxes, radio buttons default to one of the choices. But in this situation, *none* of the radio buttons are preselected. Such a thing would not be possible in a Windows application, but PC-ALAS' nine-year-old safeguards are apparently not as robust. So, although Barbara assumed her output would be sent to one of these choices, it actually disappeared without a trace — causing numerous breakdowns as she attempted to understand what was going on.

Unaware that she had not selected an output choice, Barbara accepted the dialog box, then filled out and accepted the next dialog box without incident. At this point she encountered another breakdown: she expected the report window to appear, but it did not.

Barbara's breakdown involves reinterpreting what she had done. The breakdown arose because Barbara perceived the genre as she would a Windows genre, having the same safeguards and behaving in the same fashion. This discoordination in genre perception can be traced back to the contradiction between PC-ALAS and the Windows genres that Barbara used to interpret it.

Yet we can analyze the breakdown further. In this breakdown a deeper contradiction exists between Barbara's mediational means (PC-ALAS) and Barbara's object (the
report). PC-ALAS assumed that Barbara wanted to solve problems and generate relatively complex reports that document that problem-solving — that is, it assumed that Barbara used PC-ALAS *diagnostically*. Barbara, on the other hand, just wanted to print a simple report — she wanted to use PC-ALAS *presentationally*. Barbara's activity was almost totally centered on the report that is its object; she needed a program that would be robust and would simplify report generation as much as possible. PC-ALAS did not provide that — it is geared for activities involving more complex choices for reports. Consequently, Barbara found herself having to negotiate a complicated, confusing, option-laden dialog box, one which inhibited her attempts to manage it with the report which is her object. This discoordination in genre management led to Barbara's breakdown just as much as the differences between PC-ALAS and Windows (Table 5.8).

Barbara was puzzled. This time she went back to the Report menu, but instead of selecting the enforcement report, she selected the engineering report. She accepted the Engineering Report Options dialog box (again without selecting output) and the two dialog boxes that came after it, and was again mystified when no report window appeared.

At this point she was unsure whether she was even looking in the right place. She looked at the Report menu, then the Search menu, lingering over the Select Logic option. Then she dismissed the menus and turned to me:

**Barbara:** Can't remember how to print it now! [Pause] Do you want me to print it?

**CS:** If that's what you would normally do.

**Barbara:** Normally that's what we'd do. Let's see if I can remember how. I have, probably haven't used this in a year.

Barbara glanced at the Request menu, then scrutinized the File menu. When I asked her later why she went to this menu, she explained: "I don't know, I was probably just going back to my Windows thought again." In Windows the print function is always under the File menu.
Table 5.8 Contradictions, discoordinations, and their relations to Barbara's breakdown in interpreting the absence of a report.

<table>
<thead>
<tr>
<th>Contradictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between mediational means (PC-ALAS) and genre rules (Windows).</td>
</tr>
<tr>
<td>Between mediational means (PC-ALAS) and object (diagnostic vs. presentational).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discoordinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genre perception: Windows dialog box genre vs. PC-ALAS dialog box genre (in particular, radio buttons).</td>
</tr>
<tr>
<td>Genre management: Between dialog box and report.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Breakdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara has difficulty understanding why the report wasn't generated.</td>
</tr>
</tbody>
</table>

Not finding the choice under File, Barbara went back to the Report menu and selected the enforcement report again. At this point, she finally noticed that none of the outputs were selected, and clicked on the Printer radio button. She accepted the dialog box and the one after it, and the report printed.

To sum up, in this case study, three contradictions — between mediational means (PC-ALAS) and genre rules (Windows genres), between the activities of road studies and location studies, and between mediational means (PC-ALAS) and object (diagnosis vs. presentation) — gave rise to discoordinations involving genre perception and management, discoordinations that then gave rise to breakdowns. The second and third contradictions have to do with interpenetrating activity networks. Barbara's activities as a law officer are primarily oriented toward simple locational studies of intersections, yet PC-ALAS supports complex problem-solving and studies of lengths of road. Barbara intends to simply present the report to others, yet PC-ALAS supports diagnostic and rhetorical transformations as well.

Discoordinations and breakdowns resulted. In terms of perception, Barbara perceived artifacts as belonging to certain Windows genres. That perception misled her as she attempted to print, resulting in breakdowns as she found that the expected report did not appear. In terms of management, Barbara assumed at some points that PC-ALAS
manages genres the way that Windows does: she assumed that one may print through the
PC-ALAS File menu since one can print through Windows File menus. She also
expected the print options dialog box to work smoothly with the report she envisioned
herself producing. She encountered breakdowns when her expectations were not
fulfilled. Consequently, Barbara generated a report that has fewer sections than she might
have liked.

Finally, Barbara adapted another off-screen genre, the handwritten note, to
coordinate the genres of the node map and the dialog box. In doing so, she avoided other
possible discoordinations and breakdowns. Like Mike, Barbara dealt with the
contradiction between quantitative and qualitative representations by adding a mediatory
genre.

Traffic Safety

In this case study of Sherry (P16), a worker at the local branch of the Federal
Highway Authority, her breakdowns and discoordinations arose from two contradictions:
the contradiction between PC-ALAS and Windows, and more deeply, the contradiction
between PC-ALAS and the object of traffic safety. During Sherry's session, she adapted
the off-screen genre of the printed report to help her coordinate the node map and the
dialog box.

Six of the 22 participants take part in the activity of traffic safety, which involves
devising and implementing statewide or national policy — policy that aims to reduce
accidents by impacting both road construction and drivers' behavior. At first blush, this
activity sounds quite a bit like those of engineering and law enforcement, and in fact it is
connected with these two. But whereas engineering and law enforcement have as their
object the modification of local conditions through local emphases, traffic safety's object
is to modify statewide conditions through statewide and federal initiatives and policy.
Whereas engineering and law enforcement are conducted in local agencies such as city
and county engineers' offices, police departments and sheriff's offices, the activity of
traffic safety is conducted in diverse statewide and federal agencies such as the Iowa
Department of Transportation (DOT), the Iowa Governor's Traffic Safety Board, the Iowa State Patrol, and the Iowa office of the Federal Highway Authority. The picture is further complicated because these agencies, especially the Iowa DOT, sometimes present engineering and law enforcement data to local agencies that do not have PC-ALAS.

Sherry was taking over PC-ALAS duties from a co-worker who has just retired. Prior to this observation she had used PC-ALAS only a few times, all of them during the previous two weeks. Sherry's work involved investigating accidents on Iowa's highways, diagnosing problems, and conveying those problems to the appropriate agencies for action. It also involved comparing highway accidents with accidents on municipal and secondary roads. In the past, Sherry's projects had included investigating how new bypasses affect accident rates and advising the Iowa State Patrol of the times of day when accidents are most likely to occur. Although she had never used PC-ALAS until recently, Sherry had used Windows for five years.

In the observation described below, Sherry searched for the accident rates and times of commercial vehicles, the results of which she would then report to others in her organization. She had a number of different areas across Iowa to check on, and these areas were described in paper reports that she used to guide her PC-ALAS use.

Sherry's PC-ALAS session is depicted in Figure 5.7 below. Since Sherry's session was the longest of the observed sessions (just over 50 minutes), the graph is rather crowded. Observations are so densely packed in this figure that they appear to overlap each other.

Like Mike and Barbara, Sherry managed the map and dialog box genres through an intermediary off-screen genre, the paper report. In fact, Sherry (like Barbara) did not use a node map at all during the session, or even deal with nodes: she requested data for whole cities, counties, and highway stretches within counties. She extracted the city, county, and highway numbers, not from a map, but from paper reports.

Sherry's session followed a more or less stable pattern, as Figure 5.7 suggests. With each new report page, Sherry requested information; set up a complex search; and
Participant 16

Request | Search | Report | Breakdown

Figure 5.7  Sherry's (P16) session with PC-ALAS.

printed a report which she interpreted diagnostically. At each stage she regularly encountered breakdowns.

Since Sherry requested new data each time, she did not encounter the narrowing multiple searches that Mike encountered. However, she did encounter difficulties with searches, difficulties related to the dissonance between the familiar Windows dialog boxes and the unfamiliar PC-ALAS dialog boxes. In Windows, a dialog boxes with settings always shows the *currently selected settings*. But in PC-ALAS, sometimes (as in the search options) dialog boxes show *all settings as deselected*; they show a "blank slate" that the user must fill out. Sherry interprets these dialog boxes as she would interpret Windows dialog boxes: she assumes that since the options are deselected in the dialog box, they are by default never selected until she selects them. In fact, the opposite is true: by default, PC-ALAS selects *all options*.

Furthermore, Sherry perceived search options not as restrictive (that is, narrowing a wide pool of data) but additive (that is, adding types of data to an initially empty pool; see Figure 5.8). Therefore, each time she requested new data, she was compelled to fill out series of dialog boxes: she specified a search on days of the week and selected *all* days; she specified a search on contributing circumstances and selected *all* circumstances. The laborious nature of this job led her to suggest "You really need to invent something to do 'em all at once."

If Sherry had not elected to search on days of the week, all days would be automatically selected. But PC-ALAS does not provide ways to indicate what days are
Figure 5.8  Sherry understood the process of transforming PC-ALAS data as additive: one requests a potential set of data (the dotted oval), then uses search functions to build an actual set within those boundaries (solid ovals).

Table 5.9  Contradiction, discoordinations, and their relations to Sherry’s breakdowns in entering input.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>Mediational means (PC-ALAS) vs. genre rules (Windows).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discoordinations</td>
<td><em>Genre perception:</em> Sherry perceived PC-ALAS genres as Windows genres.</td>
</tr>
<tr>
<td></td>
<td><em>Genre management:</em> Sherry’s entered or misentered input narrows the search criteria, yielding less detailed reports.</td>
</tr>
<tr>
<td>Breakdowns</td>
<td><em>Genre perception:</em> Input not entered.</td>
</tr>
<tr>
<td></td>
<td><em>Genre management:</em> None observed, but quite possible when Sherry reexamines the reports later.</td>
</tr>
</tbody>
</table>
already selected. And this is where genre perception becomes a factor. In most Windows applications, a dialog box listing search options would have the currently active options already checked to reflect the current state of the system. But some of PC-ALAS' dialog boxes, particularly those used to set search parameters, present a potential state with no options checked. Sherry, interpreting this dialog box in terms of the Windows dialog box genre, believed that the options reflect the current system state.

Because Sherry's perception of this genre led her to spend a disproportionate amount of time checking boxes and entering numerical data, she encountered another type of breakdown. PC-ALAS input fields are not as sensitive as those of Windows: the mouse cursor, which is proportionately larger than that of Windows, must be exactly positioned over the fields to enter input. Out of the 68 times that Sherry selected and deselected search parameters, 36 times (that is, 52% of all attempts) she encountered breakdowns: she tried to enter data, noticed that the data have not been entered, and re-entered them (Table 5.9). At other times, she did not notice, and consequently some accidents were excluded from her searches; these were dis coordinations that did not immediately lead to breakdowns, but could conceivably lead to them eventually (e.g. if Sherry generated a report that did not include all of the data she wanted).

Yet the picture is complicated by the activity in which Sherry was engaged. Like Mike, Sherry's object was to transform the data diagnostically, using complex searches to cull only the accidents she needed. But unlike Mike, she performed one complex search — on a multitude of cities. Since Sherry was a traffic safety worker, she was not simply concerned with one city and one intersection; she was concerned with various cities, whole cities. PC-ALAS was not built to accommodate such statewide inquiries: it did not allow Sherry to save her complex settings from city to city and it did not allow her to search all of the desired cities at once. This contradiction between the mediational means (PC-ALAS) and Sherry's object (diagnostic reports for a range of cities) led to multiple dis coordinations among menus, dialog boxes, and reports as Sherry found herself laboriously setting up and performing the same search over and over,
encountering breakdowns along the way. Beyond the breakdowns I have already discussed, these dis coordinations led to less quantifiable breakdowns: exasperation, dulled attention, and the voiced desire for "a way to do 'em all at once" (Table 5.10).

To sum up, in this case study, Sherry faced two contradictions. One was between the mediational means (PC-ALAS) and genre rules (Windows genres); the other was between the mediational means (PC-ALAS) and her object (diagnostic reports across cities). She encountered related dis coordinations involving genre perception and genre management, and encountered multiple breakdowns springing from them. By perceiving the dialog boxes as similar to the familiar Windows dialog box genre, she formed an understanding of the system, but her genre perception was at odds with the system itself, and the discoordination between the two sometimes led to breakdowns. And because she could not perform the same search for all cities simultaneously, she had to repeat the task — exacerbating her genre perception difficulties. Again, we see that contradictions among activities are important for designers to notice because they indicate possibilities for redesign, possibilities that may not be obvious from their symptoms (breakdowns).

Finally, like Mike and Barbara, Sherry coordinated the genres of the map, the paper report, and the dialog box in such a way as to distribute the work of conversion across time and across genres. This innovation allowed her to head off possible dis coordinations and breakdowns.

**Summary: Contradictions, Dis coordinations, and Breakdowns Encountered in PC-ALAS Genres During Case Studies**

In the ALAS activity network, users transform data in various ways to achieve various outcomes, outcomes useful in each of the interpenetrating activities — city/county engineering, law enforcement, and traffic safety. As the case studies of Mike, Sherry, and Barbara illustrate, the interpenetrating activities conflict in the ways they transform and use data. These contradictions underlie many of the dis coordinations and breakdowns that users experienced.
Table 5.10  Contradiction, discoordinations, and their relations to Sherry’s breakdowns with menus, dialog boxes, and reports.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>Mediational means (PC-ALAS) vs. object (diagnostic reports across cities).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discoordinations</td>
<td><strong>Genre management</strong>: Sherry has difficulty repeatedly coordinating menus, dialog boxes, and reports to produce similar results for each county.</td>
</tr>
<tr>
<td>Breakdowns</td>
<td>Reselecting menu items, reentering input into dialog boxes, regenerating reports; exasperation.</td>
</tr>
</tbody>
</table>

Yet users sometimes mitigated discoordinations among genres by adapting additional off-screen genres, thus spreading the work of coordination interpersonally and temporally. These solutions tend to involve lessening the work of conversion by distributing it across other off-screen genres. Just as a pedestrian ascending a building may find it easier to use a series of steps than to cover the distance in one mighty leap, PC-ALAS users find the work of conversion easier when they distribute it temporally and interpersonally via other genres.

By studying discoordinations, designers can get a better sense of how such breakdowns occur. And by studying the contradictions that underlie the discoordinations, designers can begin to understand how those contradictions might be managed. As I discuss in the conclusion, this tripartite perspective affords us a complex, multilayered understanding of how genre ecologies mediate, and sometimes fail to mediate, users’ activities.

**Conclusion: Usability Revisited**

At the beginning of this chapter, I asked: How do participants come to encounter difficulties in perceiving and managing the various on and off-screen genres? As we have seen, PC-ALAS users experience many such difficulties. This point has an impact on how we evaluate PC-ALAS. If PC-ALAS users find themselves having to reinterpret on-screen genres in one of every ten actions they take, as happens in these observations,
all sorts of undesirable consequences might transpire. Users may decide that PC-ALAS is too hard to use, so they rely on their (incomplete and error-prone) paper files instead, or shunt all requests through one PC-ALAS user, or use PC-ALAS simply for writing grants, or use PC-ALAS only when other accident location programs won't do the trick, or begin to develop their own unsanctioned information systems. And in fact all of these consequences are coming to fruition.

But attending to the breakdowns themselves is like attending to the flower rather than the root, or the symptoms rather than the disease. Doing so might ease difficulties in the short run, but the underlying contradictions remain. Contradictions — particularly contradictions among the various activity networks involved in the work — lead to discoordinations among the genres in the genre ecology, and eventually to breakdowns. And although users adapt off-screen genres to mitigate discoordinations and minimize breakdowns, these adaptations tend to be ad hoc, partial solutions that do not address the underlying contradictions themselves.

So, whereas some designers may recommend remedies such as making the mouse pointer smaller or rearranging menu items, I contend that information systems evaluators should deal with deeper causes. That involves studying the underlying discoordinations, particularly how users perceive and manage genres in a genre ecology. And it involves studying the contradictions that underlie these discoordinations, and how to ease or harness those contradictions while maintaining the activity. This multileveled approach can yield a more penetrating analysis that takes into account the distribution of usability across the activity network (see Chapter 3). For instance, we can see how some breakdowns are related to the historically conditioned contradiction between Windows and PC-ALAS. If we deal with the breakdowns by making the mouse pointer smaller or rearranging menu items, we may limit breakdowns due to input errors — but since the contradiction between Windows and PC-ALAS remains, we may have to contend with other discoordinations between Windows and PC-ALAS genres, resulting in other breakdowns. On the other hand, if we address the contradiction by turning PC-ALAS
into a Windows program, these discoordinations and breakdowns may be abated. (However, as we see in the next chapter, other contradictions, discoordinations, and breakdowns can result from such a transformation.)

As I note in Chapter 4, the ALAS activity network is beginning to disintegrate under the weight of its own contradictions. This disintegration has repercussions that extend outside the ALAS activity network and into its interpenetrating activity networks. If PC-ALAS data does not adequately mediate the activity of city engineers, roads may be improperly given easements, improperly signalized, improperly repaired. If PC-ALAS data does not adequately mediate the activity of law enforcement, enforcement strategies may be placed at comparatively low-risk locations and may target comparatively low-risk violations. If PC-ALAS data does not adequately mediate the activity of traffic safety, bypasses might not get built, laws might not get passed, and ecological impacts might not be properly assessed. These omissions and improprieties have undesirable consequences of their own: property damage, lawsuits, injuries, and deaths.

**Implications for Methodology**

In this chapter I have continued Chapter 4's argument that usability can be understood and investigated through the tripartite classification used by activity theorists. Adopting this tripartite classification allows us to investigate usability as I have done in this chapter: through macro-, meso-, and micro-level analyses.

**Contradictions.** At the macro level, *contradictions* are tensions and imbalances among elements of the activity network. These are the moving forces behind disturbances, innovations, change and development of activity networks. Contradictions occur at the level of activity and can be detected through macro-level analysis of an activity network. I discuss contradictions in detail in Chapter 4, where I investigate them through historical analyses of documents and interfaces, genre analyses, and retrospective interviews. If we can identify these contradictions, we may be able to manage them and the discoordinations that spring from them.
**Discoordinations.** At the meso level, *discoordinations*, which are manifestations of deeper contradictions, are difficulties in managing the joint use of genres in an ecology. They can involve difficulties with genre perception and genre management. Discoordinations generally occur at the level of activity or action and can be found through meso-level comparative analysis of users' genre perception and genre management. In this chapter, I investigate discoordinations through naturalistic observations, case studies, stimulated recall interviews, and questionnaires, but I could also have used methods such as ethnographic research, cooperative prototyping, participatory design, and intervention protocols. Once we identify discoordinations, we may be able to find ways to better coordinate genres, and thus manage the breakdowns that spring from them.

**Breakdowns.** At the micro level, *breakdowns*, which arise from discoordinations, are points at which a user finds the present interpretation of an artifact to be inadequate for the task at hand. They can lead the user to re-perceive and re-manage genres. Breakdowns generally occur at the level of operation — that is, an operation is brought to the level of action by an unexpected focus shift. These can be found through micro-level analysis of interpretive moments, that is, points when they attempt to interpret artifacts. Concentrations of breakdowns can point to individuals' difficulties with particular artifacts, but taken cumulatively, they can also point to cross-user difficulties with genres. In this chapter I investigate breakdowns through video coding and some quantitative analysis, but I could have used methods such as experimentation and quasi-experimentation, statistical analysis, discourse analysis, conversation analysis, and protocol analysis.

The community of the ALAS activity network is determined to maintain its activity. Some agencies have explored various mediational means to use, either in conjunction with PC-ALAS or in lieu of it. In the next chapter, I investigate two prototypes of GIS-ALAS, which the DOT has designated PC-ALAS' heir apparent.
CHAPTER 6
EMBEDDED CONTRADICTIONS:
TWO STUDIES OF GIS-ALAS GENRE HYBRIDS

In their paper "Technomethodology: Paradoxes and Possibilities," Graham Button and Paul Dourish speak of what they call

the *paradox of system design* — that the introduction of technology designed to support "large-scale" activities while fundamentally transforming the "small-scale" detail of action can systematically undermine exactly the detailed features of working practice *through which* the "large-scale" activity is, in fact, accomplished. It points, fundamentally, to the interdependence of minute practice and grand accomplishment. (1996, p. 19)

Button and Dourish argue that ethnomethodology can be adapted to study this interdependence. Yet they also acknowledge the fundamental drawback to using ethnomethodology as a design methodology: although its "overriding concern with the detail of practice" positions it well for critiquing designs, that concern keeps it from moving from critique to design practice. It is *descriptive*, not *prescriptive*.

Button and Dourish suggest that designers learn from ethnomethodology in various ways. Yet they do not present a detailed methodology or a program for adapting the insights of ethnomethodology in a prescriptive fashion. They leave that challenge for others. Their main goal, the attempt to connect "minute practice" (micro-level analysis), "grand accomplishment" (macro-level analysis), and the things in between (meso-level analysis) in meaningful ways, seems to be central to any attempt to develop a design methodology. I take up that challenge in this chapter by suggesting ways to link macro-, meso-, and micro-level analyses with prescriptions for information system design.

This chapter attempts to answer the question: If information system designers import some of the many off-screen genres into the interface — transforming them into on-screen genres — how do these changes affect how users perceive and manage the genres? The answer to this question is important because it speaks to Button and
Dourish's challenge of connecting *description* with *prescription*, which is to say, connecting *evaluation* with *design*. The following two studies of GIS-ALAS use suggest that hybrid genres affect discoordinations and breakdowns a great deal:

(1) I argue that hybrid genres are points at which activity networks interpenetrate, and are thus *sites of contradiction*. Each hybrid genre is a combination of two or more genres, genres that developed in different activity networks to mediate different activities with different objectives. When the two genres are combined, the resulting hybrid genre retains a *double orientation* towards the activities: it attempts to mediate them both, and thus it embodies the contradictions between them. If we think of activity networks as continental plates which "contradict" one another by traveling in different directions, then hybrid genres are points of friction, where the plates rub together. Hybrid genres are epicenters, and breakdowns are the aftershocks. Yet these hybrid genres are also sites of change and innovation, just as fault lines generate mountain ranges.

(2) Since hybrid genres are sites of contradiction, I argue, designers can use hybrid genres as starting points for addressing the contradictions embedded in information systems. Designers can address contradictions by redesigning the hybrid genres, retooling them to better support innovation, change, and learning.

(3) But designers may also go further by *helping to redesign the very activity networks that the information systems mediate*. That is, since usability is distributed across the entire activity network (as Chapter 3 argues), the designer's work must be distributed also. Designers can find ways to encourage and orchestrate related activities such as training and data collection.

To argue these points, I first analyze the ecology of genres in the GIS-ALAS interface, particularly focusing on how GIS-ALAS has hybridized PC-ALAS genres. These hybridized genres, I argue, are sites of contradictions between the activities towards which they are oriented. Next, through two studies of GIS-ALAS use, I suggest how to address the contradictions of hybrid genres through redesign. I describe naturalistic observations and interviews of four participants using an early GIS-ALAS
prototype in their accident location and analysis work. I then describe controlled
observations and interviews of 13 students — potential users of GIS-ALAS — as they
attempt to complete a series of simple tasks using a more developed GIS-ALAS
prototype. Finally, I suggest that designers can address contradictions by designing the
users' activities themselves: I provide an extended hypothetical example in which
designers can move from designing hybrid genres to designing and facilitating the
activities in which the hybrid genres are used.

Contradictions Embedded in GIS-ALAS Genres

These two studies took place in the spring of 1998, on the boundary of the PC-ALAS
era and the GIS-ALAS era — the point at which PC-ALAS was being phased out in
favor of GIS-ALAS, a geographic information system that combines PC-ALAS
functionality with features such as an on-screen map. As Chapter 5 argues, during the
PC-ALAS era, the activity network is beginning to collapse under its own contradictions.
GIS-ALAS has been developed to deal with the symptoms of these contradictions (i.e.
breakdowns) by providing a more flexible interface. It continues the trend of importing
genres into the interface. But these genres are hybridized in radically different ways.
How does hybridization affect how users perceive and manage the genres?

One insight that can guide us as we consider this question is that hybridized genres
can embody contradictions. That is, hybrid genres are material instances of the meeting
of different activities, different traditions. As I argue throughout this section, hybrid
genres are doubly oriented, that is, oriented to both of the activities in which their parents
originated. The double orientation can pull the genre — and its users — in two (or more)
different directions, resulting (as we saw in Chapter 5) in discoordinations and
breakdowns. The theory I've used in this dissertation helps us to conceptualize these
double orientations, the epicenters around which breakdowns cluster. Using this theory,
designers can focus on these hybrid genres when redesigning a system — finding ways to
redesign/realign the double orientation to encourage innovation and productive change.
In this section, I argue that GIS-ALAS hybrid genres are meeting points between (a) the activities of accident location vs. geography, and (b) the activities of using standard databases (such as PC-ALAS) vs. visualization tools (such as GIS-ALAS) (Table 6.1). These activities are contradictory, so the hybrid genres are contradictory as well: they embody contradictions. Below, I'll trace the development and outlines of these hybrid genres, showing how contradictions have become embedded in them. Later, in the two studies of this chapter, we'll see how these contradictions affect genre perception and genre management, and how designers might redesign/realign their double orientation.

Table 6.1 Two contradictions embodied by GIS-ALAS hybrid genres.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>Points at which the Contradiction Manifests</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Between accident location and geography</td>
<td>Between genre rules and division of labor</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Between standard databases and visualization tools</td>
<td>Between genres: static reports vs. dynamic displays</td>
</tr>
<tr>
<td></td>
<td>Between objects: finished analyses vs. on-the-fly analyses</td>
</tr>
<tr>
<td></td>
<td>Between mediational means and rules: serial vs. interactive data exploration, manual vs. automated genre management</td>
</tr>
</tbody>
</table>

The First Contradiction: Accident Location and Analysis vs. Geography

The first of these two embedded contradictions, as I argue below, is between the activities of accident location and geographic location. This contradiction becomes embedded in GIS-ALAS' hybrid genres in part because of the division of labor struck by the developers of GIS-ALAS, a division of labor that limited developers' abilities to design genres. As we'll see in the two studies in this chapter, this contradiction between accident location and geographic location results in numerous discoordinations and breakdowns.

I'll explain the contradiction by tracing the history of how it became embedded in GIS-ALAS. In 1996, when Center for Transportation Research and Education (CTRE) workers decided to develop a new information system for accident location and analysis,
they used on-screen genres that had developed for a quite different activity: that of geographic location (e.g. studying the relationship between geography and other factors such as environment; pollution; road and city planning; census data; hydrology; and emergency response times). Rather than building GIS-ALAS from scratch, CTRE based it on an existing geographic information system (GIS) developed by a software company, ESRI. Consequently, CTRE was quite restricted in how much it could adapt those on-screen genres for the needs of the ALAS activity network. Thus, GIS-ALAS' hybrid genres embed a contradiction between *genre rules*, which must be adapted to the activity of accident location and analysis, and *division of labor*, which includes developers of GISes designed for the very different activity of geographic location.

Since CTRE's developers based GIS-ALAS on this existing GIS, they had to combine PC-ALAS genres (oriented toward accident location) with the GIS' existing on-screen genres (oriented toward geographic location). Figure 6.1 gives some idea of the radical restructuring of the genre ecology as PC-ALAS' genres were hybridized with the GIS' genres to produce GIS-ALAS hybrid genres.

Although the GIS genres are oriented towards a *general* exploration of *geographic* data, they were combined with PC-ALAS genres, genres that are oriented toward a *more specific* exploration of *accident* data. The resulting genres are doubly oriented, a change that is more than skin deep. The previous ALAS information systems used complex points of data, too complex for the GIS to handle. The GIS was built for analyses using simpler data points — an indicator of how different accident location and analysis is from geography. To get the GIS to work with the old ALAS data, CTRE had to reconfigure ALAS' *underlying data structure*. This reconfiguration has two consequences: for spatial displays and for textual displays.

*Consequences for Spatial Displays*

To understand the nature of these changes for spatial displays, let's revisit how data are stored in the node-link system used by mainframe-ALAS and PC-ALAS. In the node-link system, workers at the Iowa DOT have designated *nodes* — six-digit numbers
that represent points in the roadway system such as intersections, bridges, and entrance ramps. These node numbers are overlaid over a paper map to form that hybrid genre, the node map.

It's important to note, though, that the nodes' *spatial relationships* are not stored in the node-link data. That is, PC-ALAS "knows" that an accident is associated with a particular node, but it doesn't "know" *where* that node is. The data are fundamentally oriented toward explicating accidents, not toward explicating their spatial relationships. (This is a legacy of the pre-automation system, in which statewide reports focused on types of accidents rather than accident locations.) Figure 6.2 shows an abstraction of the stored data.

<table>
<thead>
<tr>
<th>PC-ALAS genres</th>
<th>GIS-ALAS genres</th>
</tr>
</thead>
<tbody>
<tr>
<td>maps</td>
<td>map window</td>
</tr>
<tr>
<td>node maps</td>
<td>horizontal menu</td>
</tr>
<tr>
<td>node tables</td>
<td>vertical menu</td>
</tr>
<tr>
<td>horizontal menu</td>
<td>selection windows</td>
</tr>
<tr>
<td>vertical menu</td>
<td>text/number input windows</td>
</tr>
<tr>
<td>dialog box</td>
<td>button bar</td>
</tr>
<tr>
<td>message window</td>
<td>message windows</td>
</tr>
<tr>
<td>report</td>
<td>Identify Results window</td>
</tr>
<tr>
<td>Detailed accidents</td>
<td>Control window</td>
</tr>
<tr>
<td>Report sections</td>
<td>report table windows</td>
</tr>
</tbody>
</table>

Figure 6.1  PC-ALAS genres and their GIS-ALAS descendants.
In Figure 6.2, the accident data for each accident are stored in a *record*, a data structure that holds two node numbers, a distance between the two, and data associated with the accident that the record describes. For instance, the first row of Figure 6.2 describes an accident that occurred near a point designated 1560, three units (30 feet) towards a second point, 1660. There's no way to reconstruct the spatial location of the accident simply from this information — a human being has to plot it on a node map (Figure 6.3).

In Figure 6.3, a user has traced the road running between nodes 1560 and 1660. He has located the accident (the X) three units from node 1560 in the direction of node 1660, and in doing so has related the nodes *spatially* — something that couldn't have been accomplished with the node-link data alone.

Compare the underlying data structure of the node-link system with that of the GIS. Since the GIS is oriented towards geographic location, GIS data are *always*, *fundamentally* related spatially, through coordinates that we can think of as x (horizontal) and y (vertical) coordinates. These data are suited (oriented) to the GIS' goal: to explore geographic data, which are fundamentally spatial relationships. Figure 6.4 shows an abstraction of how accident data are stored in the GIS.

Here, the data are all tied to specific spatial coordinates (latitude and longitude) rather than related to node numbers. Therefore a user doesn't need to relate the accident to the map: the GIS has enough information to perform that task itself.

How do the data become converted from the node-link system to the GIS data structure? The CTRE developers created a simple program to combine the old data with the nodes' coordinates, resulting in the GIS data. This system works quite well when an accident is located *at* a particular node (e.g. in an intersection). But when the accident is

<table>
<thead>
<tr>
<th>Reference node</th>
<th>Direction node</th>
<th>Distance</th>
<th>Accident data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1560</td>
<td>1660</td>
<td>3</td>
<td>(All data related to this accident)</td>
</tr>
<tr>
<td>1560</td>
<td>1660</td>
<td>1</td>
<td>(All data related to this accident)</td>
</tr>
<tr>
<td>1661</td>
<td>1561</td>
<td>4</td>
<td>(All data related to this accident)</td>
</tr>
</tbody>
</table>

Figure 6.2   Data on three accidents stored in the node-link system.
Figure 6.3  A user has plotted the accident (X) three units from node 1560, towards 1660 along the road.

<table>
<thead>
<tr>
<th>$x$ (Horizontal) Coordinate</th>
<th>$y$ (Vertical) Coordinate</th>
<th>Point Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>580</td>
<td>Accident Record (A)</td>
</tr>
<tr>
<td>120</td>
<td>580</td>
<td>Accident Record (B)</td>
</tr>
<tr>
<td>120</td>
<td>580</td>
<td>Accident Record (C)</td>
</tr>
</tbody>
</table>

Figure 6.4  Accident data in the GIS. The $x$ and $y$ coordinates tell the GIS where to plot the accident horizontally and vertically on the map. The A, B, and C records refer to different aspects of the same accident (see next section).
at a link between nodes — e.g. 30 feet from node 1560 in the direction of node 1660 — the conversion program plots a straight line between the two nodes rather than following the road, as a user would (Figure 6.5). What results is a display of accidents that appear to be located off the road.

We see the contradiction between accident location and geographic location, then, in the central genre of the map window (Figure 6.6). The map window is a hybrid of the node map and the GIS' online map. As chapters 4 and 5 suggest, the node map itself embodies the contradiction between qualitative and quantitative roadway representations (that is, between users' traditional ways of talking about accidents — in terms of addresses, intersections, landmarks, and maps — and the node-link system). Both representations contradict a third representation, the coordinate representation that the GIS uses for geographic location (Figure 6.4). As we'll see in the studies, this threefold
contradiction results in dis coordinations and breakdowns as participants used the online map, a genre that is oriented towards both accident location and geographic location. The reconfiguration of the underlying data structure that I have described, then, reassigns some of the labor of accident location from the user to the machine, and in the process introduces some inconsistencies in how data are spatially displayed. This reconfiguration also has consequences for how data are textually displayed.

**Consequences for Textual Displays**

In the PC-ALAS era, all types of data for a given accident had been stored in records, data structures that consisted of numbers and characters. These records, cryptic in themselves, were converted by PC-ALAS into information that users could understand. For instance, the number 2 in a particular slot of the record would be read by PC-ALAS and converted into categorical data: in a PC-ALAS report it would indicate that the driver sustained "moderate injuries."
But when CTRE developed GIS-ALAS, the developers had to split each record into three types of records: one for the accident information itself, one for each vehicle involved in the accident, and one for each injury sustained in the accident. These three new record types were called A, B, and C records. These records, like the PC-ALAS records from which they were derived, were full of data that appears cryptic until translated by the information system. However, since CTRE's developers were limited in what they could do with GIS-ALAS' genres — genres that are oriented towards geographic location, with its simpler tabular data — they could not always make sure the records were clearly translated. The results are data displays that are often quite difficult to decipher. For instance, in Figure 6.7 the Identify Results window shows A, B, and C records for a single accident (left pane) and cryptic information for the accident (right pane). Notice that categorical data (e.g. "Severity_c") is represented by a number whose categorical meaning is not apparent.

Since CTRE and ESRI divided the labor of development, CTRE had limited choice over the look, feel, and function of GIS-ALAS. Thus, CTRE's developers were limited in how much they could adapt genres; they had less chance to mitigate the contradiction between accident location and geographic location.

The Second Contradiction: Standard Databases vs. Visualization Tools

Up to this point, I have discussed how GIS-ALAS hybrid genres embody the contradiction between accident location and geographic location. But a contradiction also exists between GIS-ALAS, a data visualization tool, and PC-ALAS, a standard database. This contradiction, as we'll see in the two studies later in this chapter, is embedded in certain hybrid genres; the resulting double orientation manifests itself in other dis coordinations and breakdowns.

PC-ALAS and GIS-ALAS are quite different tools. PC-ALAS is essentially a database with canned queries and static displays (paper and online reports, message windows). GIS-ALAS, on the other hand, is a data visualization tool, which Barbara Mirel argues is a fundamentally different type of program because "data visualizations
Figure 6.7 The Identify Results window.

are interactive and linked dynamically" (1998a, p. 491). The interactivity and dynamism of data visualizations, she argues, allow users to accrue certain benefits that are not available within standard databases:

With visualizations, users can see the data that database, spreadsheet, statistical, and graphing programs report only textually or present through static displays. Instead of having to spend hours or even days searching through 50 or more pages of reports to analyze the relationships they need for a decision, users may interact with the data visualizations to quickly retrieve and interpret data from a 10,000-foot view and from a close-up detailed view almost at the same moment. (1998a, p. 492)

PC-ALAS users have to visualize data by reading from static displays and converting the node-link system back to the map genre; they might even mark accident locations on an off-screen genre, a spot map. In contrast, GIS-ALAS is interactive: users really can take a "10,000-foot view" showing all accidents, and they really can obtain a "close-up detailed view" of a particular clump of accidents in the next moment. And since the map window is dynamic, when users request and search accidents, the accidents appear immediately on the map. Furthermore, GIS-ALAS users can show and hide themes, layers of mapped data that overlap each other like transparent layers of acetate. For instance, users can choose whether to view or hide primary, secondary, and municipal roads, and accidents by year, by clicking on their appropriate themes.
In other words, whereas PC-ALAS is geared to produce static genres (reports) that represent *finished products* of a given search, GIS-ALAS is geared to produce dynamic genres (maps, information windows) that represent *on-the-fly analyses*. (Although GISes can provide static reports of a kind, they are far less developed than PC-ALAS' because they aren't the point; they don't represent the primary activity of on-the-fly data analysis.)

The GIS' orientation toward instant data analysis has an effect on how genres are managed. In the terms I used in chapters 4 and 5, *genre management between the map genre and other genres has been largely automated* in GIS-ALAS. PC-ALAS users had to manage the map genre in conjunction with PC-ALAS on-screen genres (menus, dialog boxes, report windows) themselves — a difficult and laborious process that requires converting between two representations of the roadway system. In contrast, GIS-ALAS automatically manages the map window's relationship with other genres. It plots the accidents, not with pins or pens, but with pixels.

This fundamental shift from a static database to a dynamic visualization tool has repercussions for genres. The static genres in the PC-ALAS ecology — node maps, horizontal and vertical windows, dialog boxes, message windows, and reports — originated in a standard database, but are imported to the interface of a visualization tool and mingle with that tool's existing genres to produce hybrid genres. Some of these hybrid genres remain static while others become dynamic, in the process embedding the contradiction between static and dynamic representations. Sometimes the double orientation can even fracture genres: one genre, the report, *splits* to produce both static hybrids (the Control window, the report table) and a dynamic hybrid (the Identify Results window). And the map, once at the periphery of the genre ecology (witness the users' heroic attempts to minimize its use through mediating genres in Chapter 5), is now the central genre.

These hybrid genres, then, are sites of contradiction because each genre is doubly oriented, oriented toward contradictory activities. Thus they become sites of discoordinations and breakdowns, as we shall see.
GIS-ALAS at Work: Discoordinations and Breakdowns that Four PC-ALAS Participants Encounter when Using an Undeveloped GIS-ALAS Prototype

In the following study of four ALAS workers, we'll see specific instances of how the hybrid genres become sites of contradiction due to their double orientation. I'll point out resulting discoordinations and breakdowns as I did in Chapter 5, but the real point is to analyze how these discoordinations and breakdowns cluster around the hybrid genres themselves, like tremors around an epicenter, and thus how designers should take these epicenters as sites of redesign. I'll suggest ways in which designers might address these contradictions by redesigning genres. At the end of the chapter, I'll go further by suggesting how they might redesign the very activities in which users are engaged.

I conducted naturalistic observations of four participants using an early GIS-ALAS prototype in their workaday tasks. In these observations, I focused particularly on how participants perceived and managed genres. I videotaped these participants, but since there were so few participants, I chose not to video code these sessions. (See Chapter 2 for the methods I used when collecting and analyzing these data.)

Participants used a limited GIS-ALAS prototype (Prototype A). Prototype A allowed participants to view and customize node maps in map windows and to examine accident data in the Identify Results window, but it did not offer ALAS-specific menus, reports, or ALAS-style search capabilities.

After describing the participants and their activities, I show how two specific contradictions embedded in hybrid genres — between accident location and geographic location, and between standard databases and visualization tools — led to discoordinations and breakdowns. I suggest ways in which these hybrid genres could be redesigned to harness contradictions.
Table 6.2 GIS-ALAS users and their experience with Windows, GISes, and PC-ALAS. Mark and Dan also took part in the PC-ALAS study in Chapter 5. Blank cells indicate unanswered questions on the questionnaire.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Interpenetrating AN</th>
<th>Age</th>
<th>Experience (Years)</th>
<th>Windows</th>
<th>GIS</th>
<th>PC-ALAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam (E28)</td>
<td>County Engineering</td>
<td>37</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terry (E29)</td>
<td>Traffic Safety</td>
<td>24</td>
<td></td>
<td>5</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>Mark (E26)</td>
<td>City Engineering</td>
<td>26</td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>Dan (E27)</td>
<td>City Engineering</td>
<td>20</td>
<td></td>
<td>2</td>
<td>1.0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The Participants and their Activities

This study involved observing four GIS-ALAS users at work, then interviewing them afterwards (Table 6.2). (Chapter 2 provides a detailed discussion of the methods I used to collect and analyze data.)

Mark and Dan (E26 and E27) were jointly working on an engineering project that required them to assess the accidents along a stretch of 13th Street. They had used PC-ALAS for some of this work (see Chapter 5) but elected to use GIS-ALAS to visualize the accidents. GIS-ALAS was new to their department; they were beta testers. Traditionally, Mark and Dan had used PC-ALAS to generate printouts of accidents at a location, then store the printouts in a file along with a photocopy of the appropriate map section. In this session their role was primarily diagnostic, although it was also presentational since they want to update the file.

Sam (E28) was attempting to assess the accidents at a particular intersection. On the morning I observed him, he had been told that the intersection may need structural modifications and that he should examine the accident history to assess what modifications might be most beneficial. Sam's role was both diagnostic (he had to assess) and rhetorical (he had to convince others of his assessment).

Terry (E29), the only non-engineer of the four, worked in Traffic Safety at the Iowa DOT. He was providing help to a consultant who was performing a study on high accident locations in a small town. The consultant needed printed data on the highest
accident locations. As it turned out, Terry had kept a file on this city which included older reports and a photocopy of a map section. Terry's role was primarily presentational: he had only to look up the accidents and print them for the consultant.

These users all had experience with PC-ALAS, and three of the four used file folders containing data generated from PC-ALAS.

**The Map Window: Interpreting the Map**

When interpreting the hybrid genre of the map window, these users encountered breakdowns and discoordinations that cluster around this genre, which embodies two contradictions: accident location vs. geographic location and standard databases vs. visualization tools. As we see below, Sam grappled with the first contradiction as he dealt with a map that combined three contradictory representations of the roadway system. Terry grappled with the second contradiction, understanding Prototype A as a static database rather than a dynamic visualization tool. After examining the hybrid genre of the map in terms of its embedded contradictions, I suggest how designers might address these contradictions to encourage productive use and innovations.

**Sam's Session: Grappling with Three Contradictory Representations**

In GIS-ALAS, the GIS map replaces the genre of the paper node map with its awkward manual linkage to PC-ALAS dialog boxes (see Chapter 5). These four users certainly saw the on-screen map window as an advantage in terms of visualization and accuracy:

**Sam:** ... I kinda like being able to see where I'm at on the GIS-ALAS. With the PC-ALAS, you had to know that node number and have the map right there, and if you typed the wrong number in you were getting information possibly for what you thought was an "intersection A" and it might have been a block away. So I guess I enjoy, or I like being able to physically see and know that that's the intersection I'm wanting. ... [Y]ou're less likely to make mistakes in that regard.

On the other hand, users did experience breakdowns associated with this hybrid genre, breakdowns that can be traced to the embedded contradictions between activities.
GIS map windows display coordinates on a grid, whereas the node-link system requires plotting accidents in reference to two nodes (e.g. 20 feet from node $x$ in the direction of node $y$). The conversion between the two representations is not precise: for instance, Figure 6.8 shows how traffic accidents that occurred on Iowa roads appear to have taken place in a cornfield.

Sam encountered breakdowns while attempting to interpret the map; these were rooted in the first contradiction between accident location and geographic location, with their different ways of representing map data. In one incident, we were looking at a municipal street appearing in the map window. The street's terminus approached but did not appear to touch a perpendicular state road. Sam was familiar with the area; I was not.

We attempted to interpret *themes*, layers of mapped data.

CS: Well, let's start with how you were looking for results at this particular node. Um, this is -- it looks like a node at the end of a street?

Sam: Uh, T-intersection. ... I've noticed that some of them [themes] don't line up just exact, but when I, I'm familiar enough with the community that I know that it's, they're off a little bit, but I --

CS: That's really odd. ... But at least you knew where the node was, you identified it, and you spent a lot of time trying to figure out how to get the information out of it. What did you try first?

Sam: Basically, I was trying to have it show -- I've found in some of the different places different nodes I've looked at, they'll list the accidents 1 through however many. And I know that this had, this is only showing me one accident. And I'm pretty sure there were probably more than just one accident [at this intersection in 1994] ...

In this dialogue, Sam drew upon his knowledge of the county's roads to interpret the map, despite the errors in the map's display (e.g. the T-intersection being displayed as two unconnected roads). Sam's knowledge of local accident history told him that other accidents have occurred in that area. What Sam didn't know was that when the map depicts multiple accidents at the same spot — as it did here — *the symbols overlap each other* and appear to be one accident.
In this incident, Sam found that he had to reinterpret the map window because it appeared to present information he knew was inaccurate. This breakdown was the result of a discoordination between the map window and the node map, genres that represent accidents in very different ways. The discoordination between these genres was in turn rooted in the contradiction among the three representations of the roadway system (qualitative, node-link, and coordinate), representations that originate in different activities (Table 6.3).
Breakdowns can be opportunities for growth and change, so Sam may eventually reinterpret these phenomena in a way that allows him to make better sense of the data. But dealing with breakdowns will not eliminate the deeper contradictions and discoordinations, which may eventually manifest themselves in other breakdowns. For instance, this incident proved to Sam that he could not trust the online map's representation of the roadway system. When using the online map in the future, Sam may find himself double-checking by consulting other maps and other roadway representations — that is, he may feel the need to constantly reinterpret the map window.

How can designers address the breakdowns associated with this hybrid genre? I suggest examining the essential contradiction that has been embedded in the genre: the contradiction among roadway representations. Accident location uses the node-link system, while geographic location uses the GIS' coordinate system.

One way to address the contradiction is to find ways to re-orient the hybrid genre of the map: that is, to reconcile the activities of accident location and geographic location in such a way as to encourage innovation and productive change rather than discoordinations and breakdowns. By understanding the underlying contradiction, we are led to solutions that we might not have otherwise imagined.

(1) Since the analysis indicates a discoordination between road representations in paper and online maps, the theory leads us to consider ways of addressing this discoordination — ways that address the underlying contradiction by helping users to draw their own connections between the underlying data representations. One solution could be to lend more time and effort to re-crafting the hybrid genre for the users' needs, and empowering users to do the same. For instance, designers could more carefully align the themes so that roads belonging to different roadway systems — such as the roads that are supposed to form a T-intersection — actually touch. Perhaps the designers could even give Sam the ability to modify the representations within limits, e.g. to tweak the represented roadways, so that he could help to shape the representation in ways that he finds most useful. By addressing the discoordination between road representations, this
Table 6.3 Contradictions, discoordinations, and their relations to Sam's breakdowns in interpreting the map window.

| Contradiction | Among three types of roadway/accident representations (originating in different activities):
|               | Qualitative: Traditional maps coupled with Sam's local knowledge of roads, addresses, and accidents (traditional accident location and analysis)
|               | Node-link system: Fixed locations arbitrarily assigned numbers; accidents plotted between those locations (computer science)
|               | GIS (coordinate) system: Roads and accidents plotted on a grid with no reference to locations or features (geography)

| Discoordinations | Between paper map genre and map window (road representations):
|                 | Paper maps: T-intersections are represented with connected lines on a single sheet
|                 | Map window: Data for primary, secondary, and municipal roads are represented in layers that do not always overlap precisely
|                 | Between paper map genre and map window (accident representations):
|                 | Paper maps: Each pin or pen mark represents a single accident
|                 | Map window: Points overlap each other exactly; one visible point may represent several accidents

| Breakdown | Sam interprets the map window as presenting highly inaccurate information

solution could help users like Sam to draw on their own knowledge when attempting to understand the underlying contradiction. In doing so, users can find the contradiction to be a source of innovation and change as well as an epicenter for breakdowns. Note that this solution grows from seeing the discoordination in terms of its underlying contradiction, then finding ways to address that contradiction rather than simply the discoordination itself; by taking the contradiction as a starting point, designers can imagine solutions that they might not otherwise have imagined.
The analysis indicates a second discoordination, between accident representations in paper and online maps. Again, designers might do well to consider ways of addressing this discoordination that also address the underlying contradiction: they might be led to imagine solutions that help users to understand accidents in terms that make sense in all three of the representation systems. A solution in this vein, then, could involve providing more flexible accident representations. For instance, rather than representing each accident with an identically sized and placed dot, as GIS-ALAS currently does — ensuring that users like Sam can see only one dot in a place at a time — designers could find ways to display the accidents distinctly, as users were able to do when they used paper maps and pushpins. For instance, designers could have the different accident symbols overlap on the screen, allowing users to detect different accidents while understanding that they all occurred at the same location. Or they could have the GIS display numbers showing how many accidents appear at each location, allowing users to understand each site as having the displayed number of accidents. Finally, designers could make it easier for users like Sam to change symbols on their own, meaning that users could more easily distinguish between different types of accidents appearing in the same location. By addressing the discoordination between accident representations, this solution could help users like Sam to understand and negotiate the underlying contradiction between roadway representations. By leading designers to see the connections among breakdowns, discoordinations, and contradictions, then, this approach leads them to address the contradiction, resulting in opportunities for innovation and learning.

_Terry's Session: Static Database vs. Visualization Tool_

Like Sam, Terry also encountered breakdowns and discoordinations involving the hybrid genre of the map window, but these breakdowns and discoordinations were for the most part rooted in the second contradiction: the contradiction between the static database and the dynamic visualization tool. Terry understood the GIS-ALAS map
window to be a static database representation rather than a dynamic visualization tool, so he was limited in his use of GIS-ALAS.

Unlike Sam, Terry was unfamiliar with the county depicted on his screen. He was providing accident information to a contractor for a small rural county, a county that Terry had not even visited before. To interpret the map window, Terry turned to more familiar genres. He used a folder of information about the county; the folder included PC-ALAS reports and a photocopy of a node map. During the observation, he sat with the report folder to his left and the map on the desk between him and the screen. At times, he would trace a road on the paper map with his finger, then trace the same road in the map window. When I asked him about the paper map, he said:

Terry: It was just luck that I already had that in the file, [which we started] when the location was first mentioned. And now it's come back that the consultant has met with these people from the city, and they said, okay, we're going to do this study. And they sent us the PC-ALAS request, that's what they're used to getting, so they say, just send us that information. ... Normally I'd try to avoid [using the paper map] by using [Prototype A], but then it'd still have to take -- print this, print the [Prototype A] nodes on a piece of paper, then go to PC-ALAS. [Emphasis added]

The last sentence is somewhat surprising at first: normally users would be expected to use Prototype A as a substitute for PC-ALAS, not as a way to produce additional printed node maps. But during the interview it became clear that Terry did not regard Prototype A as a substitute for PC-ALAS. Terry had to perform some complex searches, and the prototype he was using did not have the standard ALAS-type searches programmed into it. Terry regarded the prototype as a flexible online database of maps which he could use to mediate his normal interactions with PC-ALAS. His critique of the product centered around its ability to do that job:

Terry: ... Sometimes [Prototype A] saves me the, looking up the paper copies of the node numbers. But even then these printouts, this one is actually a pretty good one as far as being able to read the node numbers, but as soon as they're crowded like that you can't read them anymore.
So, far from making Terry's task simpler, the GIS-ALAS prototype made it more complicated and more difficult for Terry to use. Rather than replacing PC-ALAS, it mediated PC-ALAS — and it itself required mediation through the Terry's file folder. Terry ended up using the map on file to identify and print the online map.

In this session, Terry also encountered breakdowns while trying to construct and print appropriate maps — i.e. maps that include readable lettering and proper labels for map features such as nodes and streets. These breakdowns reflected the discoordination between the static paper node map and the dynamic map window: the GIS map window genre is oriented toward dynamic, on-the-fly data visualizations, but the node map genre is oriented toward simple, static road representations, and Terry — who was most familiar with node maps — understood the second orientation much better. This discoordination was in turn rooted in the contradiction between standard databases and visualization tools. The designers saw Prototype A as a visualization tool: the only (or at least the primary) tool mediating the activity of accident location and analysis. Terry, on the other hand, saw it as a standard database: a tool that provides static maps that Terry can use as input for PC-ALAS. (Table 6.4).

Again, how can designers address the breakdowns associated with this hybrid genre? This analysis suggests that designers may start by examining the essential contradiction that Terry encounters in the genre: the contradiction between standard databases (PC-ALAS) and visualization tools (GIS-ALAS).

(1) Since Terry cannot see the designers' point of view (that is, since he has trouble understanding the map window's orientation towards dynamic visualization), the analysis suggests finding ways to highlight and negotiate the contradiction between dynamic visualization tools and static databases. One solution could be to force Terry to use the map window dynamically: to simplify Terry's choices in such a way as to bar his use of GIS-ALAS as a static map database. For instance, designers could turn off the print function so that Terry cannot use GIS-ALAS in this way. This solution may eventually lead Terry to understand GIS-ALAS dynamically, yet such a brute force solution could
Table 6.4 Contradictions, discoordinations, and their relations to Terry's breakdowns in interpreting the map window.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>Between the designers' understanding of Prototype A and Terry's:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The designers: Prototype A is a visualization tool that replaces PC-ALAS and its genre ecology</td>
</tr>
<tr>
<td></td>
<td>Terry: Prototype A is a standard database that supplements PC-ALAS and its genre ecology</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discoordination</th>
<th>Between map genre and map window:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Paper maps: Detailed information; high resolution; used in itself</td>
</tr>
<tr>
<td></td>
<td>Map window: Less detailed information; poor resolution; produces paper maps of lower quality than existing node maps</td>
</tr>
</tbody>
</table>

| Breakdowns | Terry has a difficult time constructing and printing a map that includes the features he expects from a node map |

... sharply limit GIS-ALAS' possibilities and, worse, could convince Terry not to use GIS-ALAS at all.

(2) Another, opposite solution could be to encourage the orientation to which Terry is already predisposed: to facilitate Terry's use of GIS-ALAS as a static database. Rather than inhibiting that use, designers could create settings that allow Terry to quickly and easily print the detailed node maps that he wants. This solution lets Terry use the map window in the way that seems most intuitive to him. Yet this solution could encourage Terry to continue using GIS-ALAS in limited ways, and continue his dependence on PC-ALAS for the "real" solutions.

The two solutions above involve using the analysis to identify and encourage one orientation over the other. But the analysis can also lead us to find ways to harness the contradiction: to highlight it in such a way as to encourage learning and innovation.

(3) A third solution could be to find ways to make Terry comfortable with both orientations (static database and dynamic visualization tool). This might involve leading Terry through steps for using GIS-ALAS as a visualization tool, steps that he could choose to follow or not. For instance, at the beginning of the session GIS-ALAS could...
present Terry with a wizard (a specialized dialog box) that helps him to fill out the proper information in order. The wizard could thus scaffold Terry through the basic steps of on-screen dynamic interpretation. In this way, Terry can learn new possibilities for using the map window — both static and dynamic — and ways to relate the two. In this solution, the analysis leads designers to address the contradiction by scaffolding Terry through one way of using GIS-ALAS, a way that opens new possibilities without precluding the old ones.

(4) Similarly, a fourth solution could be to contrast the two orientations (static database and dynamic visualization tool). Designers could fundamentally redesign the hybrid genre to provide a clear choice between static and dynamic use. Just as the PC-ALAS report genre has been split into static and dynamic components in GIS-ALAS, the map genre could also be split into static (printable) and dynamic (on-screen) genres. In this way, Terry can choose which representation to use for a given activity; he can become familiar with both orientations and choose the one he finds most appropriate for his current activities. In this solution, designers are led by the analysis to highlight differences between static databases and dynamic visualization tools — a solution that they might not have imagined without the benefit of the analysis tracing breakdowns and discoordinations back to this contradiction.

The first and second solutions are quite limited, relying on brute force solutions to forcibly guide users in one way or another. On the other hand, the third and fourth solutions multiply rather than limiting users' choices: they provide scaffolding rather than narrow choices or no choices. The contradiction between static and dynamic databases is not buried, it is explored, and thus users can make informed decisions about how to handle the contradiction.

To sum up, in this section we saw that the map window, a hybrid genre, was the center of a cluster of breakdowns and discoordinations because it embodies contradictions. By tracing this group of breakdowns back to their epicenter, the hybrid
genre with its embedded contradictions, we find opportunities to harness contradictions via redesign.

The Identify Results Window: Obtaining Accident Information

Users encountered breakdowns and discoordinations when obtaining information on specific accidents clustered around the hybrid genre of the Identify Results window. These are rooted in a contradiction embedded in the Identify Results window, the contradiction between accident location and geographic location.

All participants clicked accidents on the map to display accident data in the Identify Results window. For Mark and Dan, these data were the heart of the observed activity: they needed detailed information on each of the accidents occurring at or near a particular intersection so that they could understand how factors such as weather, road structure, signage and behavior influenced accidents. Yet they encountered breakdowns when attempting to perform this activity and obtain this information. These breakdowns can be traced to discoordinations in genre perception and genre management.

Their genre perception difficulties centered around the way in which the data were presented. Since the Identify Results window showed the underlying structure of the data (the A, B, and C records) rather than the more processed view given by PC-ALAS reports, Mark and Dan had to learn that underlying structure. This took a while. Near the end of the session, after looking at several accidents and various A, B, and C records, it dawned on Mark what the different records were for:

Mark: So one of 'em has to do with the people involved and others with, with the accidents or vehicles.

Mark was correct: each accident is represented by one "A" record (information on the accident itself), along with a "B" record for each vehicle involved and a "C" record for each injury. Yet this structure only began to make sense after Mark and Dan looked at over a dozen accidents. It was not apparent in the record identifiers.
Neither was it apparent in the field labels displayed within the records. These labels are cryptic, and the data within the fields are often represented by numerical data (e.g. Vehicle\_ty=1; Severity\_c=3; see Figure 6.9).

In Figure 6.9 the detailed accident information in PC-ALAS reports is hybridized with the GIS' Identify Results window. The two genres are oriented towards very different activities: reports facilitate the interpretation of complex underlying data for accident location, but the Identify Results window displays the underlying **data structure** "as is" since it assumes the simple, easy-to-interpret data that are usually used in GISes for geographic location.

In the mainframe-ALAS and PC-ALAS eras, the data were stored in abbreviated, primarily numerical formats (records). When the computer was asked to produce a report, it would convert the data, providing appropriate labels and explanatory text to help users interpret the data. The top of Figure 6.9 shows such a report; categorical data such as the day of week the accident occurred, represented here in a readable abbreviation ("Thu"), are stored in the database as numbers. The bottom of Figure 6.9 shows similar data in the Identify Results window, which does far less in converting the data ("Day\_of\_wee = 7"). Other data are even more cryptic ("Severity\_c=3"):  

**Mark**: [W]e don't know what those codes mean. You know, some have the severity or whatever, and it has "3" ... if we had a table that explained what the different codes with each of the things were, you know, you could get a lot more information about an accident.

Mark's remark about "a table that explained ... the different codes" is interesting in that it points to a discoordination in genre management. The field data are meant to fill the same role as the detailed section of a PC-ALAS report. But since Mark found the field data to be cryptic, Mark saw an unfilled niche and suggested that he might fill it by adapting another genre for mediating the field data, such as a table or a legend. This seems to be one way that genre ecologies expand, as we saw in Chapter 5.

To sum up, breakdowns and dis coordinations cluster around their epicenter, the hybrid genre of the Identify Results window. These are manifestations of the embedded
Reference Node: 218140  Direction node: 999999  Distance: 9.99  1/18/96
Case #: 60004663  Property Damage: $ 1500.00  Thu  Time: 1530
Fatalities: 0  Injuries [Major: 0  Minor: 1  Possible: 0]
Vehicles Involved: 2  Report Filed By: Officer  Story

VEH # 1 Initial Direction of Travel: East  ** FTYROW making Left Turn
Vehicle Action: Turning Left
Vehicle Type: Passenger Car
Vehicle Defects: No Defects
Drivers Sex: M  Age: 23
Driver Condition: Apparently Normal

VEH # 2 Initial Direction of Travel: Unknown  ** Ran Traffic Signal
Vehicle Action: Going Straight
Vehicle Type: Passenger Car
Vehicle Defects: No Defects
Drivers Sex: M  Age: 22
Driver Condition: Apparently Normal

Accident Type: Broadside / Left Turn  Hit & Run: N
Light Conditions: Day  Weather Conditions: Snow, Strong Wind
Surface Conditions: Wet, Dry

Figure 6.9  (Top) Detailed accident information from a PC-ALAS report, which repackages the underlying data structure for easy reading. (Bottom) The Identify Results window, which reveals the underlying data structures.

contradiction between accident location and geographic location. The hybrid genre is oriented toward both accident location (users need complex textual data) and geographic location (users need simple alphanumeric data). Thus users found it difficult to interpret data according to the familiar genres and they had trouble understanding the GIS' ways of interconnecting the new genres. Consequently, users tended to limit their interpretations of the data they received, either partially completing their interpretive attempts or abandoning those attempts outright (Table 6.5).
How can designers address the breakdowns associated with this hybrid genre? They can start with the essential contradiction that users encounters in the genre: the contradiction between data representations oriented towards accident location and geographic location.

(1) The analysis suggests that the contradiction between data representations gives rise to a discoordination between identifiers. One solution, then, could be to use representations that are more closely aligned to accident location activities. Designers could use terms and units that have arisen or been adapted in the ALAS activity network, making the text look more like PC-ALAS reports. Thus, users would not have to translate cryptic alphanumeric identifiers (e.g. "Severity_c=3") into more meaningful text (e.g. "moderate injuries"). And since the designers have been led to address the discoordination between identifiers, users may experience fewer breakdowns. Thus, by leading us to see the connections among breakdowns, dis coordinations, and contradictions, this approach leads us to address the contradiction, resulting in opportunities for innovation.

(2) The analysis also suggests that the contradiction gives rise to a discoordination between units of data. Designers can help to coordinate and reconcile units of data by unifying the data representations. For instance, designers can reconfigure GIS-ALAS' data so that clicking on one dot gives users a single record for each accident, not multiple A, B, and C records that represent multiple aspects of the same accident. If each accident location was represented with a single record, then users would not have to juggle three or more records to get an overall understanding of each accident. Thus, users may experience fewer breakdowns arising from the discoordination, and they may be able to better reconcile data representations.

(3) The above two solutions are not the only ways to deal with the contradiction between data representations. A third solution suggested by the analysis could be to adapt on-screen genres to allow users to see all connected accident data at once, linking the disparate types of data graphically. For instance, clicking on a single accident might
Table 6.5 Contradictions, discoordinations, and their relations to breakdowns in obtaining accident information.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>Between data representations:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>In the ALAS lifeworld:</em> Accidents represented in primarily qualitative ways, with text descriptions (for human use and management)</td>
</tr>
<tr>
<td></td>
<td><em>In Prototype A (originating in geography):</em> Accidents represented in primarily quantitative ways, with alphanumeric identifiers (for computer use and management)</td>
</tr>
<tr>
<td>Discoordination</td>
<td>Between familiar and unfamiliar units of data</td>
</tr>
<tr>
<td></td>
<td>Between familiar and unfamiliar identifiers</td>
</tr>
<tr>
<td>Breakdowns</td>
<td>Users find it difficult to interpret data according to the familiar genres</td>
</tr>
<tr>
<td></td>
<td>Users have trouble understanding the GIS' ways of interconnecting the new genres</td>
</tr>
<tr>
<td></td>
<td>Users' attempts to interpret records and fields are often only partially completed or abandoned outright</td>
</tr>
</tbody>
</table>

bring up several windows: one for basic information, one for property damage, one for injuries, etc. The windows could be connected by lines. This sort of arrangement would preserve the distinction among these types of data while still unifying them. Again, users would have a way to envision and reconcile the contradiction that the analysis identified between data representations. This solution could address the contradiction, making it a source of innovation and change for users, not just breakdowns and discoordinations.

The three solutions all involve changing things: the underlying data structure, the screen genres, or both. But by changing these representations, designers can re-orient the genres to address the contradiction between representations. In doing so, they can make this hybrid genre a site of innovations and change rather than a site of breakdowns.
Summary: Contradictions, Discoordinations, and Breakdowns Encountered by ALAS Workers

In this study, I've argued that certain hybrid genres in GIS-ALAS have embedded two contradictions: (a) between accident location and geographic location, and (b) between standard databases and visualization tools. Because hybrid genres embody contradictions, they are epicenters around which discoordinations and breakdowns form. Such breakdowns are troubling, not just for the users, but for all who are concerned about the safety of Iowa's roads. If users have difficulty interpreting the map, the accidents, and the accident data — if they find themselves routinely miscounting accidents or abandoning their interpretation of accident information — the results of their work may mislead them and others in related activities. Engineers may not become aware of the need for signage; law enforcement officers may fail to lend the proper attention to particularly dangerous areas and behaviors; legislators may believe that their initiatives have worked when in fact they have not.

Sam, Mark, Dan, and Terry dealt with half-familiar hybrid genres, genres that resemble their "parents" in the ALAS activity network. They are experienced with PC-ALAS genres and the ALAS activity network, but unfamiliar with GIS genres or geographic location. Yet new employees may face very different challenges when learning GIS-ALAS as part of learning accident location and analysis. How can designers redesign the hybrid genres to help these new users learn GIS-ALAS?

GIS-ALAS at School: Discoordinations and Breakdowns that Students Encounter when Using a Developed GIS-ALAS Prototype

We've looked at how experienced ALAS workers perceive, use, and experience difficulties with hybrid genres in GIS-ALAS because these genres are doubly oriented-towards (a) accident location and geographic location, and (b) static databases and dynamic visualization tools. The workers we have studied are experienced with PC-ALAS genres and the ALAS activity network, but unfamiliar with GIS genres or
geographic location. What happens if the situation is reversed — as will happen as new employees, who have learned GISes in school, are hired for ALAS activities such as city engineering? Will users who are familiar with GISes, but unfamiliar with ALAS activities and genres, also find these hybrid genres to be epicenters for their breakdowns? And how can designers redesign the hybrid genres to, among other things, facilitate the training of these new users? Such questions lead us to this second study, another way to test the usefulness for design of studying breakdowns associated with contradictions.

In this study, I investigated these questions by studying how 13 students in a GIS class use a GIS-ALAS prototype to perform accident location and analysis. I found that the embedded contradictions were still epicenters for contradictions, as the theory suggests. Students' breakdowns clustered around the same hybrid genres as the workers' breakdowns had — although the breakdowns themselves tended to be quite different. These differences point to a third contradiction, between students' activities (i.e. learning) and ALAS activities. Again, then, (1) hybrid genres are sites of contradiction because of their double orientation, and thus they are epicenters for breakdowns; so (2) hybrid genres are, then, points for productive redesign. (3) That redesign may involve reaching past the information system to design activities as well — in this case, training activities. I explore this third claim in limited ways here, and in a more thoroughgoing way in the next section.

**The Participants and their Sessions**

I observed 13 students enrolled in a class on geographic information systems. (Chapter 2 provides a detailed discussion of the methods I used to collect and analyze data.) Students were enrolled in a GIS class offered by Iowa State University. Students average just under half a year experience with GISes in general; four of the thirteen have experience with the particular GIS on which GIS-ALAS is based. They say it's very similar to the other GISes they have used.

The class in which the students are enrolled is offered through Community and Regional Planning, a field that bears directly on accident location and analysis, and in
fact, the professor had briefly discussed the GIS-ALAS project with the class. Ten students are Community and Regional Planning majors and one student majors in Public Administration; the other two students' majors have less to do with accident location and analysis.

Students' breakdowns tended to cluster around the map window and the Identify Results window. Again, the double orientations of these hybrid genres towards contradictory activities resulted in discoordinations and breakdowns. These embedded contradictions point to opportunities for redesign, both in terms of the interface itself and the training that users receive.

**Map Window**

Recall that the map window in GIS-ALAS is a hybridized genre, one that embodies contradictions among three roadway representations (and the activities in which those representations originated). This contradiction was the root of 17 breakdowns that students encountered while using this window. In particular, students' breakdowns had to do with *themes*, the layers of map data that users can show or hide. These breakdowns reflect the differences between accident location and geographic location, but also the differences between the complex GIS-ALAS themes, oriented toward accident location, and the simpler themes used in school activities, oriented towards learning. Since the breakdowns alert us to these differences, they can help designers to find hard-to-learn aspects of the genre (that is, aspects that are difficult to reconcile with traditional learning activities) and redesign those aspects.

Students' school activities involve learning to use GISes through *relatively simple applications*. Thus students are used to dealing with perhaps half a dozen themes with simple names such as "hydrology" and "rail." The GIS assigns a distinct color to each theme so that students can easily differentiate the themes in the map window.

But the activity of accident location and analysis involves far more complicated data, data that require users to learn *different coordination strategies* if they are to make sense of the data. In the task that I gave the students (a simple task, in ALAS terms), they had
to deal with 21-36 themes. These themes are so numerous that GIS-ALAS cannot list all of them in the themes pane. Themes are so numerous that they require longer, more complicated names to distinguish between them. Finally, themes are so numerous that although the GIS tries to assign distinct colors for each theme, it runs out of distinct shades, so the colors of different themes are sometimes quite similar. These discoordinations lead to breakdowns: students can't tell how many themes there are, what each theme represents, or which themes are associated with which features on the map (Table 6.6).

The themes are numerous partly because of how designers implemented the underlying data structure, using fragmented rather than unified representations of accidents (as I've discussed earlier). These fragmented representations reflect the contradiction between accident location and geographic location. But they also have repercussions for learning GIS-ALAS, since it involves a contradiction between accident location and students' activities. How can designers address the breakdowns associated with this hybrid genre?

(1) Since the analysis allows us to trace breakdowns and discoordinations back to the contradiction among data representations, one solution could be to unify the data representations, as I've discussed earlier. In addition to the benefits in reconciling accident location and geographic location, this move would help students: it would simplify the data for the users by presenting only one record per accident, not three or more. Since students would not have to juggle three or more records to get an overall understanding of each accident, they could more easily build up a conceptual understanding of what data are involved in accidents and how those data interrelate.

(2) A second solution that also addresses this contradiction could be to provide simplified learning activities for new users. For instance, in this study students had difficulty understanding how the startlingly large number of themes interrelated. However, if the students were initially introduced to only one accident, they could begin to learn how the comparatively small number of themes interrelated, and they could
Table 6.6 Contradictions, discoordinations, and breakdowns related to number of themes.

<table>
<thead>
<tr>
<th>Contradiction</th>
<th>Between students' activities and activities of the ALAS lifeworld:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students' activities: Schoolwork and test-taking — activities that involve simplified GISes</td>
<td></td>
</tr>
<tr>
<td>Activities in the ALAS lifeworld: Finding accidents through reducing complexity — an activity that involves a complex GIS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discoordinations</th>
<th>Between familiar themes panes and Prototype B's theme pane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between familiar theme names and Prototype B's theme names</td>
<td></td>
</tr>
<tr>
<td>Between familiar easy-to-distinguish color assignments and Prototype B's difficult-to-distinguish color assignments</td>
<td></td>
</tr>
</tbody>
</table>

| Breakdowns | Students can't tell how many themes there are, what each theme represents, or which themes are associated with which features on the map |

begin to build a base for understanding more complex maps. The result could be innovation and learning as well as the breakdowns and discoordinations that students encountered during the study.

The first solution involves changing the GIS-ALAS screen genres; the second involves actually changing the students' activities (i.e. setting up training sessions or tutorials). Both, I think, could work, but a combination of the two can make this hybrid genre a site of innovations and change for these students rather than simply a site of breakdowns.

Students also encountered breakdowns and discoordinations rooted in the contradiction between roadway representations arising from different activities. Students are familiar with the coordinate representation of roadways used by GISes, including the simple theme labels I mentioned above. But the GIS-ALAS map window also draws from the node-link system, which makes its way into the themes' names. Each accident theme begins with the number of the county in which the accidents take place. (For example, the 1994 B records in Story County are stored in the theme named "Co. 85 Yr.
Consequently, students have difficulty coordinating theme labels with the themes that show up on the map; they're not sure what each map feature represents. Students thus encounter breakdowns when trying to interpret theme names, select themes, and distinguish among different themes on the map. See Table 6.7.

How can designers address the breakdowns and discoordinations arising from this contradiction?

(1) Since the analysis helps us trace the breakdowns and discoordinations to the contradiction between roadway representations, one solution that addresses this contradiction could be to eliminate the node-link representation altogether, replacing it with the more familiar county names. For instance, the theme name I mentioned above could become "Story County Yr. 94 B Records." This change would help students to more easily interpret the themes and select themes that are appropriate for their activities because it favors one representation over the other. On the other hand, this measure does not eliminate the contradiction among roadway systems by any means — it papers over the contradiction instead.

(2) The analysis suggests another solution, one that attempts to highlight the contradiction between roadway representations rather than eliminating it. This second solution is to highlight the differences between the node-link representation and the more familiar geographic representation. Designers could use combinations of the two representations to help scaffold users, teaching them to convert from one representation to another. For instance, if the theme above were labeled "Story (Co. 85) Yr. 94 B Records," users could learn to associate Story County with its number, 85. If users encountered the node-link representation (85) later, they would be more equipped to interpret it. By highlighting this difference, designers can make this contradiction an opportunity for learning, innovation, and change.

The two solutions involve changing the theme labels, but the small difference between the two can lead to large differences in usability later: whereas the first solution papers over the contradiction between two different activities (accident location and
Table 6.7 Contradictions, discoordinations, and breakdowns related to theme labels.

<table>
<thead>
<tr>
<th>Contradictions</th>
<th>Between representations of the roadway system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discoordination</td>
<td>Between representations of county designations</td>
</tr>
<tr>
<td>Breakdowns</td>
<td>Students are unsure how to interpret theme names; select inappropriate themes; have trouble distinguishing themes</td>
</tr>
</tbody>
</table>

geographic location), the second solution highlights the contradiction, and in doing so it equips new users to learn about it, understand it, and deal with it. Thus designers can make this hybrid genre a site of innovations and change rather than a site of breakdowns.

In this section, I've argued that the map window hybrid genre continues to be an epicenter for breakdowns, and that these breakdowns can inhibit new users from learning the system. But contradictions, I've argued, can also be sources for learning and innovation. I've outlined ways that identifying hybrid genres' contradictions can help designers harness contradictions to make genres (and the concepts they represent) more learnable, and how designers can embed learning in larger activities such as training.

Below, I suggest that identifying hybrid genres' contradictions can offer ways of redesigning another genre, the Identify Results window, to furnish similar benefits.

**Identify Results Window**

The students' breakdowns, like those of the experienced workers, also clustered around the Identify Results window. This clustering points to an embedded contradiction in this hybrid genre, the contradiction between accident location and geographic location. The clustering points to learning difficulties, and thus opportunities for redesign. Below, I explore the breakdowns, discoordinations, and contradiction encountered by the students, then draw on the analysis to explore opportunities for redesigning this genre.

Students experienced breakdowns interpreting the differences among A, B, and C (accident, property damage, and injury) records so they were often unable to find the data that I asked them to find in the protocol. And even when participants were able to select the proper records from the Identify Results window, they were often unable to interpret
data identifiers within each record. These identifiers are often cryptic, as are the data they hold. For example, in the exit interview, G6 discussed difficulties with interpreting various identifiers:

G6: Well, I guess "int class" [int_class], that's something that looking at it I wouldn't know. "ref underscore node" [ref_node], that, I mean, "dir underscore node" [dir_node] -- But things like "total killed" [total_kill], "total injury" [total_inju], "total vehicle" [total_vehi], those for the most part you can look at them and figure out.

Like the workers in the previous study, these students experienced breakdowns when interpreting records and the identifiers within the records. These breakdowns arise from two discoordinations. In terms of genre perception, even students who are familiar with GIS programs are not equipped to interpret the cryptic identifiers and the categorical data that are associated with them. In terms of genre management, students cannot view A, B, and C records side by side — something that would have helped G6 in particular as she struggles to make sense of the record types. These discoordinations are rooted in a contradiction between the complex underlying data structure (representing ALAS data) and the GIS data representations with which students are familiar (Table 6.8).

Above, we see that the contradiction between the activities of accident location and geographic location leads to multiple discoordinations and breakdowns that cluster around this hybrid genre. How can designers address the breakdowns associated with this hybrid genre? They can start with the essential contradiction that students encounter in the genre: the contradiction between data representations oriented towards accident location and geographic location.

(1) The analysis suggests that designers might handle the contradiction by addressing one discoordination that develops from it, the discoordination between different data representations. One solution could be to use representations that are more closely aligned to accident location activities. Designers could use terms and units that have arisen or been adapted in the ALAS activity network (e.g. "moderate injuries"), rather than the cryptic alphanumeric identifiers that are presently used (e.g. "Severity_c=3"). And since these terms and units belong to the ALAS activity network, students can use
them in conjunction with training materials to learn ALAS activities. By leading us to see the connections among breakdowns, discoordinations, and contradictions, this approach leads us to address the contradiction, resulting in opportunities for innovation and learning.

(2) Another entry point suggested by the students' reactions is that of the second discoordination, between different ways of arranging data. One way to address this discoordination (and its underlying contradiction) could be to unify the data representations, as I've discussed earlier — that is, to present a single unified record or representation for each accident, rather than several. This move would simplify the data for the users: it would present only one record per accident, not three or more. Since students would not have to juggle three or more records to get an overall understanding of each accident, they could more easily build up a conceptual understanding of what data are involved in accidents and how those data interrelate. That is, such an arrangement may help students to learn more about the accidents — but it might also help them to learn more about how different aspects of accidents relate within the activity of accident location and analysis. Thus students can learn about the vocabulary, values, and connections of data in the ALAS lifeworld. By leading designers to see the connections among breakdowns, discoordinations, and contradictions, then, this approach leads them to address the contradiction, resulting in opportunities for students to learn and innovate.

These solutions involve changing the screen genres to help users understand and use the on-screen genres. Designers can make this hybrid genre a site of innovations and change rather than a site of breakdowns. The goal of these solutions, again, is not simply to "fix" hybrid genres, but to highlight their contradictory nature, to turn them into sites of learning and innovation rather than simply sites of breakdowns.
Table 6.8  Contradictions, discoordinations, and breakdowns related to data identifiers.

<table>
<thead>
<tr>
<th><strong>Contradiction</strong></th>
<th>Between the activities of accident location and geographic location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accident location: The data representations in the ALAS activity network</td>
<td></td>
</tr>
<tr>
<td>Geographic location: GIS data representations with which students are familiar</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Discoordinations</strong></th>
<th>Between different genres representing data:</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIS data: Involves conventional terms and categories</td>
<td></td>
</tr>
<tr>
<td>Prototype B's data: Involves cryptic labels and category numbers</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Breakdowns</strong></th>
<th>Students have difficulty interpreting records and identifiers well enough to make sense of the data</th>
</tr>
</thead>
</table>

**Summary: Contradictions, Discoordinations, and Breakdowns Encountered by GIS Students**

Like the previous study of ALAS workers, this study of GIS students suggests that GIS-ALAS' hybrid genres are sites of contradiction. Each hybrid genre is the union of genres that developed separately to support quite different, even contradictory activities; the hybrid genre retains that double orientation toward the activities. So, although students have far more knowledge about GISes than the ALAS workers, they still encounter breakdowns when using these hybrid genres. Although these breakdowns are sometimes so different from the workers' breakdowns as to appear unconnected, we can trace many of them to the same two contradictions that underlie the worker's breakdowns: particularly the contradiction between accident location and geographic location, and the contradiction between standard databases and visualization tools. In
addition, we can trace some of the students' breakdowns to a third contradiction: between accident location and students' activities. These contradictions can lead to discoordinations and breakdowns, but also to innovation and learning, if designers can manage them properly. That's part of the designer's job, I argue. In this section, I've discussed a few ways in which the theory can enable designers to manage contradictions through design. These ways involve more than just reconfiguring the interface, they involve reaching into larger activities (here, training activities).

If these hybrid genres are sites of contradictions (i.e. friction points) between different activity networks, I argue, they can also furnish designers with productive starting points for redesigning the information system. But I also argue that designers might go even further: they might work to redesign the activity in which the information system is used.

**Designing Genres, Designing Activities**

The two studies in this chapter suggest that since hybrid genres embody contradictions, they are suitable starting points for redesign. I've suggested a few ways in which designers can redesign these hybrid genres in ways that harness contradictions. And in the second study I hinted that designers should involve themselves in training as well — designing users' learning activities as well as genres. I want to take up that claim here and apply it more broadly. When possible, designers may even go farther by helping to redesign the activities that the information system mediates. This is one possible answer to Button and Dourish's challenge of connecting description (evaluation) with prescription (design).

The theory, as I've attempted to illustrate in this chapter, may help designers to imagine very different design possibilities, ones that look beyond the interface to the activities in which information systems are used. That is, rather than designing information systems in response to recurrent discoordinations and breakdowns, designers should address the deep contradictions that underlie the symptoms. Otherwise, contradictions can manifest themselves in new ways.
For instance, since the days of mainframe-ALAS, the ALAS tools have embodied a deep contradiction between two ways of representing the roadway system. Each time a new ALAS tool is designed, new ways are devised to bridge the two systems. Mainframe-ALAS involves node maps, forms, and punch cards; PC-ALAS involves node maps and dialog boxes; GIS-ALAS involves the map window. This bridging approach can be quite useful. Yet these bridges are sometimes limited in certain ways because although they address breakdowns and dis coordinations, they do not always address or manage the original contradiction; they do not manage to coordinate the interpenetrating activities of the users (see Chapter 4). In fact, GIS-ALAS goes further by introducing a third (coordinate) system in the map window. But as we saw in this chapter, such bridges are often not guided by a way to systematically recognize and productively use contradictions. Thus such bridges sometimes address the symptoms rather than the disease, the flower instead of the root.

Let's consider how to address the disease and the root. For instance, the theory suggests that instead of relying on the map window to bridge the gap, designers might decide to ease or harness the contradiction itself. The approach goes beyond information system design to the designing of work; it goes beyond software development to sustained and proactive interactions with management, users, locators, and police officers; it goes beyond addressing symptoms to addressing the contradictions underlying them. Such deep changes require consensus-building throughout the activity network. Below are three possible deep changes that designers might try to effect, in order of scale. Each involves an expanded role for designers, requiring more alliances, resources, and time; each is more politically and logistically complex than the last.

Developing a new conversion program. In place of the simple program currently used to convert node-link data to GIS-ALAS' coordinate system, CTRE's developers could develop a more intelligent program — one that knows enough to plot accidents on roadway systems rather than in cornfields, for instance (Figure 6.8). Such a program is relatively simple and could probably be written by students at the university as part of a
class project; it is firmly within the designers' purview. This solution addresses the contradiction among accident representations by unifying the data representations: by precisely converting from qualitative and node-link systems to the coordinate system used by the GIS, and thus minimizing the discoordinations and breakdowns that result from the contradictions between competing data representations. This measure minimizes the conversion work that users have to perform; it addresses the conversion between data representations at the level of the GIS' data.

Retraining locators. Designers could further address the contradiction by ensuring that new data are located using the GIS' coordinate system. Designers could help retrain the DOT's locators to locate future accidents using the coordinate system that the GIS uses, rather than the current system (in place since 1974). Since the DOT employs only a handful of locators, retraining them may not represent a very large investment in comparison with the potential benefits. In fact, designers could furnish locators with a GIS that lets them place the accidents precisely on the map. Such an innovation may make locators' jobs far easier — and thus faster, more accurate, and more efficient — because they would not have to use the cryptic node-link system at all. This effort would be a joint collaboration between the designers and the DOT's Traffic Safety Division; it would involve consensus-building between these two entities. And by using the coordinate system from the locators' work on up, designers can minimize the discoordinations and breakdowns that normally result from the contradiction between roadway representations. This measure addresses the conversion between data representations at the level of the locators' work.

Encouraging the deployment of GPSes. Finally, guided by the analysis, designers might further address the contradiction among accident representations by encouraging the rapid deployment of global positioning systems (GPSes) in patrol cars. The Mobile Accident Reporting System (MARS) project aims to put a GPS in patrol cars across Iowa in the next decade so that officers can initially locate accidents using the coordinate system, an innovation that will eventually obviate the locators' job altogether. Designers
could encourage this change by providing robust support and testing for GPS-supplied
data, by working closely with officers at test sites, and by touting its advantages to the
users with which they work. This effort would have to be a joint collaboration among the
developers, the DOT, local law enforcement agencies, and state and local authorities and
funding sources; it would involve sweeping consensus-building among all entities, and
may require designers to take part in related advocacy efforts such as studies, reports,
and grants. These efforts can lead to data being captured via the same coordinate
representation that the GIS will use to display it — that is, they could lead to the
narrowing of the different accident representation systems. At this point, data will be
captured, viewed, and searched using only one representation system (the coordinate
system) and the data will not have to be converted among representations at all. This
measure addresses the conversion between data representations at the level of data
capture, and thus at all levels.

If the designers can rally support for accomplishing one or more of these deep
changes, they could replace the node-link system with the coordinate system, a more
familiar way of plotting and locating data on maps. Changes could lead to more
"maplike" map windows, more precise and accurate analyses of accident locations, and
— most importantly — perhaps even more lives saved and more injuries avoided.

Conclusions

At the beginning of this chapter, I asked: If designers replace one information
system's genres by hybridizing them with the genres of another information system, how
do these changes affect dis coordinations and breakdowns? I argue that they affect
dis coordinations and breakdowns a great deal:

(1) Hybrid genres are points at which activity networks interpenetrate, and are thus
sites of contradiction. Each hybrid genre is a combination of two or more genres, genres
that developed in different activity networks to mediate different activities with different
objectives. When the two genres are combined, the resulting hybrid genre retains a
double orientation towards the activities: it attempts to mediate them both, and thus it
embodies the contradictions between them. These contradictions are sites of breakdowns, but also sites of change and innovation.

(2) Since hybrid genres are sites of contradiction and since workers feel the need to re-mediate them, designers can use hybrid genres as starting points for addressing the contradictions embedded in information systems. Designers can address contradictions by redesigning the hybrid genres.

(3) But designers may also go further by helping to redesign the very activity networks that the information systems mediate. That is, since usability is distributed across the entire activity network (as Chapter 3 argues), the designer's work must be distributed also.

Implications for Methodology

Given the above points, I argue that developers should seek opportunities to redesign hybrid genres in such a way as to address underlying contradictions. But when possible, designers may even go farther by helping to redesign the activities that the information system mediates. This is one possible answer to Button and Dourish's challenge of connecting description (evaluation) with prescription (design).

Description (Evaluation)

The two studies suggest that importing off-screen genres into the interface can have enormous repercussions for genre perception and genre management. When genres are introduced into the interface, they are hybridized with existing on-screen genres. The two sets of genres are oriented towards different, even contradictory activities, and the contradictions between those activities become embedded in the hybrid genre. Once the genres are combined, the hybrid is doubly oriented, that is, it embodies the contradiction between the two activities. These contradictions may lead to dis coordinations and breakdowns, both in the short and long term.
The analysis I have employed in this chapter provides a way to describe hybridization and its repercussions for users. Yet description (evaluation) is only part of my task. How can we use that description to prescribe (design) new solutions?

Prescription (Design)
Hybrid genres point to redesign possibilities — by looking at contradictions, we can imagine new ways to design hybrid genres to support users' activities, including learning activities. We should find ways to make contradictions a source of innovation and change.

Prescription, then, begins with description: examining and analyzing the genre ecology and the contradictions, discoordinations, and breakdowns associated with the information system.

(A) Ecological mediation. Designers can map out genre ecologies and the activity networks that they mediate — both currently and historically.

(B) Distributed usability. Designers can then analyze the activity networks for usability issues using the three-level evaluative structure of contradictions, discoordinations, and breakdowns.

Once designers describe the ecological mediation and distributed usability in the relevant activity networks, they can begin to redesign the information system.

(A) Based on the evaluation, designers can examine these contradictions, and the discoordinations and breakdowns that spring from them — focusing in particular on hybrid genres, which tend to embody the contradictions that spring up among intersecting activity networks. Designers can identify ways to address the contradictions and speculate how their consequences might ripple across relevant activity networks.

(B) Using one or more of the most promising avenues, designers can conduct limited design changes, again focusing on the hybrid genres that tend to be sites of contradiction. That is, designers can construct prototypes in which the contradictions are addressed. (Design techniques such as cooperative prototyping and participatory design can be useful here.) Designers can then deploy these prototypes in limited tests and examine
how they affect the activity network, the genre ecology, and the three-level evaluative structure.

(C) Using the most favorable prototype, designers can implement (or at least argue for) fuller changes. Designers can look for new contradictions to form, again keeping a close watch on hybrid genres, which are the most likely to be sites of contradiction. Contradictions are unavoidable, but the point is to identify and manage them.

In the next and final chapter, I argue that these points, taken together, provide a research methodology for information system design and evaluation.
CONCLUSION
DESIGNING FOR LIFEWORLDS

As I have argued throughout this dissertation, many methodologies guiding research into designing and evaluating information systems tend to assume the Cartesian dichotomies I discussed in Chapter 1: mind-world, form-content, situation-history, and text-context. As Chapter 1 argues, these dichotomies limit the ways in which we understand how society, culture, history, and interpretation influence information system use. They can lead to an atomistic examination of information system use, one that attempts to locate use and usability within autonomous minds and artifacts rather than among minds and artifacts.

In Chapter 1, I gave several examples of how Cartesian assumptions lead to methodological limitations: units of analysis that do not systematically take into account the complex social-cultural-historical relations among users, artifacts, and activities; concepts such as affordances that tend to narrow and artificially localize the relations between artifacts and practices; weak frameworks for exploring coherent collections of practices; and an arbitrary separation of form and content that encourages researchers not to see interfaces as semiotically unified. These four methodological limitations constrain research in ways that limit — and sometimes deny — the exploration of social, cultural, historical, and interpretive dimensions of information system use.

Throughout the dissertation I have developed, tested, and refined a research methodology that avoids the Cartesian dichotomies outlined above, one that can examine information systems at macro, meso, and micro levels of analysis. That methodology is built on the theory of artifacts discussed in Chapter 1 and involves both describing (evaluating) and prescribing (designing) information systems. To develop the methodology, I asked the following research questions:

- Where is "usability" located? That is, working within a cultural-historical framework, how can we locate and conceptualize the difficulties that people encounter as they attempt to use information systems?
• How do participants in an activity come to perceive and manage the various
genres used in that activity to mediate their work?

• How do participants come to encounter difficulties in perceiving and managing
the various on- and off-screen genres?

• If information system designers import some of the many off-screen genres into
the interface — transforming them into on-screen genres — how do these
changes affect how users perceive and manage the genres?

Below, I outline the theory of artifacts I presented in Chapter 1, then used as a basis
for the research methods outlined in Chapter 2. I explore how the results of the studies
led me to the methodology I advocate. Finally, I summarize the answers to the research
questions and discuss the implications of those answers for information system design
and evaluation.

Theory of Artifacts

A theory of artifacts is essential to any research methodology for information systems
design and evaluation. Yet most theories of artifacts used for information system design
and evaluation remain unarticulated and many tend to be formalist. In Chapter 1, I
developed and articulated a theory of artifacts meant to avoid the Cartesian assumptions
prevalent in information systems research, particularly the four problems I discussed in
Chapter 1. The theory of artifacts, based on a synthesis of activity theory and Bakhtin's
theory of genre, has five main tenets.

Activities are cyclical and developing. Artifacts derive their meaning and shape
from the activities in which they are used. Since these activities develop and change over
time, so do artifacts. Yet the genre rules that guide how we use artifacts are temporarily
stabilized: they develop slowly enough that their use becomes generalizable in the short
term.

Artifacts are interpreted segments of the material environment. Artifacts are the
confluence of the material environment and the cultural-historical genre rules developed
to deal with that environment within certain activities. That is, artifacts are not separable
from the cultural-historical milieus in which they are encountered. Recognizing an artifact entails recognizing that it belongs in one's lifeworld.

**Artifacts' rules are complex, relatively stable, co-developing, and socially distributed.** The coherent sets of genre rules that cluster around artifacts originate in activities and develop as a coherent unit to address similar activities. Collections of rules provide a sort of social memory, a way for subjects within activity networks to share and stabilize the strategies they have developed for dealing with artifacts.

**Artifacts are instantiations of cultural-historical traditions.** A genre provides a relatively stable set of habits that is easy to produce and interpret. Thus genres are especially important for designing and using artifacts. Genres provide traditions of a sort: common cultural-historical resources on which designers and users can draw.

**Artifacts occupy ecologies.** We sometimes think of activities as mediated by a single artifact. But as we investigate the users' activities, we may find that users employ several genres of artifacts to jointly mediate these activities.

To explore these tenets for information system design and evaluation, I conducted five studies of information system use, particularly concentrating on breakdowns: points at which users find that they have to reinterpret a given artifact if it is to support their work.

**Methods**

Given the constructivist theory of artifacts in Chapter 1, I selected data collection and analysis methods that allowed me to examine artifacts as instantiations of genres, that is, in terms of the cultural-historical clusters of habits used to produce and interpret these artifacts. In particular, the methods allowed me to systematically examine interpretive breakdowns as symptoms of deeper discoordinations and contradictions. The methods allowed me to develop a methodology\(^2\) based on Chapter 1's theoretical assumptions.

\(^{2}\) I use the distinction between method and methodology made by Sandra Harding and adopted by Gesa Kirsch and Patricia Sullivan: A methodology is "the underlying theory and analysis of how research does or should proceed," whereas a method is the set of steps guided by the methodology (Kirsch and Sullivan, 1992, p. 2).
The methods of data collection and analysis, by and large, are drawn from the fields of human-computer interaction and computer documentation — semi-structured interviews, naturalistic observations, questionnaires, video coding — but since my methods were guided by the constructivist theoretical assumptions discussed in Chapter 1, they were executed in quite different ways, ways that emphasized culture, society, history, and interpretation. Throughout, I adapted methods that are consonant with the theory of artifacts, using them in five studies to iteratively develop the research methodology.

Results

The theory of artifacts in Chapter 1, and the methods outlined in Chapter 2, guided the studies in chapters 3-6. Through these studies, I developed a methodology for conducting research into information system design and evaluation. This methodology involves both description (evaluation) and prescription (design) of information systems. Below, I describe the main points of the methodology that emerged, illustrating them with examples of results from the five studies.

**Description: Evaluating Information Systems**

Based on the theory of artifacts, I suggest that designers evaluate information systems — whether they are computerized, as is the case with PC-ALAS and GIS-ALAS, or uncomputerized, as is the case with the pre-automation ALAS activity network — by examining the ecological mediation and the distributed usability of the systems. **Ecological mediation** involves mapping the system's genre ecologies and the activity networks that they mediate — both currently and historically. **Distributed usability** involves analyzing the activity networks for usability issues using the three-level evaluative structure of contradictions, discoordinations, and breakdowns. Together, they help to provide designers with a picture of how genre ecologies mediate the work being done in an activity network.
**Ecological Mediation**

Mediation is a key concept of both activity theory and Bakhtin's genre theory. Yet it is seldom explored in terms of multiple artifacts. As I have argued throughout the dissertation, mediation is *ecological*, that is, compound, developmental, and hybridized. Table K.1 shows the tenets, key concepts, and methods that might be used for investigating the three tenets of compound mediation, ecological development, and hybridization.

**Compound mediation.** Traditionally, those who study mediational means tend to focus on individual mediators. Yet any given activity involves compound mediation: actors draw on a variety of artifacts and types of artifacts, in various configurations, to mediate their work. To analyze compound mediational means, in Chapter 3 I drew on Edwin Hutchins' work to develop the concept of *ecologies of genres*. For instance, I used the notion of genre ecologies to examine the many on- and off-screen genres that collectively mediate the activities of PC-ALAS users (chapters 4 and 5). Whereas traditional studies of mediation might focus on one or a few of these genres, I attempted to examine the complex interrelationships among them, arguing that understanding these interrelationships was vital for evaluating the PC-ALAS information system.

I found that understanding these relationships was indeed vital: PC-ALAS' perceived usability depended on the ecology of tools on which each user could draw. For instance, all users drew upon a variety of other genres when using PC-ALAS; users who had worked with PC-ALAS longer were more likely to use PC-ALAS in conjunction with other genres, and thus were better able to distribute the work of PC-ALAS use temporally and spatially (as Mike did in Chapter 5 by using a Post-It™ note to lessen the conversion effort between road representations). This insight may not have been available had I chosen to use a traditional methodology for studying PC-ALAS use.

**Ecological development.** Ecologies live. That is, genres in an ecology do not exist in a static configuration; they co-develop, interact, mingle, combine, and are imported in and out of ecologies. That development includes *genre perception* and *genre*
management. Those whose work is mediated by genre ecologies perceive genres in terms of their roles in the ecology; their perceptions develop as they gather experience using these genres. Furthermore, genres are managed in their ecologies, either by their users or (in automated systems) by the underlying computer code. As users continue to mediate their activities with an ecology, they learn strategies for managing the ecology's genres. They frequently adapt other genres to help them manage the genres, and in doing so they import the new genres into the ecology.

For instance, in Chapter 4 I traced how one genre ecology, that of accident location and analysis, developed over a period of four decades. The genres in the ecology did not develop in isolation; rather, their development overlapped in complex and important ways. When some genres are removed, altered, or collapsed (hybridized), the entire ecology is affected (see Chapter 6).

Hybridization. When genres are imported into an ecology, particularly an ecology of interface genres, they tend to mingle with the ecology's genres and produce hybrid genres bearing the characteristics — the history, addressivity, and genre relations — of their parents. For instance, in Chapter 6, the node map was combined with the GIS map window to produce a hybrid genre, the GIS-ALAS map window. This hybrid bears characteristics of both its parents. Yet at the same time it embodies tensions between the activity networks from which its parents sprang. Since the hybrid genre embodies these tensions (contradictions), I've found, they are important starting places for evaluation and design.

These three principles call into question the traditional understanding of usability. Usability is typically regarded as a quality of a particular artifact. Yet if mediation truly is compound, developmental, and hybridized, the nature of usability becomes far more complex: usability must be considered to be distributed across the entire activity network, as I argued in Chapter 3.
Distributed Usability

To explore usability in a cultural-historical framework, I turned to Engeström's tripartite categorization of contradictions, discoordinations, and breakdowns. These guided me to methods of data collection and levels of analysis, summarized in Table K.2 along with possible methods to use for examining them. I discussed these analysis types and levels in chapters 4 and 5, and demonstrated possible methods and conceptual units in chapters 4, 5, and 6.

Contradictions. Contradictions are tensions and imbalances among elements of the activity network. These are the moving forces behind disturbances, innovations, change and development of activity networks — including, I found, the development of genres, and hybrid genres in particular (Chapter 6). Contradictions occur at the level of activity; I found that contradictions can be detected through macro-level analysis of an activity network. For instance, one enduring contradiction in the ALAS activity network is that between roadway representations. The traditional, qualitative roadway representation contradicts the computer-based, quantitative representation of the node-link system, and both contradict the coordinate system used by the GIS in Chapter 6. By detecting contradictions between roadway representations, I was able to better evaluate GIS-ALAS and suggest redesign opportunities that took into account culture, society, history, and interpretation — opportunities that may not have presented themselves if I had used other methodologies.

Discoordinations. Discoordinations, which are manifestations of deeper contradictions, are difficulties in managing the joint use of genres in an ecology. They can involve difficulties with genre perception and genre management. That is, users experience discoordinations when they have difficulty perceiving a genre in the same way that designers or other users perceive them, or when they have difficulty using two genres together to perform an action. For instance, in the PC-ALAS era, the contradiction between roadway systems surfaces as discoordinations between the genres representing those systems: the node map and the dialog box (Chapter 5). I found that
users have difficulty in *perceiving* the genres (they have trouble understanding one genre in relation to the other) and *managing* the genres (they have trouble converting from one roadway representation to the other). Such discoordinations, I found, spring from contradictions, and thus provide a starting point for tracing the essential contradictions in a given activity network. At the same time, discoordinations often result in breakdowns, so tracking discoordination can help researchers to locate potential sites of breakdowns and to examine the relations among genres with an eye toward redesign.

Discoordinations generally occur at the level of activity or action and can be found through meso-level comparative analysis of users' genre perception, and genre management. I discuss discoordinations in detail in chapters 5 and 6.

**Breakdowns.** Breakdowns, which arise from discoordinations, are points at which a user finds the present interpretation of an artifact to be inadequate for the task at hand. They can lead the user to re-perceive and re-manage genres. Breakdowns generally occur at the level of operation — that is, an unconscious operation is brought to the level of conscious action by an unexpected focus shift. I found that one can detect breakdowns found through micro-level analysis of interpretive moments, that is, points when users attempt to interpret artifacts. Concentrations of breakdowns, I found, can point to individuals' difficulties with particular artifacts, but taken cumulatively, they can also point to cross-user difficulties with genres. Such concentrations can provide valuable starting places for evaluation and design, I found, because they can help researchers to detect underlying discoordinations and breakdowns.

For instance, in Chapter 5, nine of the 12 observed PC-ALAS users had noticeable difficulty entering input into dialog boxes: they made attempts, then tried to repair those attempts when they were unsuccessful the first time. These difficulties involved how the users entered input. Ten users attempted to enter data using the mouse as they would if they were using Microsoft Windows or MacOS (two operating systems that were more familiar to the users than PC-ALAS' DOS-based interface). Since users consistently encountered breakdowns when attempting this operation, I was able to more fully
explore the operation in terms of the genres that were involved and the discoordinations and contradictions that underlie the users' difficulties.

The tripartite concept of usability (contradictions, discoordinations, and breakdowns) provides a view of mediation suitable for the research methodology I envision. Once designers have evaluated an information system in terms of ecological mediation and distributed usability, they can begin to design new information systems.

**Prescription: Designing Information Systems**

Designers often focus on the obvious problems of information systems, the breakdowns and discoordinations. Yet as I discussed in the previous section, breakdowns and discoordinations are only symptoms of deeper contradictions. Since contradictions engender discoordinations, which give rise to breakdowns, I suggest that designers find ways to ease or even harness the contradictions, letting the results ripple through the information system in the form of decreased discoordinations and breakdowns. A design process based on this approach might involve speculating about the results of addressing contradictions; conducting limited design changes; and implementing fuller changes. Such changes might involve designing activities, not just information systems. Table K.3 shows the three design stages and methods that designers might use during these stages.

**Speculating about the Results of Addressing Contradictions**

Based on the evaluation, designers can examine the contradictions and the discoordinations and breakdowns that spring from them — focusing in particular on hybrid genres, which tend to embody the contradictions that spring up among intersecting activity networks. Designers can identify ways to address the contradictions and speculate how the consequences of their solutions might ripple across relevant activity networks.

For instance, Chapter 4 describes how workers at the Center for Transportation Research and Education (CTRE) designed a geographic information system (GIS), GIS-ALAS, to replace PC-ALAS. Yet they did not examine the contradictions,
discoordinations, and breakdowns associated with PC-ALAS in a systematic way. In Chapter 6 I found that the redesign introduced more contradictions into the activity network, contradictions that resulted in various different discoordinations and breakdowns. The results of the redesign, I found from a study of four workers using GIS-ALAS, may very well ripple across the ALAS activity network: users found that their traditional ways of coordinating on-screen and off-screen genres had to be rearranged and rethought, and they often did not know where to start. In addition, GIS-ALAS' hybrid genres presented problems in perception and management, both for the ALAS workers and for students experienced in GIS use.

In Chapter 6 I suggested that the design process could be quite different. Designers redesigning GIS-ALAS (Chapter 6), I suggested, could begin by cataloguing the contradictions in the ALAS activity network, particularly in terms of the hybrid genres (such as the map window and report window genres) used by GIS-ALAS. The designers could then speculate about the possible results of addressing these contradictions in certain ways. For example, the designers could consider addressing one contradiction — among the three roadway systems — by reducing the three representations to one system via converting the existing data and making sure new data are entered in a certain format. But the designers would have to think about the consequences ecologically, in terms of how these might engender new contradictions or aggravate old ones. Would the approach actually address the contradiction, or would it simply bury it? Would the approach create new contradictions in other parts of the activity, such as the locators' job? I found that the theory, methodology, and vocabulary developed throughout the dissertation gave me a framework for asking and exploring such questions.

This design stage relies heavily on the thoroughness and insight of the evaluation that preceded it.

Conducting Limited Design Changes

Once designers have speculated about the possible effects of design changes, they can begin to conduct limited changes and monitor users' behavior to detect whether these
changes do indeed support users' activities. Such an approach, which has strong parallels with participatory design, provides a way for designers and users to work together in fine-tuning the system. In contrast, traditional design methods rarely allow developers to evaluate information systems to the degree that I advocate in this dissertation, and thus developers who rely on these methods tend not to gain the guidance they need if they are to design for lifeworlds.

For example, when CTRE developed GIS-ALAS, the developers did not iteratively prototype the new system: they did not systematically observe users interacting with the system at various stages of development. Instead, the bulk of GIS-ALAS development relied on the developers' understanding of the users; on informal walkthroughs; and on unsystematized opinion-gathering. Consequently, I found that users encountered multiple breakdowns, discoordinations, and contradictions when using GIS-ALAS prototypes (Chapter 6); the GIS-ALAS developers did not envision many of these breakdowns, and lacked the theory and vocabulary to describe the underlying discoordinations and contradictions. Furthermore, designers sometimes actively misjudged the users: I found that the very points at which designers assumed users would encounter few difficulties — hybrid genres such as map windows and reports — tended to be sites of many user difficulties.

The theory and vocabulary, I suggest, provide a framework for conducting more detailed and meaningful iterative design testing. Once designers have identified one or more promising avenues for redesign, they can conduct limited design changes, again focusing on the hybrid genres that tend to be sites of contradiction. That is, designers can construct prototypes in which the contradictions are addressed. (Design techniques such as cooperative prototyping and participatory design can be useful here.) Designers can then deploy these prototypes in limited tests and examine how they affect the activity network, the genre ecology, and the three-level evaluative structure.

For instance, suppose that designers decide to redesign GIS-ALAS, at least partially, by reducing the three roadway representations to one representation. The designers
should conduct a series of limited tests to determine whether their speculations were correct: that is, whether the change really does address the contradiction and whether it avoids introducing new ones. Such tests will involve not only testing the prototype with users, but also testing the altered process with affected personnel. For example, if the designers' change affects personnel such as police officers and locators, they should determine how the changes affect these personnel's work as well.

Implementing Fuller Changes

Once designers and users have worked together to test limited design changes, designers can use the most favorable prototype to implement fuller changes — again, monitoring how these changes support users' activities. Such an approach continues the designer-user partnership and should result in an information system whose contradictions are managed. This approach stands in contrast to much current design work, guided by less culturally and historically oriented theoretical frameworks.

For instance, the GIS-ALAS project was nearly finished at the time of the studies I described in Chapter 6, yet I found that the designers' informal evaluations had missed difficulties that the users would certainly encounter. As I suggest above, GIS-ALAS' developers might have been better able to detect these difficulties if they had a theoretical framework suitable for this sort of work. Yet the developers did not have access to such a framework; rather, they relied on reproducing PC-ALAS genres in the structure of the GIS. I found that this strategy had the opposite of the intended effect: these hybrid genres tended to embody the contradictions between the activities for which PC-ALAS and the GIS were designed. Consequently, I found, the hybrid genres became epicenters for dis coordinations and breakdowns.

As I have found in the studies in chapters 4-6, what we commonly conceive as usability problems are merely the symptoms of deeper contradictions. Thus, designers should strive to address — that is to say, manage, allow for, and sometimes perhaps even ease — the contradictions that underlie users' difficulties. Below, I suggest one way of doing so.
Using the most favorable prototype (produced through iterative prototyping, described above), designers can implement fuller changes, always with careful monitoring of the tool's use. Designers can look for new contradictions to form, again keeping a close watch on hybrid genres, which are the most likely to be sites of contradiction. Contradictions are unavoidable, of course, but the point is to identify and manage them.

For instance, once designers have tested the new GIS-ALAS prototype with the unified roadway representation, they may choose to develop a fuller version and deploy it within selected offices for a longer period of time. They will want to evaluate its use at the various offices, concentrating especially on the hybrid genres that so often embody contradictions, as well as new breakdowns that may indicate other contradictions. Contradictions will inevitably form. Yet if designers are alert, they can manage the contradictions by providing ways to address them and encouraging users to adapt genres for addressing them.

Implications

Earlier in this chapter I reiterated the four research questions that guided my research. Below, I give answers to these questions, using them to explore the implications of these answers for those who research the design and evaluation of information systems.

**Locating Usability**

Where is "usability" located? That is, working within a cultural-historical framework, how can we locate and conceptualize the difficulties that people encounter as they attempt to use information systems?

Usability is customarily seen as a property of a single artifact. This is a logical conclusion given the Cartesian assumptions of much information systems research (see Chapter 1). Yet the study in Chapter 3, based on a non-Cartesian theory of artifacts, suggests that usability is a quality of the entire activity network. That is, it is distributed across the entire activity of people and their tools, not located in the individual tools that
mediate that activity. For instance, developers at the three sites understood the grep utility as having different affordances, uses, and usefulness — not because of the inherent qualities of this tool, but because of how it interacted with the culturally and historically developed rules that had been associated with the tool at different sites. Whether grep was "usable" or not depended on the qualities of the artifact itself, but also on the experience and training of the individual users, the objects the tool was used to transform, the community using the tool, the community's division of labor, the rules that the community had devised for regularly using the tool, and the other tools that were available to the users.

What does this perspective mean to us as we attempt to evaluate information system use? I suggest that it pushes us to spend more time exploring the activities and genre ecologies in which artifacts might be used. Such exploratory studies provide the grounding that is too often lacking in traditional lab-based usability testing, in which individuals are isolated with particular artifacts and asked to perform controlled tasks. This approach goes beyond ecological validity as it is commonly conceived: information system use, rather than something that happens in a larger ecology, is seen as a function of the ecology and inseparable from it. Thus, although traditional lab-based usability testing can be valuable for examining certain micro-level interactions and breakdowns, it makes sense only after researchers establish how people and their genres interact in activity networks.

For instance, when developers go about designing information systems such as databases, websites, or online help, they may do well to start by examining the ways of information gathering that users have already established (as I demonstrate in Chapter 3). By examining users' activities and the ecology of genres they have available, designers can begin to trace out existing intersections and relationships that they can use later when designing information systems that will eventually be introduced into the genre ecology. In doing so, designers can forge partnerships with the users; they can be more confident
that the new information systems will be incorporated into the users' genre ecologies and support their activities.

This leads me to the second implication of the concept of distributed usability. It implies a rethinking of design as essentially collaborative: the designer doesn't dictate what will work, she or he attempts to design something that will support the current ecology and activities (and sometimes, attempts to design the ecology and activities as well). Thus, design involves users and artifacts as well as designers — a philosophy at odds with traditional Cartesian design methodologies, which tend to assume that users are interchangeable and that one design can truly fit all (see Chapter 1). Artifacts should be designed to be ecologically harmonious: they should include the potential to let users add their own innovations, including the ad-hoc adaptation of other genres into the ecology. I further develop this point below.

Developing Genre Ecologies

How do participants in an activity come to perceive and manage the various genres used in that activity to mediate their work?

Chapter 4 suggests that through their activities, people collectively develop ecologies of genres by adapting existing genres to help mediate between other genres in the ecology. In doing so, they learn how to interpret the genre in such a way as to mediate between other genres. Eventually, some of these genres may combine with on-screen genres to produce hybrid genres.

What does the development of genre ecologies mean to us as we design and evaluate information systems? For one thing, I suggest, it gives us clues as to how users understand, use, and supplement information systems. We can find discoordinations among genres by looking for ad hoc genres that users have adopted to mediate among them (see chapters 4, 5, and 6 for examples). We can attempt to standardize those ad hoc genres by importing them into the interface. And we can recognize that any information system we design may not always be adequate for the users' needs — so we should design information systems with enough flexibility that users can easily adopt other
genres. This last solution gives users the freedom to be co-designers: users can influence the artifact, not just through participatory design (a collaboration with the developers), but also through their own (unforeseen, ad hoc) innovations.

For instance, once designers have mapped out the genre ecology into which they hope to introduce an information system, they should explicitly recognize that the information system they design cannot replace the entire genre ecology — rather, users must be encouraged to incorporate it into the ecology. The information system doesn't substitute for the multitude of genres and practices that have developed in conjunction with this activity; rather, it becomes a part of these genres and practices; perhaps incorporates a few; perhaps displaces a few. And it should support later incorporations of other genres: it should be flexible enough that users can mediate its genres with those that they introduce. For example, interactive websites might give users the chance to introduce their own hyperlinks into the structure, pointing to useful relevant sites; databases might give users a chance to build their own additions, note their own practices, incorporate their own demonstrations of helpful techniques, and automate commonly used tasks; online help might offer the opportunity for users to build on, rewrite, or rearrange the text for their needs, or even adapt based on an automated analysis of their work habits (cf. Terveen, Selfridge, and Long, 1995). In these examples, I attempt to give a feeling for the sorts of possibilities that open-ended design might support for users; more specific projects may yield more specific possibilities.

**Examining Usability Issues**

How do participants come to encounter difficulties in perceiving and managing the various on- and off-screen genres?

In Chapter 5, I argue that contradictions among the various activity networks involved in the work lead to discoordinations among the genres in the genre ecology, and eventually to breakdowns. This tripartite structure, I argue, enables us to conduct detailed analyses — analyses that allow us to examine not just the surface usability issues (breakdowns), but also the tensions among genres (discoordinations) and the larger
tensions among activities (contradictions) that underlie them. And if we can indeed conduct these detailed analyses, we can move beyond evaluating artifacts to evaluating the activities in which the artifacts are used. Once we can perform such an evaluation, we can begin to design information systems. The above implies that although designers are not the sole arbiters of how well information systems support activities, designers are in a unique position to observe, analyze, and even shape and guide the larger activities that the information systems support.

For instance, suppose that evaluators are examining an online help system for an HTML editor. Evaluators can use the tripartite structure to develop a multileveled analysis of online help use, one that lets them understand how the online help fits into the users' larger activities and genres (including the HTML editor and webpage production); how individual users use the online help system for performing specific actions, and what genres they use; and which genres (such as pop-up windows and drop-down menus) and artifacts (such as the window defining "Font" and the File menu) are most often associated with interpretive breakdowns. Through such an analysis, evaluators can address usability issues at various levels: not just by minimizing the breakdowns that are normally taken to be usability problems, but also by addressing the dis coordinations and contradictions that underlie them. Such solutions may involve redesigning not just the online help system itself, but also other parts of the genre ecology (the HTML editor, storyboards, the desktop) — and even the activity itself, as I argue below.

**Designing Information Systems — and Their Activity Networks**

If information system designers import some of the many off-screen genres into the interface — transforming them into on-screen genres — how do these changes affect how users perceive and manage the genres?

I argue in Chapter 6 that when genres are introduced into the interface, they are hybridized with existing on-screen genres. These on-screen genres are often perceived and managed quite differently from their off-screen counterparts. In the short term, users
may encounter multiple discoordinations and breakdowns because they have to find new ways to perceive and manage the hybridized genres. In the long term, contradictions between the old (off-screen) and new (on-screen) systems may result in discoordinations because the hybrid's parents have been perceived and managed in quite different ways. These discoordinations may give rise to breakdowns.

What impact do these insights have on information system design? I suggest that information system designers must become aware that hybrid genres are meeting points of different activity networks, and therefore sites of contradiction. How can designers understand, predict, and manage or harness such contradictions? How can they push beyond designing hybrid genres and genre ecologies, and actually begin to affect the activity networks in which the information systems are used? Can designers go beyond designing for lifeworlds and actually help to design lifeworlds? I have attempted to outline answers to these questions above, but they need further research — research that involves the full-scale design of an information system according to the principles laid out in this dissertation.

Now that it's developed, the methodology should be flexible enough to be applied to broader concerns involving on-screen and off-screen genres. These concerns might include website development, applications development (software engineering), document design, and even the deployment of technologies outside the computer interface.

Conclusion

The theory of artifacts forms the foundation of the research methodology I propose. If we want to test and craft information systems that genuinely support and facilitate users' everyday activities, we can use a methodology such as this one, a methodology that rejects the Cartesian dualisms under which designers have traditionally labored. Throughout this dissertation and the five empirical studies it describes, I have attempted to articulate, test, and demonstrate the essentials of such a methodology, one that is suitable for designing for lifeworlds.
APPENDIX A
INTERVIEW QUESTIONS FOR PILOT STUDY

The following questions were asked by the researcher in the pre-observation interview.

**Background**

- What development environment are you using? (Unix, Windows, etc.)
- What are you working on, and how long have you been working on it?
- How long have you been working with C/C++?
- How long have you been working with Schlumberger’s C/C++ libraries?
- Which libraries are you using the most today? In this project? In general?

**Documentation needs**

- How do you usually get information about Schlumberger’s libraries? (e.g. asking other developers, examining the libraries, reading the documentation.)

Under what circumstances might you use each avenue of gathering information?
- What are the advantages of getting information each way? The disadvantages?
- What types of documentation or reference materials have you used while programming in this language? About how long have you used these? What are their advantages and disadvantages?
APPENDIX B

QUESTIONNAIRES FOR PC-ALAS AND GIS-ALAS USERS

Questionnaire

Instructions: The information in this section is designed to help us better understand your employment and education. Please take a few moments to answer the following questions.

Sex: (Circle one) Male Female

Age: 

Education

| Highest degree you received, or degree you are currently pursuing (please circle): |
|---|---|---|---|---|---|
| High School | Associate's | Bachelor's | Master's | Ph.D |

Institution rewarding degree:

Major:

Year of graduation or anticipated graduation:

Experience with accident location

At work: _____ months

At school: _____ months

Which of the following have you engaged in? How often?

<table>
<thead>
<tr>
<th>Activity</th>
<th>Frequency (A=Very Frequent, E=Never)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examining statistics on accident/crash locations</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Creating reports based on traffic statistics</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Using traffic flow information (including accidents) for your school or work duties</td>
<td>A B C D E</td>
</tr>
</tbody>
</table>
## Computer Experience

<table>
<thead>
<tr>
<th>Platform</th>
<th>Years of experience</th>
<th>Level of comfort using this platform (A=very comfortable, C=neutral, E=very uncomfortable)</th>
<th>Level of expertise using this platform (A=expert, C=average, E=novice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM PCs</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Macintoshes</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Other (write here):</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Other (write here):</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Other (write here):</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
</tbody>
</table>

## Operating System

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Years of experience</th>
<th>Level of comfort using this OS (A=very comfortable, C=neutral, E=very uncomfortable)</th>
<th>Level of expertise using this OS (A=expert, C=average, E=novice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOS</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Windows 3.1</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Windows 95</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Windows NT</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>MacOS</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Unix</td>
<td></td>
<td>A B C D E</td>
<td>A B C D E</td>
</tr>
<tr>
<td>Type of Program</td>
<td>Years of experience</td>
<td>Platform (PC, Mac, other)</td>
<td>Level of comfort using this type of program (A=very comfortable, C=neutral, E=very uncomfortable)</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database (Access, dBase, FoxPro, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>Spreadsheet (Lotus 123, Excel, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>Statistical (Minitab, SPSS, SAS, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>GIS (ArcView, Mapintude, MapInfo, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>ArcExplorer</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>Other GIS-based programs (Please list here:)</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accident-tracking software besides ALAS (Please list here:)</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALAS (all versions)</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Mainframe ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>• PC-ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>• GIS-ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>• Explorer-ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
</tbody>
</table>

| Web browsers (Netscape, Internet Explorer, Lynx, etc.) | A | B | C | D | E | A | B | C | D | E |

| CAD programs (AutoCAD) | A | B | C | D | E | A | B | C | D | E |
PC-ALAS

Based on your knowledge of PC-ALAS, please circle the appropriate answer. If you have no knowledge of PC-ALAS, leave this section blank.

For what I do, PC-ALAS seems...

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very useful</td>
<td>Not useful at all</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compared to other accident-tracking software that I've used, PC-ALAS seems...

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very useful</td>
<td>Not useful at all</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compared to other PC-based software I've used, PC-ALAS seems...

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very useful</td>
<td>Not useful at all</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rank the following actions in PC-ALAS, where A=very easy, C=average, E=very hard.

<table>
<thead>
<tr>
<th>Action</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entering data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrieving node data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualizing data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selecting data nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comparing data nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Making queries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyzing accident data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating reports</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Comments

Please feel free to write any comments about ALAS here.
APPENDIX C
INSTRUCTIONS FOR PC-ALAS RATERS

VIDEOCODING SCHEME
Watch the designated spots on the videotapes and make entries for the following categories:

- artifacts
- actions
- operations
- breakdowns

These categories are explained below. I have indicated typical responses for each category; if you want to include additional or modified responses, please indicate these in your coding sheets. Each entry below is marked with a red two- or three-letter mnemonic designation: if you find these useful you can code using them rather than typing out the full names. The mnemonics are optional.

Each entry includes the following:

**ARTIFACT**: The segment of the interface that is involved in the current action. Use these terms to describe the basic types of artifacts:

<table>
<thead>
<tr>
<th>HM</th>
<th>Horizontal menu (at top of screen)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>Vertical menus</td>
</tr>
<tr>
<td>OH</td>
<td>Online help windows</td>
</tr>
<tr>
<td>BK</td>
<td>PC-ALAS background</td>
</tr>
<tr>
<td>RP</td>
<td>Report windows</td>
</tr>
<tr>
<td>DE</td>
<td>Data entry dialog boxes (in which users enter data)</td>
</tr>
<tr>
<td>MS</td>
<td>Message dialog boxes (in which users read a message and click &quot;OK&quot;)</td>
</tr>
</tbody>
</table>

**ACTION**: A conscious goal-directed activity. Actions are associated with artifacts. For instance, filling out a specific dialog box may take several actions: entering a county number, entering a city number, selecting options, and accepting the changes that one has made to a dialog box. I've used very specific actions in my coding, but you can use the following simplified actions:

<table>
<thead>
<tr>
<th>Menus (Horizontal and vertical)</th>
<th>Report Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>Select menu item</td>
</tr>
<tr>
<td>RH</td>
<td>Review online help</td>
</tr>
<tr>
<td>CH</td>
<td>Close online help</td>
</tr>
<tr>
<td>PC-ALAS Background</td>
<td>ET Enter text/number into a field</td>
</tr>
<tr>
<td>BK</td>
<td>Close menus (This is the only reason users clicked on the background.)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OPERATION**: The automatized, habitualized strategies that are used to complete or fulfill the actions noted above. For instance, when entering data into a text field of a dialog box, users can (a) click on the text field with the mouse, then type; (b) click on the text field's label and type; or (c) tab into the text field with the Tab key, then type.
Operations aren't necessarily linked to certain artifacts or actions. They're avenues or habitualized strategies for carrying out actions. One user may only have one operation in her repertoire for dealing with text fields; another may have two or three -- or more.

Operations include the following.

<table>
<thead>
<tr>
<th>Navigating</th>
<th>Entering Data</th>
<th>Controlling Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAK Type arrow key</td>
<td>CCB Click on checkboxes/radio button or their labels</td>
<td>CMI Click on menu item</td>
</tr>
<tr>
<td>CSB Click on scrollbar/scroll box/scroll arrows</td>
<td>CTF Click on text/number field or its label</td>
<td>OKB Click &quot;OK&quot; button</td>
</tr>
<tr>
<td>KEY Tab into field (NOTE: This is the same code as KEY in the other two columns)</td>
<td>CFN Click on filename</td>
<td>CAN Click &quot;Cancel&quot; button</td>
</tr>
<tr>
<td>STB Shift-tab back into field</td>
<td>CB? Click some other button</td>
<td>DIS Click dismiss button (top left corner of dialog box or report)</td>
</tr>
<tr>
<td></td>
<td>BAC Back over data (Backspace key)</td>
<td>BKG Click on PC-ALAS background</td>
</tr>
<tr>
<td></td>
<td>SFM Select field text with mouse</td>
<td>F3B Click &quot;F3 Close Report&quot; button (at bottom of screen)</td>
</tr>
<tr>
<td></td>
<td>KEY Type on keyboard, other than to enter data into a text/number field (NOTE: This is the same code as KEY in the other two columns)</td>
<td>KEY Key combination (used for opening menus or closing reports)</td>
</tr>
</tbody>
</table>

BREAKDOWN: An unexpected focus shift. When coding breakdowns, note focus shifts that cause the user to look at rather than through the interface -- for instance, a user who has trouble scrolling an on-screen report might focus on the report's controls rather than on the data in the report. This is the most complicated part of the coding: it doesn't happen that often, but you have to be alert for it.

<table>
<thead>
<tr>
<th>Breakdown</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON Confused about report results</td>
<td>Voices confusion about a report or the lack of a report.</td>
</tr>
<tr>
<td>DIS Dialog box doesn't dismiss</td>
<td>Clicks OK button or dismiss button (in upper left corner of dialog box), but dialog box does not dismiss.</td>
</tr>
<tr>
<td>DNR Dialog box not relevant/desired</td>
<td>Clicks Cancel button in dialog box</td>
</tr>
<tr>
<td>INC Entered incorrect data</td>
<td>Clicks on checkbox to select, then clicks again to deselect.</td>
</tr>
<tr>
<td>SCR Error when scrolling report</td>
<td>Clicks on radio button to select, then clicks on another radio button instead.</td>
</tr>
<tr>
<td>SCR Error when scrolling report</td>
<td>Types into a text/number field, then backs over it.</td>
</tr>
<tr>
<td>SCR Error when scrolling report</td>
<td>Clicks scroll arrow without effect (for instance, clicks down arrow when she is already at bottom of report window).</td>
</tr>
<tr>
<td>SCR Error when scrolling report</td>
<td>Clicks scrollbar instead of scroll arrow. Indicated by comments, e.g. &quot;oops!&quot;</td>
</tr>
<tr>
<td>INP Input not entered</td>
<td>Clicks or tries to click on a checkbox or radio button; the item is not selected so the user has to click again.</td>
</tr>
<tr>
<td>INP Input not entered</td>
<td>Clicks or tries to click on a text/number field and types; no text</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CRA PC-ALAS crashes</td>
<td>• PC-ALAS becomes inactive or unresponsive.</td>
</tr>
<tr>
<td></td>
<td>• PC-ALAS exits without the user issuing an Exit command.</td>
</tr>
<tr>
<td>HLP Relevant help topic not found</td>
<td>Closes online help</td>
</tr>
<tr>
<td>WRO Selected wrong text or number field</td>
<td>Clicks on text/number field in dialog box, but does not enter text before clicking on a different item.</td>
</tr>
<tr>
<td>SYS System reports input error</td>
<td>Error dialog box appears.</td>
</tr>
<tr>
<td>VMX Vertical menu does not appear</td>
<td>Selects a menu item in the horizontal menu, but a vertical menu does not appear (usually because a dialog box is still active on the screen).</td>
</tr>
<tr>
<td>VMC Vertical menu does not close</td>
<td>Selects item from a vertical menu, but the menu stays open.</td>
</tr>
<tr>
<td>VMO Vertical menu does not stay open</td>
<td>Selects a menu item in the horizontal menu, but the vertical menu only opens for a moment — it closes before the user can make a choice.</td>
</tr>
<tr>
<td>VMR Vertical menu not relevant/desired</td>
<td>Opens a vertical menu, then goes to a different menu.</td>
</tr>
</tbody>
</table>
**VIDEOTAPE TABLE OF CONTENTS**

The table below gives you a "table of contents" showing whether to code the session; *approximately* where each session starts (within a minute — your VCR's mileage may vary); how long to code it; and what user is being coded. Make sure to start at the beginning of each session.

### Tape 1

<table>
<thead>
<tr>
<th>Code?</th>
<th>Start</th>
<th>Duration</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0:00</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>26:26</td>
<td></td>
<td>P13</td>
</tr>
<tr>
<td>Y</td>
<td>38:14</td>
<td>8</td>
<td>P4</td>
</tr>
<tr>
<td></td>
<td>50:00</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td>Y</td>
<td>1:17:00</td>
<td>8</td>
<td>P16</td>
</tr>
<tr>
<td></td>
<td>2:14:38</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>2:24:00</td>
<td></td>
<td>P17</td>
</tr>
</tbody>
</table>

### Tape 2

<table>
<thead>
<tr>
<th>Code?</th>
<th>Start</th>
<th>Duration</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0:00</td>
<td>P17 (continued)</td>
<td>P18</td>
</tr>
<tr>
<td></td>
<td>9:54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>14:40</td>
<td>8</td>
<td>P19</td>
</tr>
<tr>
<td></td>
<td>42:48</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td>Y</td>
<td>1:25:56</td>
<td>3 (entire session)</td>
<td>P22 (with P21)</td>
</tr>
<tr>
<td></td>
<td>1:39:09</td>
<td></td>
<td>P24</td>
</tr>
<tr>
<td>Y</td>
<td>1:40:42</td>
<td>5 (entire session)</td>
<td>P25</td>
</tr>
<tr>
<td></td>
<td>2:16:19</td>
<td></td>
<td>P23</td>
</tr>
</tbody>
</table>

### Tape 3

<table>
<thead>
<tr>
<th>Code?</th>
<th>Start</th>
<th>Duration</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0:00</td>
<td>P28: GIS-ALAS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5:28</td>
<td>P29: GIS-ALAS</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>16:56</td>
<td>8</td>
<td>P27 (help from P26)</td>
</tr>
<tr>
<td></td>
<td>31:14</td>
<td></td>
<td>P26 (help from P27): GIS-ALAS</td>
</tr>
</tbody>
</table>
## APPENDIX D

### PC-ALAS GROUPS

**Table D.1** The Interview-Only group. Italicized numbers are probably inaccurate. Blank spaces indicate incomplete data; averages are calculated without them.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Sex</th>
<th>Age</th>
<th>Interpenetrating activity</th>
<th>MS Windows experience (Years)</th>
<th>PC-ALAS experience (Years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3</td>
<td>M</td>
<td>47</td>
<td>City/county engineering</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>P4</td>
<td>M</td>
<td>50</td>
<td>City/county engineering</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>P5</td>
<td>M</td>
<td>50</td>
<td>City/county engineering</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>P6</td>
<td>M</td>
<td>46</td>
<td>City/county engineering</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>P7</td>
<td>M</td>
<td>25</td>
<td>City/county engineering</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>P8</td>
<td>M</td>
<td></td>
<td>Traffic safety</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>P9</td>
<td>M</td>
<td>37</td>
<td>City/county engineering</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>P10</td>
<td>F</td>
<td>35</td>
<td>Traffic safety</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>P11</td>
<td>M</td>
<td>44</td>
<td>City/county engineering</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>P20</td>
<td>M</td>
<td>45</td>
<td>Law enforcement</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td></td>
<td></td>
<td></td>
<td>42.1</td>
<td>4.2</td>
</tr>
</tbody>
</table>

**Table D.2** The Observation group. Italicized numbers are probably inaccurate. Blank spaces indicate incomplete data; averages are calculated without them.

<table>
<thead>
<tr>
<th>User</th>
<th>Sex</th>
<th>Age</th>
<th>Interpenetrating Activity</th>
<th>MS Windows experience (Years)</th>
<th>PC-ALAS experience (Years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P4</td>
<td>M</td>
<td>50</td>
<td>City/county engineering</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>P13</td>
<td>F</td>
<td>31</td>
<td>Traffic safety</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>P16</td>
<td>F</td>
<td>56</td>
<td>Traffic safety</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>P17</td>
<td>M</td>
<td>47</td>
<td>Traffic safety</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>P18</td>
<td>M</td>
<td>33</td>
<td>City/county engineering</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>P19</td>
<td>F</td>
<td>23</td>
<td>Traffic safety</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>P22</td>
<td>M</td>
<td>23</td>
<td>City/county engineering</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>P23</td>
<td>M</td>
<td>44</td>
<td>City/county engineering</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>P24</td>
<td>F</td>
<td>32</td>
<td>Law enforcement</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>P25</td>
<td>M</td>
<td>33</td>
<td>City/county engineering</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>P27</td>
<td>M</td>
<td>20</td>
<td>City/county engineering</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>P32</td>
<td>M</td>
<td></td>
<td>Law enforcement</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td></td>
<td></td>
<td></td>
<td>35.6</td>
<td>4.2</td>
</tr>
</tbody>
</table>


Figure E.1 Action sequences as a percentage of each observation. Participants varied greatly in proportion of time spent on each action sequence.
Table E.1 Breakdowns by frequency.

<table>
<thead>
<tr>
<th>Breakdown</th>
<th>P4</th>
<th>P13</th>
<th>P16</th>
<th>P17</th>
<th>P18</th>
<th>P22</th>
<th>P23</th>
<th>P24</th>
<th>P25</th>
<th>P27</th>
<th>P32</th>
<th>Total breakdowns</th>
<th>Total actions</th>
<th>% of breakdowns to actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confused about report results</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>25.00</td>
</tr>
<tr>
<td>Dialog box doesn't dismiss</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>3</td>
<td>25.00</td>
</tr>
<tr>
<td>Dialog box not relevant/desired</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>23</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>35</td>
<td>6</td>
<td>50.00</td>
</tr>
<tr>
<td>Entered incorrect data</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21</td>
<td>5</td>
<td>41.67</td>
</tr>
<tr>
<td>Error when scrolling report</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>16.67</td>
</tr>
<tr>
<td>Input not entered</td>
<td>1</td>
<td>2</td>
<td>36</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td>54</td>
<td>9</td>
<td>75.00</td>
</tr>
<tr>
<td>PC-ALAS crashes</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>10</td>
<td>33.33</td>
</tr>
<tr>
<td>Relevant help topic not found</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>8.33</td>
</tr>
<tr>
<td>Selected wrong text or number field</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td>8.33</td>
</tr>
<tr>
<td>System reports input error</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>13</td>
<td>58.33</td>
</tr>
<tr>
<td>Unexpected search results</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>16.67</td>
</tr>
<tr>
<td>Vertical menu does not appear</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>16.67</td>
</tr>
<tr>
<td>Vertical menu does not close</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td>8.33</td>
</tr>
<tr>
<td>Vertical menu does not stay open</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>16.67</td>
</tr>
<tr>
<td>Vertical menu not relevant/desired</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17</td>
<td>8</td>
<td>66.67</td>
</tr>
<tr>
<td>Total breakdowns</td>
<td>10</td>
<td>3</td>
<td>64</td>
<td>6</td>
<td>1</td>
<td>44</td>
<td>4</td>
<td>11</td>
<td>6</td>
<td>12</td>
<td>5</td>
<td>14</td>
<td>180</td>
<td>10</td>
</tr>
<tr>
<td>Total actions</td>
<td>116</td>
<td>36</td>
<td>457</td>
<td>148</td>
<td>17</td>
<td>624</td>
<td>38</td>
<td>91</td>
<td>33</td>
<td>51</td>
<td>67</td>
<td>67</td>
<td>1745</td>
<td>10</td>
</tr>
<tr>
<td>% of breakdowns to actions</td>
<td>8.6</td>
<td>8.3</td>
<td>14.</td>
<td>4.0</td>
<td>5.x</td>
<td>7.0</td>
<td>10.</td>
<td>12.</td>
<td>18.</td>
<td>23.5</td>
<td>7.4</td>
<td>20.8</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>
#### Table E.2 Breakdowns by genre.

<table>
<thead>
<tr>
<th>Breakdown</th>
<th>Horizontal menu</th>
<th>Vertical menu</th>
<th>Dialog box</th>
<th>Message window</th>
<th>Report window</th>
<th>Online help</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confused about report results</td>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>Dialog box doesn't dismiss</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Dialog box not relevant/desired</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>Entered incorrect data</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>Error when scrolling report</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Input not entered</td>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>54</td>
</tr>
<tr>
<td>PC-ALAS crashes</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Relevant help topic not found</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Selected wrong text or number</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>field</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>System reports input error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Unexpected search results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Vertical menu does not appear</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Vertical menu does not close</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Vertical menu does not stay open</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Vertical menu not relevant/desired</td>
<td>16</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>20</strong></td>
<td><strong>6</strong></td>
<td><strong>120</strong></td>
<td><strong>23</strong></td>
<td><strong>10</strong></td>
<td><strong>1</strong></td>
<td><strong>180</strong></td>
</tr>
</tbody>
</table>
APPENDIX F

QUESTIONNAIRE FOR STUDENT PARTICIPANTS

Questionnaire

Instructions: The information in this section is designed to help us better understand your experience with GIS and with related activities. Please take a few moments to answer the following questions.

Sex: (Circle one)   Male   Female

Age:

Education

Degree you are currently pursuing (please circle):

Bachelor's  Master's  Ph.D.

Major:

Year of graduation or anticipated graduation:

Experience with accident location

Which of the following have you engaged in? How often?

<table>
<thead>
<tr>
<th>Activity</th>
<th>Frequency (A=Very Frequent, E=Never)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examining statistics on accident / crash locations</td>
<td>A  B  C  D  E</td>
</tr>
<tr>
<td>Creating reports based on traffic statistics</td>
<td>A  B  C  D  E</td>
</tr>
<tr>
<td>Using traffic flow information (including accidents) for your school or work duties</td>
<td>A  B  C  D  E</td>
</tr>
</tbody>
</table>
### Computer Experience

<table>
<thead>
<tr>
<th>Platform</th>
<th>Years of experience</th>
<th>Level of comfort using this platform (A=very comfortable, C=neutral, E=very uncomfortable)</th>
<th>Level of expertise using this platform (A=expert, C=average, E=novice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM PCs</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Macintoshes</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Other (write here)</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Other (write here)</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Other (write here)</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
</tbody>
</table>

### Operating System

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Years of experience</th>
<th>Level of comfort using this OS (A=very comfortable, C=neutral, E=very uncomfortable)</th>
<th>Level of expertise using this OS (A=expert, C=average, E=novice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOS</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Windows 3.1</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Windows 95</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Windows NT</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>MacOS</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Unix</td>
<td>A B C D E</td>
<td>A B C D E</td>
<td></td>
</tr>
<tr>
<td>Type of Program</td>
<td>Years of experience</td>
<td>Platform (PC, Mac, other)</td>
<td>Level of comfort using this type of program (A=very comfortable, C=neutral, E=very uncomfortable)</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database (Access, dBase, FoxPro, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>Spreadsheet (Lotus 123, Excel, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>Statistical (Minitab, SPSS, SAS, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>GIS (ArcView, Mapintude, MapInfo, etc.)</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>ArcExplorer</td>
<td></td>
<td></td>
<td>A B C D E</td>
</tr>
<tr>
<td>Other GIS-based programs (Please list here:)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Accident-tracking software (Please list here:)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>ALAS (all versions)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Mainframe ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>• PC-ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>• GIS-ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>• Explorer-ALAS</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Web browsers (Netscape, Internet Explorer, Lynx, etc.)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>CAD programs (AutoCAD)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
GIS Experience

<table>
<thead>
<tr>
<th>GIS</th>
<th>Months or years of experience</th>
<th>Level of comfort using this GIS (A=very comfortable, C=neutral, E=very uncomfortable)</th>
<th>Level of expertise using this GIS (A=expert, C=average, E=novice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArcView</td>
<td></td>
<td>AB C D E</td>
<td>AB C D E</td>
</tr>
<tr>
<td>MapIntude</td>
<td></td>
<td>AB C D E</td>
<td>AB C D E</td>
</tr>
<tr>
<td>MapInfo</td>
<td></td>
<td>AB C D E</td>
<td>AB C D E</td>
</tr>
<tr>
<td>Other (write here):</td>
<td></td>
<td>AB C D E</td>
<td>AB C D E</td>
</tr>
</tbody>
</table>

What school or workplace tasks have you used the above GISes for?

<table>
<thead>
<tr>
<th>GIS</th>
<th>School or work? (Circle one)</th>
<th>Briefly describe tasks that you have used this GIS for.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>school / work</td>
<td></td>
</tr>
<tr>
<td></td>
<td>school / work</td>
<td></td>
</tr>
<tr>
<td></td>
<td>school / work</td>
<td></td>
</tr>
<tr>
<td></td>
<td>school / work</td>
<td></td>
</tr>
<tr>
<td></td>
<td>school / work</td>
<td></td>
</tr>
<tr>
<td></td>
<td>school / work</td>
<td></td>
</tr>
</tbody>
</table>

Comments

*Please feel free to write any comments about GISes or accident location here.*
APPENDIX G
INSTRUCTIONS FOR GIS-ALAS RATER

GIS-ALAS VIDEOCODING SCHEME
Watch the designated spots on the videotapes and make entries for the following categories:

- artifacts
- actions
- operations
- breakdowns

These categories are explained below. I have indicated typical responses for each category; if you want to include additional or modified responses, please indicate these in your coding sheets. Each entry below is marked with a (red) two- or three-letter mnemonic designation: if you find these useful you can code using them rather than typing out the full names. The mnemonics are optional.

NOTE: For simplicity, the following special cases are handled differently from other entries:

- **Looking at menu items.** If the participant views several vertical menus but selects a menu item from only one, code the series as one entry and note in the margin that the entry covers multiple items. If the participant ends up choosing from none of the menus, don’t code it at all — just make a note.

- **Clicking on the map with no recognizable effect.** If the participant clicks on the map and the map doesn’t change (zoom in, zoom out) and an Information window doesn’t appear, do not code the entry. Write a note in the margin instead.

Each entry includes the following:

**Artifact:** The segment of the interface that is involved in the current action. Use these terms to describe the basic types of artifacts:

<table>
<thead>
<tr>
<th>HM</th>
<th>BB</th>
</tr>
</thead>
<tbody>
<tr>
<td>HM</td>
<td>Horizontal menu (at top of screen)</td>
</tr>
<tr>
<td>VM</td>
<td>Vertical menus</td>
</tr>
<tr>
<td>BG</td>
<td>GIS-ALAS background</td>
</tr>
<tr>
<td>DE</td>
<td>Data entry dialog boxes (in which users enter data)</td>
</tr>
<tr>
<td>MS</td>
<td>Message dialog boxes (in which users read a message and click &quot;OK&quot;, or select from buttons &quot;YES&quot; and &quot;NO&quot;)</td>
</tr>
<tr>
<td>OT</td>
<td>Other artifacts not in this list (note these)</td>
</tr>
</tbody>
</table>


**ACTION**: A conscious goal-directed activity. Actions are associated with artifacts. Use the following actions:

<table>
<thead>
<tr>
<th>Menus (Horizontal and vertical)</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI Select menu item</td>
<td>SW Select window</td>
</tr>
<tr>
<td>CM Close menus by clicking elsewhere</td>
<td>MV Move window</td>
</tr>
<tr>
<td>Dialog Boxes (Data entry DBs; message DBs)</td>
<td>RS Resize window, panel, or table column</td>
</tr>
<tr>
<td>ET Enter text/number into a field</td>
<td>SC Scroll window or panel</td>
</tr>
<tr>
<td>SD Select drop-down item</td>
<td>XW Close window with dismiss button (the x in the top right corner)</td>
</tr>
<tr>
<td>AD Accept dialog box</td>
<td>SA Select accident(s) from map</td>
</tr>
<tr>
<td>CD Cancel dialog box</td>
<td>SI Select non-accident item from window (icon, text)</td>
</tr>
<tr>
<td>Button Bar</td>
<td>ST Select/deselect themes</td>
</tr>
<tr>
<td>SB Select button from bar</td>
<td>ZM Zoom in/out (map)</td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>OT Other</td>
<td></td>
</tr>
</tbody>
</table>

**OPERATION**: The automatized, habitualized strategies that are used to complete or fulfill the actions noted above. For instance, when zooming in on a spot on a map, users can use the magnifying glass tool to (a) click on the map or (b) select an area on the map.

Operations aren’t necessarily linked to certain artifacts or actions. They’re avenues or habitualized strategies for carrying out actions. One user may only have one operation in her repertoire for dealing with text fields; another may have two or three -- or more.

Operations include the following.

<table>
<thead>
<tr>
<th>Navigating</th>
<th>Controlling Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSB Click on scrollbar/scroll box/scroll arrows</td>
<td>CMI Click on menu item</td>
</tr>
<tr>
<td>SAR Select area</td>
<td>OKB Click &quot;OK&quot; button</td>
</tr>
<tr>
<td>CMP Click on map</td>
<td>CAN Click &quot;Cancel&quot; button</td>
</tr>
<tr>
<td></td>
<td>COB Click on other button in window or window frame (not button bar)</td>
</tr>
<tr>
<td>Entering Data</td>
<td>CBB Click on button bar</td>
</tr>
<tr>
<td>CDD Click on drop-down selector or selection box</td>
<td>CBG Click on GIS-ALAS background</td>
</tr>
<tr>
<td>SFM Select field text with mouse</td>
<td>CIW Click or double-click on item in window (record name, report label, icon)</td>
</tr>
<tr>
<td>CTF Click on text in field, type</td>
<td>CWF Click on window frame or frames within the window; could involve double-clicking, moving, or resizing a window, pane, or table column</td>
</tr>
<tr>
<td>TYP Type over old data</td>
<td></td>
</tr>
<tr>
<td>BAC Back over data (Backspace key); type</td>
<td></td>
</tr>
<tr>
<td>THM Click on themes</td>
<td></td>
</tr>
<tr>
<td>CFN Click or double-click on filename</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Miscellaneous</strong></td>
</tr>
<tr>
<td></td>
<td>MISC Some other operation not covered above.</td>
</tr>
</tbody>
</table>
**BREAKDOWN:** An unexpected focus shift. Breakdowns will be coded solely by participants' comments. They are categorized thus:

<table>
<thead>
<tr>
<th>Breakdown</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART Question about artifact's properties</td>
<td>Participant wants to know whether an artifact has certain properties.</td>
<td>Can this move out?</td>
</tr>
<tr>
<td>AST Assistance needed in interpreting</td>
<td>Participant has trouble interpreting an interface feature.</td>
<td>From -- so these are like the records of the accidents and stuff?</td>
</tr>
<tr>
<td>CLA Requests clarification</td>
<td>Participant requests clarification of researcher’s previous comment.</td>
<td>CS: And find one of these dots and click squarely on it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P: Find one of the dots?</td>
</tr>
<tr>
<td>PRO Procedural question</td>
<td>Participant indicates confusion about the procedure outlined in the instruction sheet.</td>
<td>Is this where you're talking about the layers? Listed on the left side of the map? You just turn them on and off?</td>
</tr>
<tr>
<td>UNX Unexpected behavior</td>
<td>Participant indicates that the software is behaving unexpectedly.</td>
<td>Something not working here? [CS: Ah] Just trying to get back // out?</td>
</tr>
</tbody>
</table>
VIDEOTAPE TABLE OF CONTENTS

The table below gives you a "table of contents" showing whether to code the session; approximately where each session starts (within a minute or two -- your VCR's mileage may vary); and what participant is being coded. Make sure to start at the beginning of each session.

<table>
<thead>
<tr>
<th>Code?</th>
<th>Start</th>
<th>Participant</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:00</td>
<td>P1</td>
<td></td>
</tr>
<tr>
<td>8:26</td>
<td>P2</td>
<td></td>
</tr>
<tr>
<td>17:06</td>
<td>P3</td>
<td></td>
</tr>
<tr>
<td>29:10</td>
<td>P4</td>
<td></td>
</tr>
<tr>
<td>39:18</td>
<td>P5</td>
<td></td>
</tr>
<tr>
<td>54:00</td>
<td>P6</td>
<td></td>
</tr>
<tr>
<td>1:04:39</td>
<td>P7</td>
<td></td>
</tr>
<tr>
<td>1:14:15</td>
<td>P8</td>
<td></td>
</tr>
<tr>
<td>1:21:45</td>
<td>P9</td>
<td></td>
</tr>
<tr>
<td>1:28:47</td>
<td>P10</td>
<td></td>
</tr>
<tr>
<td>1:40:42</td>
<td>P11</td>
<td></td>
</tr>
<tr>
<td>1:55:09</td>
<td>P12</td>
<td></td>
</tr>
<tr>
<td>2:04:17</td>
<td>P13</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX H
GIS-ALAS QUANTITATIVE DATA

Table H.1 Genres and breakdowns for student group. I concentrate on three genres and three breakdown categories (italicized and shaded).

<table>
<thead>
<tr>
<th>Genre</th>
<th>Procedural question</th>
<th>Assistance needed in interpreting</th>
<th>Unexpected behavior</th>
<th>Requests clarification</th>
<th>Question about properties</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify Results Window</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>Report Table Window</td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>Map Window</td>
<td>9</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>Dialog box</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Message Window</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Vertical Menu</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Control Window</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Button Bar</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Horizontal Menu</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>29</strong></td>
<td><strong>23</strong></td>
<td><strong>10</strong></td>
<td><strong>3</strong></td>
<td><strong>2</strong></td>
<td><strong>67</strong></td>
</tr>
</tbody>
</table>
Identifying Accident Locations

Thank you for participating in this user test. As I mentioned earlier, this user test will help the Center for Transportation Research and Education (CTRE) to improve its accident tracking software. The research will also be used for my dissertation.

The software you will use is GIS-ALAS (a customization of ArcView). GIS-ALAS is used by city and county engineers to locate concentrations of traffic accidents with the goal of finding trouble spots and developing ways to deal with them. For instance, if a certain intersection has a high concentration of accidents involving left turns, engineers might determine that the intersection needs a left turn arrow or a turn lane.

In this test, you will use GIS-ALAS to locate traffic accidents in Story County and examine specific accidents. *This test is not timed*—I'm interested in how you interact with the software, not in how quickly you can accomplish the tasks. Please follow the instructions below.

*Select accidents to examine.*

The map shows primary, secondary, and municipal roads; hydrology (bodies of water); and accidents from 1991-1995. Accidents are marked "Co. 85 Yr. 9x A Records," where \(x\) is the year of the accidents.

1. In the map window, turn off the layers of accident indicators by unchecking the box next to each year's accident records.
2. Select *Run Location* from the *Request* menu.
3. Select *Mile Pointed Request* and click *OK*.
4. Select \(B_\text{(Traffic_Both_Direction)}\).
5. Enter 030 for the route number (that is, Highway 30) and click *OK*.
6. Enter 000 for the beginning milepoint. Click *OK*. 
7. Enter 9900 for the ending milepoint. Click OK. GIS-ALAS will search files and announce how many accidents occurred on 30 in 1990-1995. The accident locations should appear on the map.

*Examine the map of Story County.*

Take some time to look at the concentrations of accidents showing up on the map. If you like, you may turn layers on or off by clicking the checkbox for each. (Layers are listed on the left side of the map.)

Now, click on the Zoom In tool in the button bar (the magnifying glass with the plus sign). Zoom in closer to examine a cluster of accidents.

*Examine one of the accidents.*

Take a closer look at one of the accidents.

1. Click on the Information Tool in the button bar (The "i" in the circle).
2. Using the Information Tool, click on one of the accidents. Information on this accident location should appear in a separate window.
3. Examine the accident records by clicking on each A record. Each A record contains information on a different accident. Determine the number of vehicles and the property damage for each accident.

*Generate a report based on these accidents.*

To create a report, select Engineering Report from the Report menu. This will create summary sheets on accidents occurring on Highway 30. The summary sheets will appear in the window marked av_alas.apr.

Move to the window marked av_alas.apr and double-click on some of the summary sheets. Each summary sheet that you double-click will bring up a window with a table of accident statistics. Examine one of these tables and tell me what it appears to be saying.

Thanks again for participating in this test. Feel free to ask me if you have questions.

*Clay Spinuzzi - spinuzzi@iastate.edu - 294-9325 - 206 Ross Hall*
A wide variety of geographic data is now available on the Internet. The Internet also allows for file transfers (FTP) between individual Internet accounts. This is especially useful because geographic data files can be large and it is inefficient to transfer these files by diskette. File compression formats (tar, pkzip, etc.) can be used to make file transfers (and storage) less cumbersome. In addition, geographic data not currently in MapInfo format will need to be converted using a conversion program like ArcLink.

Download Department of Natural Resources (DNR) data files

1. Within Windows Explorer create a new subdirectory called c:\mytemp to save the files you will be downloading. 2. Using WSFTP (on start menu: Programs > Internet Programs > WS_FTP > WS_FTP LE), connect to the DNR data archive using the following information.

   Host Name/Address: ftp.igsb.uiowa.edu
   Use "Anonymous Login"
   Click "OK"

   This sets up a connection with the ftp site. The ftp window should show your local directories on the left side (the local system) and the ftp site's directory on the right (the remote system). On the local system, change the directory to c:\mytemp. On the ftp site, move to the subdirectory pub/gisdata and select the files index.txt, how_to.txt, and iowa_gis.txt (use the CTRL key to select more than one file). Click on the left-pointing arrow to move the files to c:\mytemp.

   Now change to the /pub/gisdata/lastate subdirectory. Select the county coverage -- three files will be needed for each coverage, county.taz, county.met (or .doc), and
county.txt. Make sure that you have selected the "Binary" option within WSFTP and then start the transfer to c:\mytemp. 3. Move to c:\mytemp in Explorer (or "My Computer"). Double click on the county WinZIP file (the one with the "folder-in-a-vise" icon). This will bring up an "unregistered version" of WinZip. Click "I agree" since you are using WinZip only for evaluation and thus can use the free version. The next window will ask if you want to decompress the .taz file you downloaded. Click "Yes."

A WinZip window opens that displays three files contained within county.tar. Select all three files. The county.e00 file is the county coverage for the state of Iowa in Arc/Info's export format (as indicated by the .e00 extension). Choose Actions>Extract. Overwrite any existing files. In the next window, make sure to type c:\mytemp in the "Extract to:" space. Click on "Extract."

4. Start MapInfo and choose Tools > ArcLink > ARCINFO->MapInfo.

Change into the directory (folder) that has the .e00 file that you downloaded and un-compressed. Select the Export file county.e00 and then click the Save and Set Output button.

Click the "Projection File" button to set the proper projection system. In this case it is Category: Universal Transverse Mercator (NAD 27) and Category Member: UTM Zone 15 (NAD 27). Click OK.

From the Save and Set Output Parameters dialog click OK. Then Click Translate. When it is done, use "Cancel" to leave the translator.

Open the table countyp.tab (polygons), countyl.tab (lines), and/or countya.tab (annotation).

5. To access DNR files from the Web, first start Netscape. Go to the DNR web page at http://www.igsb.uiowa.edu/nrgis/gishome.htm

Select county and state-wide coverages to download. For example, click "Iowa State-Wide Coverages" then "Cultural and Political Geography". Click on the DOC option to view metadata. Choose the FTP option for Hospitals in Iowa. Follow the necessary steps to save the file to a folder. Use the steps above to uncompress the file
and import it to MapInfo. Open the new MapInfo table and display it on top of the county coverage.
# APPENDIX K

## METHODOLOGY TABLES

Table K.1 Tenets and concepts of ecological mediation, with potential methods for investigating them.

<table>
<thead>
<tr>
<th>Tenets</th>
<th>Key Concepts</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compound mediation</td>
<td>Ecology of genres</td>
<td>Naturalistic observation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interviews</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethnography</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protocol analysis</td>
</tr>
<tr>
<td>Ecological</td>
<td>Importation</td>
<td>Historical analysis</td>
</tr>
<tr>
<td>development</td>
<td>Genre perception</td>
<td>Genre analysis</td>
</tr>
<tr>
<td></td>
<td>Genre management</td>
<td>Document analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rhetorical analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conversation analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discourse analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protocol analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cooperative prototyping</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Participatory design</td>
</tr>
<tr>
<td>Hybridization</td>
<td>Hybrid genres</td>
<td>Genre analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rhetorical analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conversation analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discourse analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protocol analysis</td>
</tr>
</tbody>
</table>
Table K.2 Distributed usability: types of analysis, conceptual units, levels, and potential methods.

<table>
<thead>
<tr>
<th>Type of analysis</th>
<th>Conceptual Unit</th>
<th>Level and Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contradictions</td>
<td>Activity Network (Activity)</td>
<td>Macro-level (Cultural-Historical)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Historical analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Document analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cultural criticism</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Genre analysis</td>
</tr>
<tr>
<td>Discoordinations</td>
<td>Genre (Activity, Action)</td>
<td>Meso-level (Personal and Organizational)</td>
</tr>
<tr>
<td>(Genre Perception, Genre Management)</td>
<td></td>
<td>Ethnographic research</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naturalistic observations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Case studies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stimulated recall interviews</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Questionnaire and survey analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cooperative prototyping</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Participatory design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intervention protocols</td>
</tr>
<tr>
<td>Breakdowns</td>
<td>Interpretive Moment (Action, Operation)</td>
<td>Micro-level (Episodic and Cumulative)</td>
</tr>
<tr>
<td>(Reinterpretation)</td>
<td></td>
<td>Experimentation and quasi-experimentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quantitative and statistical analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discourse analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conversation analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protocol analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Video coding and audio coding</td>
</tr>
</tbody>
</table>

Table K.3 Design stages and methods.

<table>
<thead>
<tr>
<th>Design Stage</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speculating about the results of addressing contradictions</td>
<td>Mapping of genre ecologies</td>
</tr>
<tr>
<td></td>
<td>Mapping of activity networks</td>
</tr>
<tr>
<td>Conducting limited design changes</td>
<td>Cooperative prototyping</td>
</tr>
<tr>
<td></td>
<td>Participative design</td>
</tr>
<tr>
<td>Implementing fuller changes</td>
<td>Limited deployment in actual work</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


Iowa Department of Transportation. (1990). The DOT's shining STARS. *INSIDE, 2.*


