An Infrastructure to Support Usability Problem Data Analysis

Jonathan R. Howarth

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Rex Hartson
Mary Beth Rosson
Manuel A. Pérez-Quiñones

May 6, 2004
Blacksburg, Virginia

Keywords: Usability Engineering, Usability Problem Analysis, Usability Problem Data Management

An Infrastructure to Support Usability Problem Data Analysis

Jonathan R. Howarth

Abstract

Increasing the usability of software by integrating usability engineering into the development cycle has become common practice. Although usability engineering is effective, it can be expensive, and organizations want to receive the best possible returns on their investments. Oftentimes, however, organizations spend large sums of money collecting usability problem data through activities such as usability testing, but do not receive acceptable returns on those investments during redesign. The primary reason is that there is an almost complete lack of methods and tools for usability problem data analysis to transform raw usability data into effective inputs for developers. In this thesis, we develop an infrastructure for usability problem data analysis to address the need for better returns on usability engineering investments. The infrastructure consists of four main components: a framework, a process, tools, and semantic analysis technology. Embedded within the infrastructure is the User Action Framework, a conceptual framework of usability concepts, which is used to organize usability data. The process addresses extraction of usability problems from raw usability data, diagnosis of problems according to usability concepts, and reporting of problems in a form that is usable by developers. The tools leverage the framework and guide practitioners through the process, while the semantic analysis technology supplements the capabilities of the tools to automate parts of the process.

Dedication

I dedicate this work to my family. They have always encouraged me and supported me, and I just want to thank them. I love you Mom, Dad, and Liz.

Acknowledgements

I owe special thanks to Dr. Rex Hartson. He has been my mentor at Virginia Tech and has taught me much about the world of academia as well as life in general. He has inspired me and encouraged me. His hard work and dedication were instrumental in the development of this thesis. I hope that someday I will be able to provide a student with the same support and friendship that he has shown me.

I would also like to thank the Air Force Office of Scientific Research for their support with funding for phase 1 of Small Business Technology Transfer Topic AF03 – T001 through Knowledge-Analysis-Technologies, Boulder, CO.

In addition, I would like to thank Darrell Laham of Knowledge-Analysis-Technologies for his help with latent semantic analysis technology.

Table of Contents

List of Figures

List of Tables

1 Overview

1.1 Motivation

Increasing the usability of software by integrating usability engineering into the development cycle has become common practice. In fact, usability is one of the primary reasons why a piece of software fails or succeeds in the competitive software market. Downstream in the process from usability testing, however, developers often receive a low return on their usability engineering investments because of information losses between usability problem capture and redesign. This information loss is due to a single fact: there is an almost complete lack of methods and tools for usability problem data analysis to transform raw usability data into effective inputs to redesign for fixing usability problems found in usability testing. Thus, there is a definite need for effective usability data analysis techniques and support tools that significantly increase returns on investments in usability testing.

Our work was motivated by the realization that usability practitioners cannot expect to know what to "fix" in redesign without understanding the problem in terms of its type, how it interferes with the user's cognitive and physical actions in task performance, and its causes within the interaction design. Usability problems that look similar on the surface can have different underlying causes, and vice versa. Trying to fix the wrong problem can lead to misdirected solutions and wasted resources, lost opportunities to fix the right problem, and can even cause new usability problems. Additionally, we realized that an accessible repository of usability problems and their analyses would support re-use of usability problem data and analysis effort.

1.2 Scope and Setting of the Work

Figure 1 depicts the usability data management cycle, which occurs within the overall usability engineering development process. In this thesis, we concern ourselves with a subset of this cycle, “B” in Figure 1, which addresses the analysis of raw usability data to produce problem reports that facilitate design correction. In particular, analysis is composed of the activities of usability problem extraction, diagnosis, and reporting.

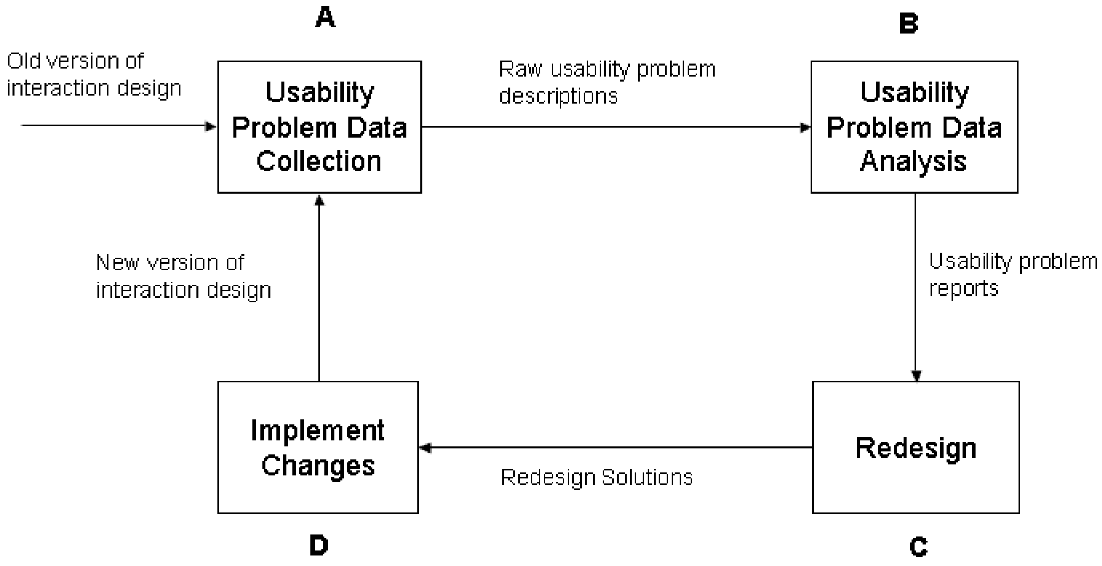


Figure 1: Usability data management cycle

Usability testing at "A" of Figure 1 produces raw usability data in the form of critical incident and usability problem descriptions. In principle, through usability problem data analysis, individual problem descriptions are later extracted and diagnosed at "B" and converted into usability problem reports for subsequent fixing with design/redesign solutions at "C." After changes are implemented at "D," the process can be iterated with successive testing, analysis, and redesign. In practice, however, the lack of methods and tools for analysis at "B" leads to poor communication of usability data, obstructing appropriate redesign solutions at "C."

A scenario illustrates the problem of low returns with the existing usability data management cycle. A usability practitioner is performing usability testing to evaluate the interface of a new software application. The practitioner observes a participant as she performs benchmark tasks and records critical incidents that occur when she has a problem. The practitioner has to record a large number of problems and, out of necessity, writes brief descriptions of the problems, so that he can keep up with the participant. These problem descriptions represent the raw usability data that comes out of the collection stage. The practitioner has a minimum amount of time to work with the data before the next project and performs basic analysis including extraction and grouping of the problems to create problem reports. He then hands these reports off to the developers who will use them to redesign the interface. When they receive the reports, however, the developers are working on optimizing the backend of the system and cannot make immediate use of them. The reports sit unused for two months before the developers are ready to redesign the interface. When they read the reports, the developers have several questions about the problems and are not sure what constitutes an appropriate fix. The brief descriptions that were good enough for the practitioner at the time of collection are too vague for the developers who were not present for the usability testing. They contact the practitioner who performed the evaluation, but he has since performed two usability evaluations and can no longer remember exactly what interaction design flaws were responsible for the problems. As a result, the developers make informed guesses and create new problems by fixing nonexistent problems.

We have addressed the problem of low returns on usability engineering investments by creating an infrastructure for effective usability problem data analysis in “B.” We have divided our work into two phases as illustrated in Figure 2.

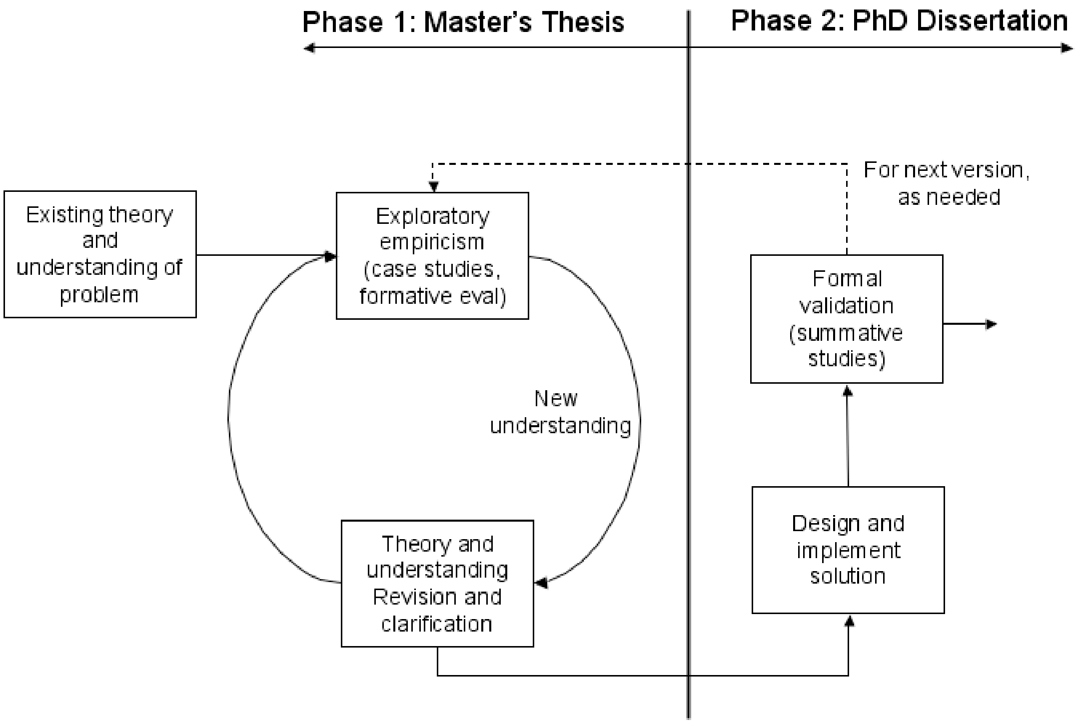


Figure 2: Epistemological cycle

Phase 1, which provided immediate returns, is the primary focus of this thesis. The major technology for this phase was the User Action Framework (UAF). We performed a series of exploratory and formative studies to better our understanding of usability problem data analysis and develop strategies for improving it. The result of our work with phase 1 is an infrastructure for usability problem data analysis. Phase 2 is more involved and requires considerable resources, but it has the potential to provide valuable long-term results. Phase 2 will be the primary focus of a PhD dissertation and involves using latent semantic analysis (LSA) technologies to extend our work on our infrastructure by providing measures of semantic similarity for automating analysis. Phase 2 will involve formal validation through summative studies.

The following is a high level overview of our work on phase 1 in terms of the epistemological cycle (Figure 2); we describe this work in the Technical Discussion (section 6):

Exploratory Empiricism

We performed the following exploratory studies to aid in tuning the UAF:

- A two-part study helped us to learn about the diagnosis performance of new and intermediate users of the UAF. The results of the study helped us improve the content of nodes in the UAF and the overall structure.
- We performed a walkthrough of the UAF Planning sub-tree with another UAF expert. The expert identified some important problems in the UAF concerning consistency and structure, but also confirmed the efficacy of the new editing process that included look-ahead fields.

We performed the following exploratory studies to better understand the usability data management cycle:

- We studied existing processes for usability problem data analysis to determine what was limiting returns on usability investments in the usability problem data collection stage. We discovered that the problem reports that were generated during usability problem data analysis were often insufficient.
- Once we knew that usability problem reports were insufficient, we needed to understand why before we could improve them. Studies helped us determine that because problem reports were ad hoc and incomplete, differences in time, place, and people between usability problem data collection and redesign made it difficult for developers to correctly interpret the reports.
- Our work had given us an understanding of the ad hoc, incomplete nature of problem reports, but we still needed to find what information was necessary and how to structure that information. We performed studies that revealed that it is critical to capture immediate intention.
- Finally, we performed studies to determine how to capture immediate intention, and discovered that it is necessary to elicit intention information from the user during usability problem data collection.

We performed the following study to better understand the immediate intention identification process:

- The results of the exploratory studies of intention prompted us to study the effectiveness of the Wizard, a tool for identifying immediate intention. The study helped us to understand key differences between stages in the Interaction Cycle and develop questions that focused on those differences.

Theory and Understanding

The exploratory empirical studies helped us to better understand and develop theories for the following:

- The user's immediate intention is the kind of action that the user is doing or attempting in terms of the stage within the Interaction Cycle that the user is operating in when a problem occurs. We have developed a new usability data management process based on our working hypothesis that initial diagnosis is necessary to capture immediate intention information.

- The process of diagnosis in the UAF involves associating a usability problem with a path of nodes that completely describes the problem and its causes. We have changed the way that we organize content in UAF nodes. In particular, we believe that cross references should appear first to put a user back on the right path before they view the contents of a node and that look-ahead fields increase consistency and guide users down a path.

Design and Development

We implemented and evaluated the following fully functional prototypes:

- The UAF Viewer is a tool that allows practitioners to navigate the UAF and view the content of individual nodes. The Viewer can be used for usability problem extraction and diagnosis and usability inspection.
- The UAF Editor is a tool that allows practitioners to edit the structure and content of the UAF. The Editor can be used for tuning the UAF or to extend it to new interaction styles.
- The UAF PRT is a tool that allows practitioners to catalogue usability problem data in the form of usability problem reports. The PRT allows practitioners to search and review reports in a manner that facilitates the identification of patterns and encourages reuse.

We developed a static prototype of the following:

- The UAF Wizard is a tool that facilitates the identification and reporting of immediate intention. The Wizard is used to help new users of the UAF learn the Interaction Cycle and help experienced users write better problem descriptions.

We performed the following initial work with LSA:

- We trained LSA on 537 articles from HCI journals and conferences.

2 Problem Statement

Many methods have been developed for usability evaluation, but few current processes can effectively make use of the problem data collected during evaluation. Organizations spend large sums of money on “A” (in Figure 1) buying usability equipment, building laboratories, and training developers, but they are not receiving acceptable returns on those investments downstream in “C” and “D”. The effects of the loss of necessary information during analysis are particularly evident during design/redesign. With current processes, the design/redesign and implementation process usually follows data gathering after a delay in time, is often done by different people, and frequently occurs at a different location. From extensive empirical investigation (Keenan, 1996), we know that most usability problem reports produced by usability professionals are incomplete ad hoc descriptions containing what was believed to be salient at the moment of capture.

Much of the critical knowledge is maintained in the mind of the problem collection facilitator and never gets recorded at “B.” When usability problem communication relies too heavily on human memory and word of mouth, developers at “C” can only try to interpret and reconstruct the missing usability information. The primary reason for the losses is the lack of an appropriate infrastructure for usability problem data analysis that ensures the capture and recording of necessary information; the existing infrastructures do not have the following:

1. A theory-based conceptual framework that organizes usability knowledge
2. An appropriate process that enables extraction, diagnosis, and reporting
3. Tools that guide professional usability engineering practitioners through the process
4. Technology for semantic processing to aid extraction, diagnosis, and reporting

3 Goals

The overall goal of this work was to develop an infrastructure for usability problem data analysis that enables effective usability problem data analysis – precise extraction, consistent diagnosis, and complete reporting. We achieved this goal by:

1. Adapting a theory-based conceptual framework
2. Creating an appropriate analysis process that leverages the framework
3. Conceptually designing tools and creating, building, and evaluating prototypes of those designs
4. Performing initial work in adopting and integrating technology for semantic analysis

4 Approach

We individually address our efforts to achieve the goals of adapting a conceptual framework, creating a process for usability problem data analysis, designing tools, and integrating technology for semantic analysis. These efforts helped us to create an infrastructure for usability problem data analysis, which will extend in phase 2 by integrating LSA technology (see section 7). The following subsections contain high-level overviews; we develop each in detail in the Technical Discussion (section 6).

4.1 Framework

A conceptual framework is essential for organizing problem data and establishing a common vocabulary for describing data. We chose the UAF, a structured knowledge base of usability concepts and issues, as our conceptual framework. It is an emerging technology, so we adapted it as we embedded it in our infrastructure.

Much of the work associated with integrating and adapting the framework involved review and revise tuning of the UAF. The UAF is a tree structure, the nodes of which represent usability concepts. The content of each node describes the usability problem represented by the node as well as its relation to other nodes. Initial studies involving usability problem diagnosis confirmed that the UAF needed further development to better define the nodes and distinguish them from one another. We refined the content of existing nodes, modified the structure of existing nodes, and supplemented the tree with new nodes based on information from formative evaluations of usability engineering student's diagnoses, focused formative evaluations utilizing verbal protocol, and walkthroughs performed by UAF experts.

In addition to tuning the UAF itself, our approach to the framework involved developing content to associate with nodes. The framework is an important part of our approach to usability problem diagnosis because it provides an organizing structure for content. This structure allows us to provide usability engineering professionals with critical information when they diagnose problems. We have designed our tools to associate case studies, literature, and information about guidelines with each node because such information helps practitioners understand the problem and relate it to other similar problems.

4.2 Usability Problem Data Analysis Process

We began by doing exploratory studies of current, existing processes for usability problem data analysis. Work by Keenan (Keenan, 1996) revealed that very few processes exist for problem analysis, and as a result, usability problem reports are ad hoc and insufficient for purposes of redesign and implementation. Simply knowing that the problem reports were insufficient, however, was not enough; we needed to understand what the processes were lacking or doing incorrectly before we could develop a more appropriate process. Our approach to determining this information was empirically driven. We conducted a series of case studies with analyst subjects serving as usability engineers performing usability problem data analysis. These studies revealed that differences in time, place, and people between collection and redesign makes it difficult for developers to understand the problems as they are documented in usability problem reports.

To generate ideas for a more appropriate process that included our understanding of the deficiencies with the current process, we took a broader perspective on diagnosis and compared diagnosis in usability engineering with diagnosis in other fields. The typical process in usability engineering, as shown in Figure 1, has separate sequential activities for data collection and analysis. If medical doctors followed a similar process, they would diagnose a patient by reading a nurse’s notes from a physical evaluation without ever seeing the patient. Instead, however, medical doctors use an iterative hypothetico-deductive process that ties data collection and analysis together closely; data collection drives analysis, but partial analysis also guides data collection (Capra, Hartson et al., 2002). Medical doctors use data to formulate potential diagnosis and determine what further information is necessary to rule in or out potential diagnosis. They then ask questions or perform tests to obtain the necessary information. The existing usability data management cycle limits similar interactions between practitioners and participants because the participants leave after usability data collection and before analysis.

Our experience with hundreds of real-world usability problem reports has shown that a strict separation of analysis from data collection makes effective problem diagnosis impossible. We believe that the best way to prevent information loss is to move part of diagnosis to the collection stage. This early diagnosis captures the participant’s intention when a critical incident occurs while the participant is still available for clarification purposes. Although iteration between collection and early diagnosis can incur a higher cost up front in terms of time and money, it limits costs later in the usability engineering cycle. Early partial diagnosis enables effective analysis by making diagnosis possible.

There is little point to limiting costs early in the data management cycle if it negates the payoff of the whole cycle.

4.3 Tools

We have developed tools based on the UAF that guide practitioners through the usability problem data analysis process. Figure 3 illustrates how the tools relate to the process. Usability problem analyst (UP analyst) denotes the usability practitioner responsible for usability problem data analysis. The UAF Editor allows the UAF developers to revise the content and structure of the UAF, while the UAF Viewer allows usability problem analysts to access the UAF. The Wizard helps analysts identify intention, and the UAF Problem Reporting Tool (PRT) allows analysts to catalog usability problem reports. Latent semantic analysis (LSA) technologies perform semantic analysis of usability problem descriptions supplied by usability problem analysts to help prepare them for cataloging in the UAF PRT.

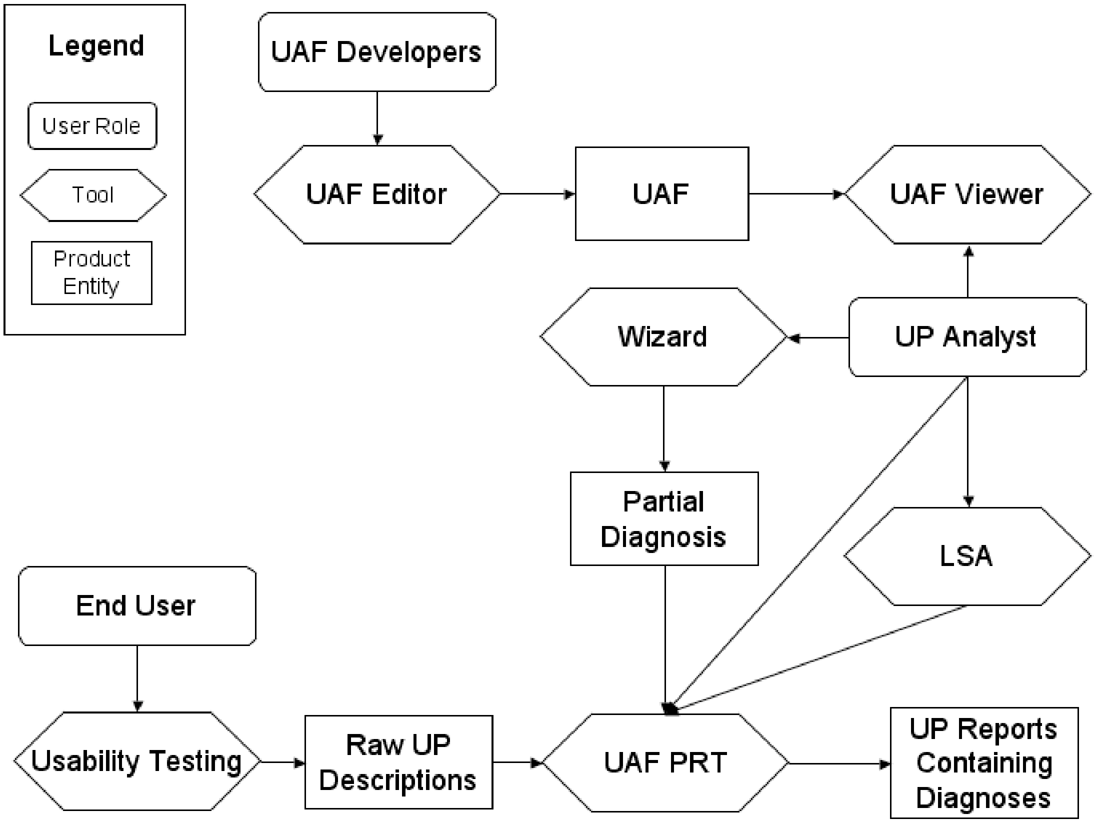


Figure 3: How the tools relate to the process

4.3.1 UAF Viewer, UAF Editor, and UAF PRT

We have developed three major tools that leverage the UAF as a theoretical basis and guide practitioners through the proposed problem analysis process: the UAF Viewer, the UAF Editor, and the Usability Problem Reporting Tool (UAF PRT).

- UAF Viewer – The UAF Viewer helps practitioners isolate individual usability problems from critical incident data and categorize them in a complete and precise manner that identifies the causes and provides solutions. Practitioners can navigate through the UAF tree and view individual nodes. Each node has content such as usability concepts and examples.
- UAF Editor – The UAF Editor allows practitioners to edit the structure and content of the UAF. We have primarily used the UAF Editor for tuning the existing UAF by review and revision, but practitioners can use it to extend the UAF to new interaction styles and to adapt it for their own environment.
- UAF PRT – This tool facilitates usability data management for analysis and reporting purposes. The tool helps practitioners catalog problem data in a manner that facilitates the identification of patterns or process dependencies both within a development effort and among multiple development efforts.

4.3.2 Wizard

The top-levels of the UAF represent the Interaction Cycle (see section 5.2.1). Our initial case studies involving usability problem data analysis confirmed the importance of choosing the correct (best match to the usability problem) top-level category, the stage in the Interaction Cycle, as the first step in diagnosis. In fact, analyst subjects who did not initially choose the correct top-level category rarely retraced their steps, thereby producing incorrect diagnoses. We felt that getting a correct start by selecting the right top-level category was so important that it was worth the effort to find effective ways to achieve it. We had been trying to improve analyst performance by revising and enhancing the top-level UAF category descriptions, but gains were limited. Evaluations of prototypes of our tools involving verbal protocol, however, provided us with the critical information that we needed to improve top-level diagnosis.

The evaluations helped us understand top-level diagnosis by allowing us to follow analyst subjects' thought processes while they tried to map usability problem descriptions to stages in the Interaction Cycle. The analyst subjects generally understood what was represented by the stages of the Interaction Cycle, but they had no process for comparing them. We noticed that when we coached analyst subjects at making this top-level diagnostic decision, it helped to break the multi-way decision down into a sequence of dependent two-way decisions, allowing the subjects to focus on a single issue or question at a time. Encouraged by initial success with our approach, we codified it into a sequence of two-answer questions, for which the answer to each would prune the number of choices remaining. The questions directly compared one stage with the other stages based on the distinguishing attributes of that stage. We refer to the collective set of questions as the Wizard. Through a process of elimination, the Wizard helps analyst subjects choose the right top-level category beginning with the least likely stages of the Interaction Cycle and continuing to the most likely stages. Initial testing indicated relatively fast top-level diagnosis performance could result from the Wizard approach combined with more user intention information.

4.4 Technology for Semantic Analysis

Our proposed usability problem data analysis process moves parts of usability problem diagnosis forward to the usability data collection stage. Early diagnosis requires that practitioners who are collecting data be able quickly and consistently to perform high-level diagnosis. Part of our approach to enabling early diagnosis involves the development of tools that help practitioners determine if they have collected enough data to identify intention. In addition, we also have the goal of automating parts of diagnosis. Our approach involves investigating the possibility of having our tools automatically select a diagnosis path in the UAF based on a problem description. For this task, as with identifying intention, the tools will have to work at the semantic level.

We decided to use Latent Semantic Analysis (LSA) technology to provide our tools with the necessary semantic processing capabilities. LSA is a technology that simulates human understanding of words and text (Landauer, Foltz et al., 1998). LSA builds a semantic space from a set of training texts and then uses that space to find semantic distances between other texts. The process for constructing the space takes into account a variety of factors such as co-occurrence of words and frequency of appearance to give LSA capabilities beyond those of term-matching technologies for determining semantic meaning. In this phase, we worked with Knowledge Analysis Technologies, Boulder, CO, a commercial company specializing in LSA technology, to train LSA with HCI literature that addresses user interaction with computer systems. The training is critical for our work in phase 2 because the LSA space must be appropriately constructed before we can apply it. We created an infrastructure in phase 1 that we will extend with LSA technology in phase 2 (see section 7).

5 Background and Related Work

This section provides detailed information on work related to the usability data management cycle, the UAF, and LSA. The related work provides the background necessary for understanding the difficulty of the problem of creating an infrastructure for usability problem data analysis. Existing work specifically addressing usability problem data analysis is limited and specifies methods and techniques that are too general or theoretical to be useful to practitioners. With our work, we want to provide usability engineers with a practical infrastructure for usability problem data analysis. More specifically, we want to enable precise extraction, consistent diagnosis, and complete reporting.

5.1 Usability Data Management Cycle

The usability data management cycle occurs within the usability engineering development process. We begin this section with a brief high-level discussion of usability engineering to give some context to the usability data management cycle, but the majority of the section is devoted to descriptions of the primary parts of the cycle.

5.1.1 Usability Engineering

Usability engineering is a process for improving the interaction design of interactive software systems to increase user effectiveness, efficiency, and satisfaction. Usability engineering has its roots in human performance evaluation and design technique development during World War II when machines became so complex that they were difficult to operate safely (Butler, 1996). Since then, usability engineering has become an essential part of modern software development efforts.

Although usability engineering is necessary, developing and maintaining a successful usability engineering process requires much effort. First, the manager or team leader must allocate sufficient budget and time. After the resources are in place, the project team must be convinced to buy in to the effort and participate in the process (Aucella, 1997). Developers may claim that the usability engineering process and the principles that it incorporates are obvious or common sense. Early research, however, found that only a small percentage of developers are even aware of the most common principles (Gould and Lewis, 1985). Practitioners also have to expend effort to modify theories and methods developed in academic circles for practical use. Oftentimes application of research technologies encounters a scale that simply cannot be replicated in the laboratory (Brooks, Ruven, 1990).

5.1.2 Overview of Usability Data Management Cycle

The usability data management cycle, which is depicted in Figure 4, is an important part of the larger usability engineering development process. Our research work for this thesis focuses on usability problem data analysis – extraction, diagnosis, and reporting.

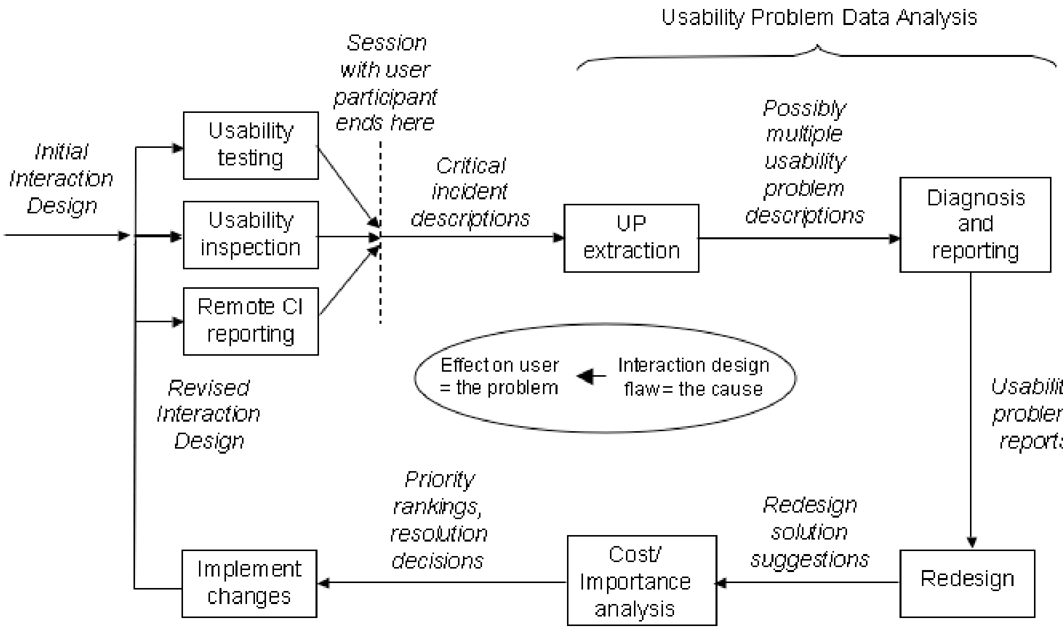


Figure 4: Existing usability data management cycle

The usability data management cycle is modified as needed to fit different work contexts and situations, but the overall cycle of analyze, design, build, and evaluate is fairly consistent across documented usability engineering efforts (Butler, 1996; Gould and Lewis, 1985). The usability data management cycle in Figure 4 consists of the following stages: usability data collection, usability problem extraction, usability problem diagnosis, usability problem reporting, cost/importance analysis, and redesign. The stages are repeated in an iterative fashion as allowed by budget and time constraints.

The usability engineering process is designed to identify and correct interaction design flaws that lead to usability problems. In particular, a design flaw may lead to one or more problems, and one or more problems may be fixed by correcting a design flaw.

5.1.3 Usability Data Collection

Usability data collection involves gathering critical incident data. Critical incidents may be referred to by other names such as errors, but they are used to capture events that occur while a participant is working and that have a significant effect on the task performance or user satisfaction (Norman, 1983). In particular, critical incidents indicate an interaction design flaw and form the basis for problem extraction and diagnosis.

Critical incidents can be gathered by one of four primary methods. The first method, usability testing, involves monitoring participants while they use software. This type of monitoring allows the facilitator to measure user task performance, observe behavior, and assess user satisfaction (Allen and Buie, 2002). Usability testing is expensive, but it is usually effective at identifying most of the usability problems.

The second method, usability inspection, is significantly less expensive than usability testing, but does not yield results that include input from actual users. Usability inspection involves evaluating an interface by examining and critiquing it. An example usability inspection method is Nielsen's discount usability engineering with heuristics (Nielsen and Molich, 1990). Heuristic evaluations generally find a large number of problems and have a reasonable learning curve, but they also have the weakness that they may generate false positives. Another type of inspection method is cognitive walkthroughs, which focus on the goals and knowledge of the user to detect possible interaction design flaws (Polson, Lewis et al., 1992; Wharton, Bradford et al., 1992). The cognitive walkthrough method has been traditionally the most difficult inspection method to learn and does not provide much information on users' physical actions and assessment of their actions. A third type of inspection method is guidelines that provide general rules to which the interface being evaluated should conform (Mayhew, 1992; Shneiderman, 1998).

Remote critical incident reporting, the third method, is performed by users of software in their own work environments when they experience critical incidents while doing real work-related tasks (Castillo, Hartson et al., 1998). Such an approach is certainly inexpensive in terms of collecting critical incident data, but it may result in problem descriptions with vague or misleading information on the user's intention and overall goal.

A fourth category of collection methods contains those that are automated. For example, Hilbert and Redmiles discuss automatically capturing UI information such as the frequency with which users perform a certain action (Hilbert and Redmiles, 2000). Automated methods are not intended to be used alone, but instead to complement other methods.

5.1.4 Usability Problem Extraction

Critical incident descriptions are raw data, which must be processed to be usable. The first step in usability problem data analysis processing is extracting usability problems. In work in the field of software engineering, Ostrand defines problems generally as situations that arise when an error has been committed (Ostrand and Weyuker, 1984). Dealing with errors involves detecting them and then isolating them to determine the precise nature of the faults that result from the error. The same concept is applicable to the field of usability engineering; the error is a design flaw, and the problem is the effect on the user. It is important to distinguish individual problems because one critical incident may contain multiple problems, each of which map to different interaction design flaws.

Previous work with problem extraction is limited. An example is Cockton and Lavery's framework for usability problem extraction called Structured Usability Problem EXtraction (SUPEX) (Cockton and Lavery, 1999). A main goal of the framework is to help developers reliably and consistently transform usability data into distinct usability problems. The framework represents a good start for post-collection data management, but does not address how to use the extracted problems to improve interaction design.

5.1.5 Usability Problem Diagnosis

Usability problem diagnosis is the next part of usability problem data analysis and is performed on an individual problem to determine problem type, subtype, and causes. The space for usability problems can be described as multidimensional, and diagnosis involves navigating this space to select the dimensions, which best encompass the problem.

The primary motivation for diagnosis is that it yields a clear, complete, and unambiguous statement of the design flaw to be fixed. A secondary motivation for diagnosis is its role in terms of normalizing problem descriptions for comparison and evaluation. There are multiple usability evaluation methods for usability problem collection that may result in problem descriptions of varying detail and quality. No one method has been proven most effective, so different methods are used in different contexts. Muller, Dayton, and Root discuss how documented comparisons of different usability evaluations disagree as to which one is the most effective (Muller, Dayton et al., 1993). Gray and Salzman take a different approach and discuss how different studies of evaluation methods may have been flawed or inconclusive with respect to statistical, internal, construct, and external validity (Gray and Salzman, 1998). In another study, Molich et al. submit an application for review by four usability firms (Molich, Bevan et al., 1998). The number of problems identified in the firms' reports varied from 4 to 98 and there was no significant overlap in problems identified. Because the input coming from the collection stage has the potential to be quite varied, diagnosis is necessary to provide some basis for problem organization and comparison.

The idea of usability problem diagnosis is not novel; several researchers have worked to develop diagnosis methods for usability data. An early and highly recognized method is Nielsen and Molich's work with heuristics (Nielsen and Molich, 1990). Example heuristics include use simple and natural dialogs, speak the user's language, and provide feedback. Heuristics can be used for diagnosis in the sense that they provide a way for identifying and grouping problems; the problem, however, is that they are at such a high-level that they do little to identify specific solutions.

Design guidelines are in many ways similar to heuristics. Guidelines provide general rules that are intended to help moderate a design effort. Mayhew and Schneiderman have published textbooks in this area that integrate years of design experience and research into simple principles (Mayhew, 1992; Shneiderman, 1998). The main issue with guidelines is their necessarily overly general nature. In addition, it is often difficult to determine exactly which guidelines apply to a given situation and which guidelines take precedence over others. Thus diagnosis with guidelines is oftentimes dependent on the experience of the practitioner or simply preference.

Other researchers have explored classification, an area related to diagnosis, in general (Mack and Montaniz, 1994) or by source and location in dialogue (Brooks, P., 1994; Nielsen, 1993). Simple schemes are used for classifying problems by severity or importance (Desurvire, 1994; Nielsen, 1993; Rubin, 1994), or based on type of user error (Vora, 1995). In general, however, these approaches to classification have been ad hoc, incomplete, unstructured, and usually unhelpful for finding solutions to usability problems in an interaction design.

As an additional note on diagnosis, previous research has compared diagnosis in usability engineering with diagnosis in other disciplines. For example, Capra makes an interesting comparison between usability problem diagnosis and medical diagnosis (Capra, Hartson et al., 2002). In particular, both forms of diagnosis rely on expertise (i.e., skill and experience) rather than just factual knowledge and require the ability to focus on relevant information and discard irrelevant information. Work by Griffen, Schwartz, and Sofronoff on implicit processes in medical diagnosis also has some interesting parallels with diagnosis in usability engineering (Griffin, Schwartz et al., 1998). In particular, diseases may have many signs or symptoms, only some of which are present for a given instance of the disease. In much the same way, usability problems often manifest themselves with users in different manners, and very different design flaws may have similar manifestations. Usability engineering, however, has the advantage that there are often opportunities to observe the cause (in the design) and effect (on the user) relationship in fairly close proximity if practitioners are sensitive to it.

5.1.6 Reporting of Usability Problems

Also included in usability problem analysis is the reporting of problems. The results of usability problem diagnosis must be reported in a way that is useful to developers who are responsible for redesign and implementation. Lavery and Cockton describe problem reports as representations of predicted usability problems in the case of inspection methods or actual problems in the case of usability testing (Lavery and Cockton, 1997). The key to problem reports is determining what to represent and how to represent it. Lavery, Cockton, and Atkinson discuss structuring problem reports in an attempt to standardize problem reports for comparing usability problem detection rates among collection methods (Lavery, Cockton et al., 1997). They maintain that some early attempts at developing a structure for usability problem reports did not include a clear definition of what constitutes a usability problem. For example, work by Jeffries reports usability problems in terms of proposed solutions (Jeffries, 1994). Such an approach is problematic because proposed solutions do not necessarily imply a cause and thus do not always help to identify design flaws. Lavery, Cockton, and Atkinson give a definition of what constitutes a usability problem and create a problem report structure that captures causes and outcomes associated with a problem. The causes are responsible for the breakdowns that the user experiences as outcomes. The rationale behind capturing both causes and outcomes is that design changes that fix causes will avoid breakdowns and no longer produce negative outcomes.

As part of this thesis, we will create a database of usability problem reports. Previous work with databases includes that of Pernice and Butler who describe using databases to compare charting features in Lotus 1-2-3 and Lotus Freelance Graphics (Pernice and Butler, 1995). Lotus uses databases to hold lists of tasks for testing, contact information and computer abilities of possible test participants, report templates, and problem reports. Work by Scapin also describes the use of databases for usability data of a different nature (Scapin, 1990). In particular, the author collected recommendations and guidelines from human factors publications and organized them according to criteria such as consistency, level in terms of abstract layered models, and type of interface object. Wilson and Coyne provide an additional discussion of cataloging (Wilson and Coyne, 2001) by discussing the merits of reporting usability problems in a corporate database that also contains functional bugs. If the same database is used, the usability problems receive more attention, but run the risk of being dismissed as less important than functional bugs.

5.1.7 Redesign and Cost/Importance Analysis

The reports must be made accessible, so that they can be used for redesign and for improvement of the usability engineering process. Redesign is concerned with fixing problems found in the current development effort and consists of cost/importance analysis, design, and implementation, which are described in the following paragraphs. Improving the usability engineering process, on the other hand, involves monitoring problems among multiple development efforts and discovering trends and deficiencies by querying and filtering of a collection of problem reports.

The next step consists of iteration between cost/importance analysis and design. Cost/Importance analysis involves ordering usability problems based on their perceived importance and cost to fix in terms of hours needed to implement a design change (Hix and Hartson, 1993). Design involves determining appropriate fixes for the interaction design flaws that caused the problems. Some amount of iteration is necessary because it is important to identify both candidates for fixes and the cost of possible fixes. The end result of this iterative step is a priority ranking of problems with their associated fixes and a cutoff line that represents budget and time limitations.

The need for iteration in the usability data management cycle is particularly apparent in the design phase. Style may be subjective, but good design is not. “Research shows no relationship between users’ subjective ratings of a product and their objectively measured performance in using it” (Allen and Buie, 2002). Iteration is necessary to incorporate and test design changes; methodical iteration can improve design and lead to an overall increase in usability.

After appropriate design decisions have been made, they are implemented to produce a new version of the software. Permitting budgetary and time limitations, the resulting interaction design can be subjected to the usability engineering data management cycle for as many iterations as necessary to obtain a sufficient level of quality.

5.2 UAF

The User Action Framework (UAF) is a theory-based conceptual framework of usability concepts. The UAF was conceived of and developed by Hartson and represents an extensive effort, which integrates existing work on usability problem identification and diagnosis (Hartson, Andre et al., 1999). The UAF is the theoretical base for our work on developing an infrastructure for usability problem data analysis.

This section provides an overview of the UAF and related work. We begin with an introduction to the Interaction Cycle and an explanation of how it is used in the UAF, followed by a discussion of Norman's seven-stage theory of action model, the basis for the Interaction Cycle, and other research that has incorporated it. We next describe diagnosis using the UAF and previous research efforts involving diagnosis. Thereafter, we discuss the results of evaluation of the UAF.

5.2.1 Interaction Cycle

The UAF is essentially a tree structure with each node representing a usability concept. The sibling nodes directly under the root node are organized according to the Interaction Cycle, an extension of Norman's seven-stage theory of action model (Norman and Draper, 1986). These nodes represent planning, translation, physical actions, outcome and system functionality, and assessment and demonstrate the role of interaction design in supporting the cognitive, physical, and sensory actions of computer users. The following figure illustrates the Interaction Cycle:

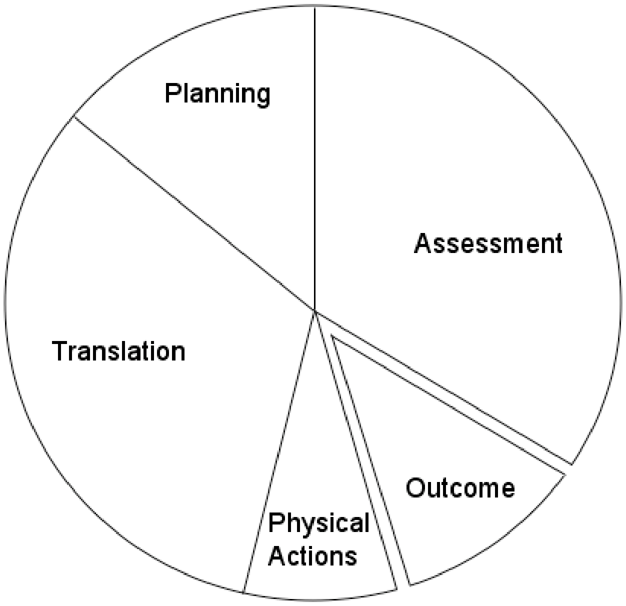


Figure 5: Interaction Cycle

Planning involves the cognitive processes of users as they decide what to do; it is important that users have an understanding of the system model as well as their progress towards the completion of a task. During planning, users work to understand their task and potential approaches to successfully completing it. Users select an approach and associate a goal with that approach, which they then use to formulate one or more intentions that will determine their interaction with the interface.

During translation, users determine how to specify the actions that correspond to their intentions. More specifically, users interact with design features that help or enable thinking or knowing about an aspect of the interface. Norman refers to such features as perceived affordances and Hartson refers to them as cognitive affordances (Hartson, 2003).

The physical action stage actually involves the user interacting with an interface object to perform an action. Real affordances, in Norman's terms, or physical affordances, in Hartson's terms, provide users with an idea of how to physically manipulate objects in the interface. The physical action stage is concerned with both the efficiency of physical manipulations and the user's ability to perform them.

The outcome and system functionality stage deals strictly with issues internal to the system and has nothing to do with issues about interaction design. This stage is included in the Interaction Cycle to capture problems that indicate malfunctioning or missing functional affordances (non-user-interface functionality).

Assessment, the final stage in the cycle, is where users determine, based on feedback from the system, how effective their actions were in accomplishing a task. Sensory affordances, Hartson's term for design features that help a user see, hear, or feel the response from the system, play a large role in the user's ability to determine if the system has responded. The user's cognitive facilities determine whether or not the response corresponds to the desired outcome.

5.2.2 Norman's Model

The basis for the Interaction Cycle is a model of action proposed by Norman for a human's interaction with any type of machine (Norman and Draper, 1986). The seven stages of the model are organized according to one of three basic categories: execution, physical activity, and evaluation. The execution category maps to planning and translation in the Interaction Cycle, the physical activity category maps to physical actions, and the evaluation category maps to assessment.

Because it is general enough to represent human interaction with a variety of machines in a variety of contexts, Norman's model lends itself to adaptation and extension. As a result, other researchers have incorporated or leveraged Norman's model for their own work.

One example is Lim, Benbasat, and Todd's work with Norman's model as a basis for determining why and in what context direct manipulation is superior to other types of interfaces (Lim, Benbasat et al., 1996). During the research process, the authors use action identification theory and the theory of automaticity to compare menu-based interaction and direct manipulation in terms of time spent performing motor activities and cognitive activities. The authors conclude that familiar tasks map to Norman's idea of goal composition while unfamiliar tasks are seen at the action specification level.

Rizzo, Marchigiani, and Andreadis describe a modification of the cognitive walkthrough based on Norman's model (Rizzo, Marchigiani et al., 1997). In particular, the authors document a process that is tailored for the AVANTI project, an effort that required the cooperation of design teams in different cities and the ability to make high-level design decisions. The authors propose a modified version of Norman's model that takes into account goal shifts that result from realizations or the inability to perform an action. The walkthrough was effective because it allowed the team members to communicate problems clearly at a high level.

Kaur, Maiden, and Sutcliffe describe another application of Norman's theory (Kaur, Maiden et al., 1999). The authors develop a model of interaction for virtual environments. Specifically, they modify Norman's model to account for exploratory, opportunistic, and reactive behaviors because many objects in virtual environments are either not present or partially automated. The task action mode, the mode of interaction based on Norman's model, is combined with two other modes to better describe interaction.

5.2.3 Diagnosis with the UAF

The UAF is designed to aid practitioners in diagnosing problems according to usability concepts. Diagnosing problems allows practitioners to better document them, communicate them, and spot trends and relationships among them. Diagnosis involves associating a usability problem with the correct usability concept. Work by Springett, however, suggests that consistently making this association may be difficult because the link between the surface characteristic of an error and the root cause are often difficult to determine (Springett, 1998). As such, usability practitioners need a structure and process that will help them to consistently determine the correct link and translate it into a diagnosis. The UAF provides this structure and consistency.

The process of diagnosis in the UAF involves associating a usability problem with a path of nodes that completely describes the problem and its causes. In more abstract terms, the tree structure of the UAF represents the multidimensional space of problem data. A tree structure allows a user to navigate the dimensions of the space, ultimately arriving at a specific location within the space. Each level of the tree structure maps to a dimension, and each node at a given level maps to an attribute or value within that dimension. Selecting one of the nodes at a given level is equivalent to removing unnecessary attributes, thereby filtering or pruning off sub-trees. Traversing the tree is equivalent to selecting a path within a decision tree, where each node selection results in sub-tree pruning, leading to the identification of a specific dimension and attribute that best represent the problem and its causes.

Diagnosis with the UAF provides precision, reliability, and completeness. Once a problem has been associated with a node, the path to that node contains all the information needed to specifically identify the problem. Precision is ensured by the standardized usability vocabulary used. Reliability is ensured because other problems that have the same attributes will be placed in the same node, and completeness is also ensured because only one path leads to a given node.

Diagnosis with the UAF is aided by the UAF's tree structure. In addition to providing completeness and precision, a tree is a natural way to organize usability problem data. Existing techniques for understanding data include affinity diagrams, priority ranking, and Pareto diagrams; such techniques require grouping data and have the ultimate goal of organization (Nayak, Mrazek et al., 1995). Trees provide the same functionality, but do so with a structure that can be reused in future development efforts.

A final note on diagnosis with the UAF is that the tree structure facilitates redesign by organizing problems in a way that facilitates the identification of design changes. Nayak, Mrazek, and Smith argue that techniques that are easy to translate to solutions increase team acceptance (Nayak, Mrazek et al., 1995). By completely specifying a problem, the tree allows developers to understand the specific causes of the problem and the changes necessary to correct it. Through time, developers can associate generic solutions with nodes and increase the speed of the correction process by reusing problem analysis effort.

5.2.4 Evaluation

The UAF has been evaluated in two major studies. The first study conducted by Andre et al. tested the reliability of the UAF (Andre, Hartson et al., 2001). The second study by Andre, Hartson, and Williges compared the usability problem inspector (UPI), an inspection tool interface to the UAF, with heuristics and cognitive walkthroughs (Andre, Hartson et al., 2002).

The goal of the reliability study was to document the level of agreement among professional usability engineering practitioners when the UAF was used as a diagnosis structure. The results of diagnosis with the UAF were compared to results from an evaluation based on Nielsen's heuristics and to results from a study with the Usability Problem Taxonomy (Keenan, 1996; Keenan, Hartson et al., 1999), an earlier diagnosis structure that was helpful in the creation of the UAF.

In the study, 10 usability professionals with brief training on the UAF structure classified 20 usability problems using the UAF. The usability problem descriptions given to the usability professionals were edited, so that they only contained one major usability problem. The authors used the kappa statistic to measure reliability because it is commonly used to measure agreement involving lists or taxonomies. A kappa value is scaled for the range –1 to 1. A value of 0 indicates chance agreement and values greater than 0 indicate stronger agreement. The authors showed measures of reliability at each level in the UAF, within each major Interaction Cycle part (Planning, Physical Actions, Assessment), and overall. The UAF showed very strong agreement for all measures. For comparison with the Usability Problem Taxonomy and the heuristic evaluation, only the UAF's overall score was used. The UAF had a kappa value of .583, which was significantly better than the heuristic evaluation's score of .325. The UAF also improved upon the Usability Problem Taxonomy's score of .403.

The second study focused on comparing the following usability inspection methods: the UPI, heuristics, and the cognitive walkthrough. The UAF serves as the theoretical base for the UPI. Because there are no standard criteria for comparing usability evaluation methods, the authors compared the methods in terms of thoroughness, validity, and effectiveness, which are measurements derived from the work of Hartson, Andre, and Williges. (Hartson, Andre et al., 2003). Thoroughness is the ratio of real usability problems identified by the usability evaluation method over the base set of problems, validity is the ratio of the base set of problems over the identified problems, and effectiveness is the product of thoroughness and validity. The authors used an address book application and developed a base set of problems by performing usability testing with 20 participants. The authors assigned 30 usability practitioners one of the three usability evaluation methods and recorded and analyzed the results.

The results of the study indicate that the UPI and the cognitive walkthrough have higher levels of thoroughness, validity, and effectiveness than heuristics. Much of the effort in the heuristic evaluations was directed towards the identification of problems that were not in the base set yielding low validity measures. The UPI and the cognitive walkthrough performed equally. The authors conclude that the UPI will perform better in the long term because it provides more detailed problem information.

Although the UAF is reliable when used by professional usability engineering practitioners, we have run informal studies that show that novice usability engineers who are not familiar with the UAF cannot achieve acceptable levels of reliability. We theorize that new users do not have the experience to fill in gaps in intention in the usability problem reports. Our work with this thesis will improve new users' performance and subsequently strengthen advanced users' performance.

5.3 LSA

Landauer, Foltz, and Laham describe LSA as “a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text” (Landauer, Foltz et al., 1998). There are two basic uses of LSA: simulating human understanding of words and text and as a model of how knowledge is acquired and used. For the purposes of this thesis, we are concerned only with the former use.

We begin this section with a brief summary of the functioning of LSA. Thereafter, we give an overview of research detailing the uses of LSA in terms of simulating human understanding of words and text.

5.3.1 Functioning of LSA

Landauer, Foltz, and Laham give a general introduction to the functioning of LSA (Landauer, Foltz et al., 1998). The LSA process begins with the creation of a matrix from the target set of text. The rows of the matrix represent unique words found in the text while the columns represent the smallest text unit such as a sentence or paragraph. After the matrix has been constructed, a transformation is performed to weight the value in each cell. The weighting function takes into account a word’s importance in the particular passage and the degree to which the word relates to information in the domain of discourse in general. The transformation greatly increases the accuracy of the process. Singular value decomposition (SVD) is next applied to the matrix; thereafter, the number of dimensions of the matrix is reduced to some optimal size. A text is represented by a vector in this reduced space, and the cosine between two vectors serves as a measure of semantic distance in the space.

Two aspects of LSA merit further description: the transformation and SVD. The transformation consists of two steps. The first step is to take the log value of each cell. The second step, computing the entropy value, consists of calculating the row’s entropy value (p log p) and then dividing the cell’s value by the row’s entropy value. Dividing by the row’s entropy value weights the frequency of occurrence of a word in a text unit by its importance in a passage and inversely by the degree to which knowing that a word occurs provides some indication of the passage. Work by Harmon discusses the transformation process and provides details as to how it achieves the desired weighting relationship among words in a text (Harmon, 1986).

SVD decomposes a matrix A into three matrices U, D, and V: $A = UDV^T$. U and V have orthogonal columns and V^T represents V transposed. D is a diagonal matrix. The top k columns are selected from U and V^T during the reduction process based on the values in the diagonal matrix D. The values in D are a measure of importance; k is selected such that for integer values of $n > 0$, the value of $D[k+n, k+n]$ is relatively insignificant. The three matrices are multiplied after the top k columns have been selected to yield the reduced space, $A' = UD' V^T$, where $D' = D[k, k]$ (Dumais, Furnas et al., 1988).

The described process gives LSA certain properties or characteristics. One property from which LSA derives its name is that it is able to establish deeper relations than simple contiguity frequencies, co-occurrence counts, or correlations. LSA does not compare words within a text unit. A word is associated with an entire text unit; LSA does not count tuples of word occurrences. As such, LSA is able to relate words that never appear together in the same text unit. Another property is that LSA does not rely on an understanding of grammar or syntax; it does not process word order.

5.3.2 Simulating Human Understanding of Words and Text

Researchers have used LSA for a variety of purposes, some of which are particularly important for our work. In this section, we discuss these relevant purposes and give examples from the literature.

5.3.2.1 Synonymy and Polysemy

Synonymy is the diversity of words that humans use to describe the same concept or object, and polysemy is the ability of a word to have more than one meaning. Synonymy and polysemy represent research challenges because they are important in determining meaning. For the purposes of this thesis, research on synonymy and polysemy involving LSA can help us to compare nodes and paths, determine the best wording of nodes, and automate the diagnosis process.

A major research area using LSA to help address synonymy and polysemy is information retrieval. Ideally, users of retrieval systems want to be able to access information according to meaning. However, term matching, the most common retrieval method, is hampered by synonymy and polysemy. To compensate, information retrieval systems have taken approaches such as restricting allowable vocabulary, augmenting query terms with terms from a thesaurus, and constructing explicit models that reflect the semantics of the domain (Dumais, Furnas et al., 1988).

Studies by Dumais et al. and Deerwester et al. describe comparisons of document retrieval with LSA and with term matching (Deerwester, Dumais et al., 1990; Dumais, Furnas et al., 1988). Dumais et al. found that document retrieval supported by LSA outperforms keyword retrievals when the documents are not too similar. Deerwester et al. compared document retrieval with LSA to term matching and other automated methods. LSA did not perform significantly better than the automated methods, most likely because the authors did not perform any preprocessing such as the log entropy method on the original matrix. The authors did find that LSA handled synonymy well, but had difficulty with polysemy.

Another study of synonymy and polysemy that is not directly related to information retrieval demonstrates LSA's ability to simulate human understanding of words. Landauer, Foltz, and Laham describe a study in which LSA was trained on an encyclopedia and then given a TOEFL language exam (Landauer, Foltz et al., 1998). LSA scored 65%, which is average for foreign students applying for college in the United States. Such a result demonstrates that LSA is able to semantically relate words and need not rely on explicit word pairings.

5.3.2.2 Simulating subject matter knowledge

LSA's ability to simulate subject matter knowledge can be used to determine the coverage of the UAF. If an appropriate space is constructed from HCI literature, we will be able to determine if the UAF covers that space or if we need to expand the UAF.

A study by Landuaer, Foltz, and Laham tests LSA's ability to simulate a psychology student's knowledge (Landauer, Foltz et al., 1998). The authors trained LSA on the psychology book being used by the students in an introductory psychology class and then had LSA take the same exam as the students. LSA scored below the class average, but its score was still high enough to pass the class.

Another study by Foltz, Britt, and Perfetti tested LSA's ability to understand concepts particular to a given subject (Foltz, Britt et al., 1996). More specifically, the authors trained LSA on articles on the Panama Canal and then had it make judgments on the relatedness of 120 pairs of concepts. The same literature and the same task were given to a set of participants who were novices in the domain and a set who were experts. LSA's relatedness judgments correlated more with the experts' judgments.

5.3.2.3 Determining sources

Much like simulating subject matter knowledge, the ability to determine the source of a piece of information can help us to improve the UAF's coverage of essential usability concepts. Work by Foltz, Britt, and Perfetti studied the composition of essays by students (Foltz, Britt et al., 1994). Students read essays on the Panama Canal and were asked to write an essay demonstrating what they had learned. The authors decomposed the essays into propositions and then linked those propositions using LSA to their likely source. Two raters who were experts in terms of the source documents and the domain also linked the propositions to their likely source. The raters had a 63% agreement between them and a 56% and 49% agreement with LSA, showing that LSA was reasonably accurate at determining the source of information.

5.3.2.4 Categorization

LSA has also been used in categorization research. Categorization ability is useful in the scope of this thesis because it can help us automate the process of diagnosis. It can also give us the ability to check problem reports as they are being created to determine if they have enough information about intention to be mapped to a path in the UAF.

A study by Dumais and Nielsen addressed the issue of automatically assigning papers to reviewers (Dumais and Nielsen, 1992). The process involved matching up area of expertise statements submitted by researchers with abstracts for submitted conference papers. The authors created different LSA spaces from the following datasets: the abstracts of submitted papers, journal paper databases, and domain textbooks. The best spaces produced results that were noticeably poorer than manual paper distribution. The variability among spaces was smaller than expected; the worst space constructed only from Schneiderman and Kearsley's book *Hypertext Hands-on! An introduction to a new way of organizing and accessing information* produced a result of 4.9 relevant articles in the top 10 returned articles while the best space constructed from a combination of datasets produced a result of 6.5 relevant articles.

Soto describes a study that had somewhat of the opposite goal from Dumais and Nielsen's study (Soto, 1999). While Dumais and Nielsen were concerned with associating papers with categories that were predefined by the abilities of the reviewers, Soto attempted to create categories for interface labels that describe the enclosed task descriptions. In particular, the study used LSA to develop a range of labels that described a task.

5.3.2.5 Measuring coherence

Measures of coherence are important in helping us understand how well or poorly the nodes of the UAF are worded. The better the coherence of the node, the more direct the process of diagnosis is for the UAF's users.

Research by Foltz, Kintsch, and Landuaer re-analyzed two earlier studies of the role of text coherence in reader' comprehension (Foltz, Kintsch et al., 1998). Both studies determined that increases in coherence lead to increases in user understanding and recall rate. In both studies, LSA measured coherence by treating each sentence as a text unit and then comparing the cosine of the vector of one sentence with that of the following sentence; a general measure for the entire document was obtained by averaging all the values of the pairs. The authors of the first study varied the amount of sentence-to-sentence repetition of key content words in one document to create four forms of the document with increasing coherence. LSA's measures of coherence were significantly correlated with the original authors' coherence measures. The authors of the second study manipulated the original text in more subtle ways by substituting words and phrases of related meaning. Once again, LSA predicted coherence levels consistent with the author's modifications to the text. Because the word overlap was similar for all coherence levels of the document, Foltz, Kintsch, and Landuaer concluded that LSA was able to interpret an improvement in coherence that was due to an improved flow of semantic content.

6 Technical Discussion

6.1 Review and Revise Tuning of the UAF and Evaluation

We have been editing the UAF in a review and revise manner since September of 2003. Our editing has led to three releases of the UAF; version 3.2 was released on October 23, 2003, version 3.3 was released on November 14, 2003, and version 3.4 was released on April 5, 2004.

We began by editing the UAF based on general feedback from previous work. Thereafter, we performed a review after each round of revisions to assist us in assessing the clarity and effectiveness of our changes. The reviews took the form of formative studies and walkthroughs.

6.1.1 Formative Study to Assess Diagnosis with the UAF

We conducted a formative study (heretofore referred to as the diagnosis study) during November and December of 2003 that had two parts. The first part focused on the performance of analyst subjects who were new users of the UAF diagnosing problems with the UAF, and the second part utilized verbal protocol to help us better understand the diagnosis process by analyst subjects who were intermediate users of the UAF.

6.1.1.1 Performance of analyst subjects who were new users of the UAF

The first part of the formative study was intended to get an indication of how well analyst subjects who were new users of the UAF could use it to diagnose usability problems and what we could do to improve the accuracy of diagnoses. The study involved 25 graduate usability engineering students who were new users of the UAF. The students used the UAF to diagnose problems contained in 20 usability problem descriptions. The students had two weeks to complete the diagnoses and did it in a time and place of their choosing.

We gave each student a unique username and password pair for the UAF Problem Reporting Tool (PRT) and told them to use it to report their answers. We entered the 20 problem reports in the PRT as exercise originals. Each exercise original contained a problem description, an expert diagnosis path within the UAF, and a description of the top-level choice. The expert diagnosis path was not visible to students until they had already selected their own diagnosis path and submitted it. Students used the PRT to create their own instances of the exercise originals for submission. When a student created her own instance of an exercise original, she would be presented with a form that contained the problem description and empty text boxes for the diagnosis and explanation of top-level category. The student would then use the UAF Viewer to find the most appropriate path through the UAF for the problem description and paste that path into the form. The student would also provide information as to why she chose the top-level category in the path. The student could revise each diagnosis as much as she liked until she confirmed it. Upon confirmation, the system would present the student with both her diagnosis and the expert's diagnosis from the exercise original. The student then had the opportunity to compare the diagnoses and submit an explanation of the differences.

We compared the students' diagnoses to the expert's diagnoses to determine how similar they were. At the time of the study, the top three levels of the UAF were relatively stable, but the lower levels were still being refined. As a result, we considered the students' diagnoses to match the experts' if they had the same top three levels. Also, even though we used problem reports from professional usability engineering labs, some of the problem descriptions were not clear about the problem or contained more than one problem. We gave credit for a match if students described how flaws in the problem description led to a misdiagnosis or if they diagnosed one of the other problems contained in the description that we had not diagnosed. The students received a point for each match, so the best possible score would have been a 20. The scores for the students ranged from 10 to 19, with an average of 15.

The scores suggested that new users were able to quickly learn the Interaction Cycle because they were able to identify the correct paths at the higher levels of the tree. The scores also confirmed that the top levels of the tree are relatively stable. The poor accuracy of the students at levels below the top three, however, indicated that we needed to further improve the UAF to achieve acceptable levels of accuracy.

One important improvement was understanding and incorporating appropriate inter-category distinguishers. A distinguisher between two UAF Nodes, X and Y, is a set of words that best captures the essential difference between the semantics of X and the semantics of Y. New users were the target audience because they would benefit most from clearly distinguishable choices at each level of the UAF. The comments of the students suggested that they often encountered situations where they had to choose between two nodes at a given level. This pairing process helped them to identify key concepts and make decisions based on the differences between those concepts. If the right words were present, the decision would be much easier.

The study revealed a number of key paths that were difficult for analyst subjects in terms of distinguishing among concepts. We refined the content of the nodes in these paths by adding words and phrases to help UAF users choose among them. For example, the most frequently encountered problem was the difference between the “Translation” and “Physical Actions” nodes. New users had difficulty distinguishing between problems with determining what actions to make and actually making those actions. We corrected the problem by emphasizing the terms knowing, understanding, and meaning in the description for the “Translation” node. Another example problem area was distinguishing between the concepts of false cognitive and physical affordances. The solution was to define cognitive affordances in terms of helping computer users choose the right user interface object and physical affordances in terms of showing computer users how to manipulate a user interface object.

6.1.1.2 *Diagnosis process of analyst subjects who were intermediate users of the UAF*

The second part of the formative study used verbal protocol taken from 6 usability engineering graduate students to help us better understand the diagnosis process. These students were intermediate users of the UAF and had already participated in the first part of the study and had a training session.

We worked with each student for two hours. During a session, a student used the PRT to classify usability problems in the same manner as in the first part of the study. Via verbal protocol, we asked the students to talk us through the node decision process and tell us when they felt that they were confused. The study revealed interesting information about the type and order of information in a node.

At the time of the study, a node’s content consisted of a short description, a long description, and cross references. The description fields included examples that could be specific to the node itself or refer to one of the node’s descendants. The students’ comments revealed several ways in which node content was ineffective. First, none of the students read any of the long descriptions. In fact, the students often read only the first sentence of the short description. As a result, we decided to exclude the long description and make the most important of its content bullets under the short description. Second, when students read the entire short description, they did so to try to find examples. To highlight the examples, we decided that it would be beneficial to have a separate examples field that followed the short description and bullets. Third, the students did not use the cross references because they were located at the bottom of the screen and were often scrolled out of view. One of the students was not even aware that there were cross references. We moved the cross references, so that they are first in the node, where they can more appropriately serve their function of ruling out (or in) alternate choices before proceeding on the current path.

6.1.2 Walkthrough to Confirm New UAF Editing Process

Towards the end of phase 1, Knowledge Analysis Technologies gave us access to an early prototype of a tool for diagnosing problem descriptions. The tool is based on the LSA space constructed from HCI articles, especially usability engineering, that we selected for training purposes. The tool accepts queries in the form of problem descriptions or problem keywords and returns paths in the UAF that are potential diagnoses of the problem.

Our initial work with this tool and the results of the second part of the diagnosis study have confirmed the need for more consistent and precise wording in the UAF. We have addressed this need by making major changes to the UAF editing tool, so that it enforces a more consistent editing process. The tool now uses a series of look-ahead fields to propagate information up the tree from the branches to the root. The look-ahead fields reduce the need to repeat information in nodes and make the content associated with a node more consistent. The look-ahead field is displayed in the parent node that is two levels higher in the tree. Look-ahead fields are used conservatively, so that nodes in the upper levels of the tree do not become overfilled with information for nodes lower in the tree. Each node has look-ahead bullets and examples. A similar approach is used for cross references; the content of cross references is placed in target nodes, so that all references to the same node have the same content. See section 7.2 for a more thorough description of the contents of nodes.

As of this writing, we have completely updated the Planning category of the UAF with the new editor. As a result, the nodes in the branch have been streamlined and are more distinguishable from their siblings and more related to their parents and children. To verify our efforts, we performed a walkthrough of the planning branch with another UAF expert. The expert identified some important problems in the UAF concerning consistency and structure, but also confirmed the efficacy of the new editing process.

The expert pointed out an important inconsistency concerning redirecting UAF users. We had been using dummy nodes in certain places to redirect users to the appropriate stage of the interaction cycle for their problem. The expert noted that having two very different mechanisms, dummy nodes and cross references, which served the same purpose of redirecting users was confusing. We corrected this inconsistency by altering the editing process, so that we removed dummy nodes and treated their contents as cross references.

The expert also noted three structural problems. First, the node titled “Supporting planning for error avoidance” had only one child node titled “Avoid potential post-completion errors.” We removed the child and made its contents an example for the parent. The second problem was that two nodes, “Supporting learning through use, exploration” and “Help users learning – forming correct conceptions, beliefs, and expectations,” had essentially the same content. We corrected the problem by removing the latter of the two nodes because it was more deeply nested. The third problem was that we had combined nodes that address task sequencing and goal decomposition as children of the “Goal decomposition” node. We corrected the problem by creating a new node “Task/step structuring, sequencing” that was a sibling of “Goal decomposition.”

The editor with look-ahead support will allow us to efficiently update the remaining branches of the tree. This refinement of UAF content has proven to be very labor intensive, but it continues apace.

6.2 A Usability Problem Data Analysis Process for Capturing Intention

An important goal of usability problem data analysis is to determine the true nature of problems, which are ultimately dependent upon the immediate intentions that users have when they encounter design flaws. For our purposes, we define the user’s immediate intention to be the kind of action that the user is doing or attempting in terms of the stage within the Interaction Cycle (see section 5.2.1) that the user is operating in when a problem occurs. For example, if a user is having trouble understanding an error message, the intention is associated with the Assessment stage of the Interaction Cycle. In other words, the immediate intention of the user at the time that the problem occurred is to assess the outcome of an action in terms of whether it was successful in moving towards a goal formed in the Planning stage. On the other hand, if a user understands that an error occurred and why it occurred, but is having trouble determining how to recover from it, the intention is associated with the Translation stage of the Interaction Cycle. That is, the user’s immediate intention at the time of problem occurrence is to determine what action to make on what user interface object to translate the plan to recover from the error into an action specification.

We studied usability problem descriptions taken from professional usability labs and discovered that most did not contain clear descriptions of user intention. For example, the following problem description does not connect the stated design flaw to the user by including information about immediate intention: “The tabs at the top of every page [of a digital library website] are ordered so that information-seeking tasks are mixed with other interaction tasks.” If the user’s intention is to understand the site and determine what it can be used for, the user is in the Planning stage of the Interaction Cycle when the problem occurred, and the tab ordering problem is a planning problem. In contrast, if the user has already planned for the task and is simply trying to determine which tab to click on, the user and the tab ordering problem are in the Translation stage. The difference in intention results in two very different diagnoses with potentially two different solutions. Reordering the tabs may be sufficient to solve a problem in the Translation stage because the user has already formulated a goal and developed a high-level task sequence to accomplish that goal. An error in the Planning stage, however, may require additional organization of the tabs, possibly into groups labeled by high-level task and work flow categories.

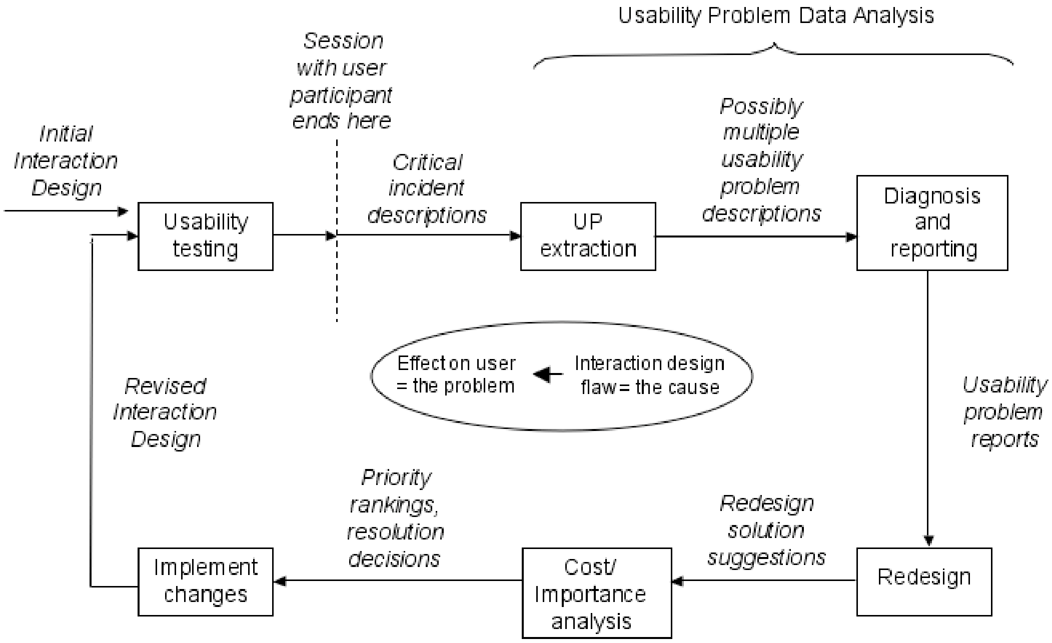


Figure 6: Existing data management cycle (slight adaptation of Figure 4)

The lack of intention in the problem descriptions from professional usability labs caused us to question whether it was possible with existing processes (e.g. Figure 6) to include such information in problem descriptions. Clearly, we could not expect this kind of information to appear in ad hoc usability problem reports. We concluded that is necessary to move some of the analysis and diagnosis work forward to the usability problem collection stage to reduce the data loss that occurs in the current process. The part to be moved forward would have to be the minimum amount of analysis to determine and document users' intentions. We believe that this is a crucial conclusion, and it has reshaped our thinking about how usability problem data analysis should be performed within the overall usability data management life cycle. Figure 7 shows how the usability data management cycle can be adapted to accommodate moving some usability problem analysis earlier in the cycle.

In the existing cycle in Figure 6, the typical failure to capture user intention information during usability testing or immediately thereafter, often stands in the way of successful analysis – extraction, diagnosis, and reporting. Because the user participant who helped with usability testing is now gone and because the person who is analyzing the data may be different than the person who conducted the usability testing, it is not always possible to recapture the necessary intention information.

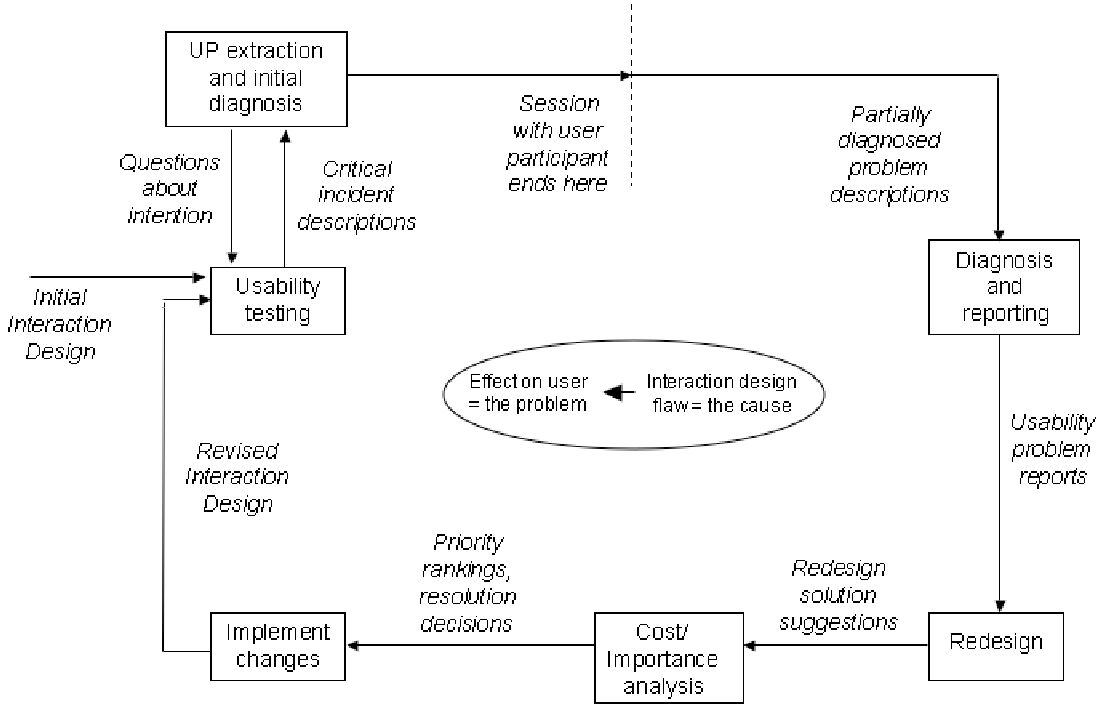
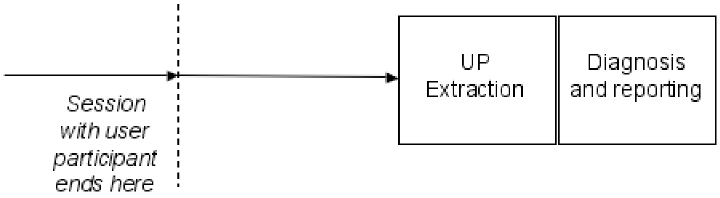


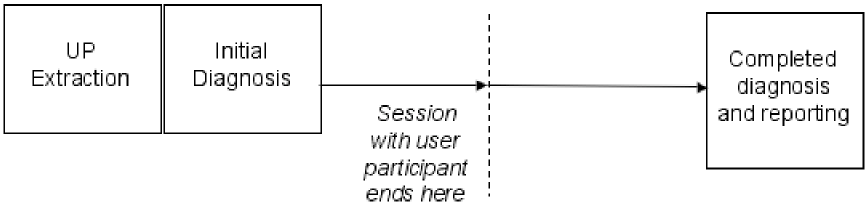
Figure 7: Proposed usability data management cycle

The proposed usability engineering process in Figure 7 features rapid, local iteration between usability data collection and initial analysis, so that the usability testing facilitator can work with the subject to clearly identify and record intention for usability problem extraction and initial diagnosis. Having captured the necessary intention information, the practitioner can complete problem diagnosis and reporting after the testing subject is gone.

Figure 8 illustrates the changes to usability problem data analysis that would result from performing initial diagnosis in an iterative fashion with data collection. The key difference is that problem extraction and high-level diagnosis are performed while the user participant is present.



Usability problem data analysis in the existing cycle



Usability problem data analysis in the proposed cycle

Figure 8: How analysis is divided in the proposed process

The key to limiting the added time necessary while the user participant is still present and, correspondingly to limiting the added cost of the proposed change, is increasing the speed of initial diagnosis. From early indications, we believe that the Wizard will provide some measure of speed and a great deal of accuracy, but LSA will most likely provide largest increase in speed by automating some of the diagnosis process. In phase 2, we will determine how much of the diagnosis process we can automate and what level of speed we can achieve (see section 7). In sum, it is our working hypothesis that the proposed process will require a greater initial investment in terms of time, but it will save much effort in later steps of the usability engineering process.

6.3 Tools and Evaluation

Previous tools have leveraged the UAF as a conceptual framework. We used these tools as a source for ideas, but we have created our own generation of tools. Colaso created a tool for usability inspection that helped practitioners identify potential problems in an interaction design by asking questions based on concepts in the UAF (Colaso, Hartson et al., 2003). In addition, Virginia Tech computer science students have created prototypes of the following tools during independent studies: Ideal, a tool for collecting usability problem data; the Usability Problem Database, a tool for organizing usability problem reports; and the UAF Explorer, a tool for viewing the content of nodes.

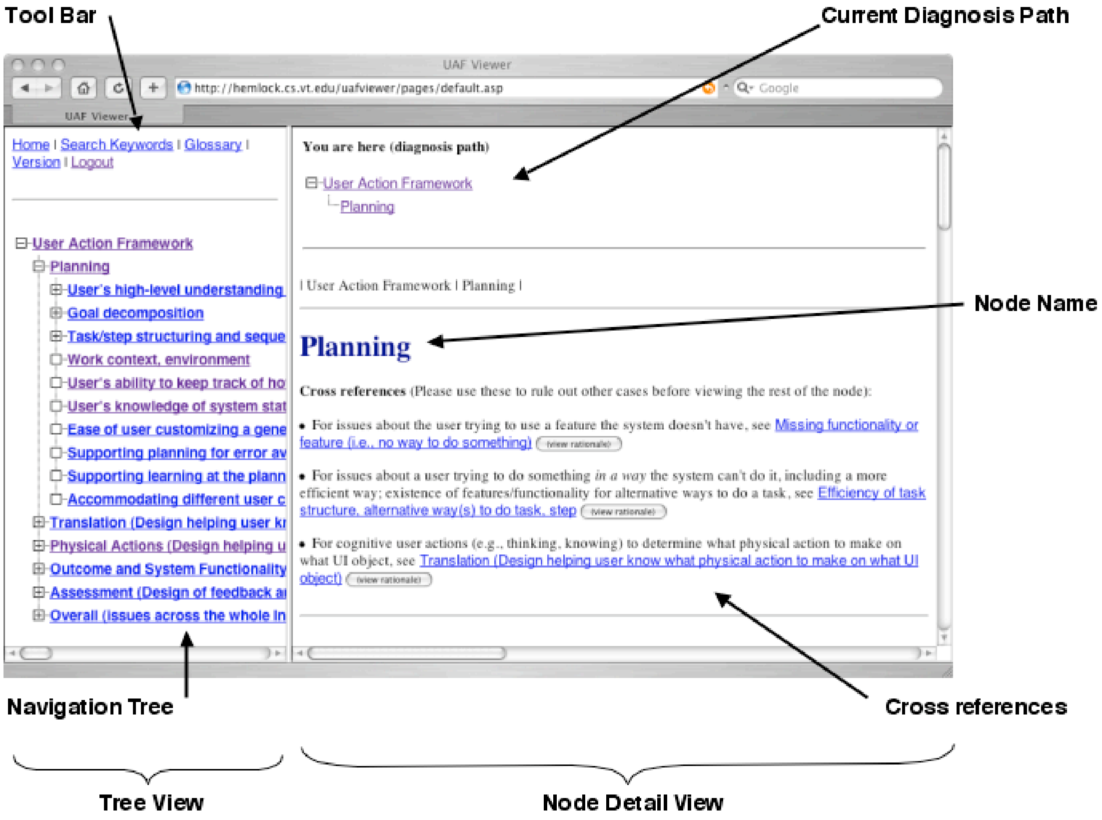
We have designed and prototyped three major tools that leverage the UAF as a theoretical framework and guide practitioners through usability problem data analysis: the UAF Viewer, the UAF Editor, and the Usability Problem Reporting Tool (UAF PRT). The tools are written in Active Server Pages (ASP) and linked to Access databases; they are web accessible. In this section, we provide screenshots, descriptions, and evaluations of each of the tools.

6.3.1 UAF Viewer

The UAF Viewer allows practitioners to navigate the tree structure of the UAF. In this section, we describe the Viewer, its evaluation and redesign, and two of its applications.

6.3.1.1 Description

Figure 9 – Figure 11 are screenshots of the viewer. All the screenshots are of the Planning node, but each contains a different section of content that is displayed as the right-hand frame is scrolled.



Cross references (Please use these to rule out other cases before viewing the rest of the node):

- For issues about the user trying to use a feature the system doesn't have, see Missing functionality or feature (i.e., no way to do something)
- For issues about a user trying to do something *in a way* the system can't do it, including a more efficient way; existence of features/functionality for alternative ways to do a task, see Efficiency of task structure, alternative way(s) to do task, step
- For cognitive user actions (e.g., thinking, knowing) to determine what physical action to make on what UI object, see Translation (Design helping user know what physical action to make on what UI object)

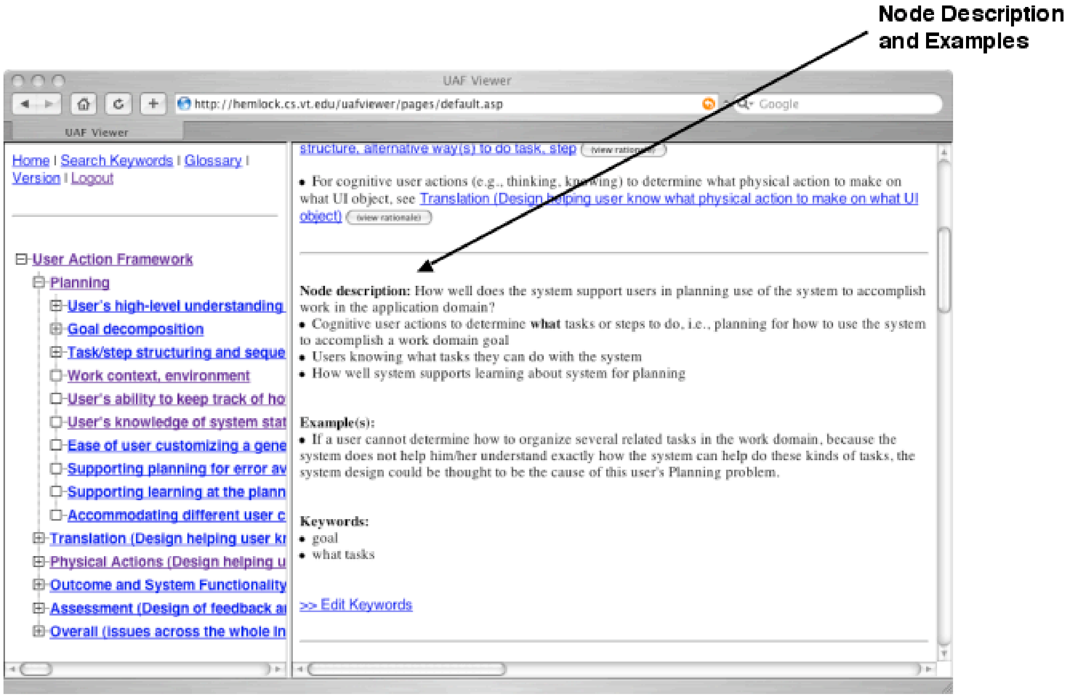
Figure 9: UAF Viewer (1)

Figure 9 shows the two major parts of the Viewer: the tree view and the node detail view. The tree view allows practitioners to work with the UAF at a higher level. The tool bar in the upper left corner contains options that apply to the entire UAF. The home option returns the user to the Viewer welcome screen, which briefly describes the tool and the content associated with nodes. The search keywords option allows a practitioner to locate nodes associated with keywords. Although we plan to supply the UAF with a few baseline keywords, the Viewer allows practitioners to add keywords of their choice to any node. For example, a practitioner may add the keyword “wording” to the Translation node if she discovers that many of the problems that she identifies during usability testing deal with the wording of labels, buttons, or dialogs as they affect a participant’s ability to determine how to perform an action. Glossary, the next option in the toolbar, contains definitions for frequently used terms, such as user interface object or affordance. Terms in the glossary appear as hyperlinks in node content, so that practitioners can quickly link to the definitions. The version option contains information about the current version of the UAF and how it differs from previous versions.

The navigation tree on the left-hand side allows practitioners who are familiar with the UAF to quickly traverse it. Practitioners that are not familiar with the UAF can traverse the tree using the node detail view. The tree is modeled after the Windows Explorer tree view and uses minus signs for expanded nodes, plus signs for expanding nodes with children, and empty boxes for terminal nodes. Selecting the hyperlink for a node in the navigation tree will display the content of that node in the node detail view. Selecting the box to the left of the hyperlink will perform the appropriate action on the navigation tree, such as expanding a node with a plus, without refreshing the node detail view.

The screenshot shows what is displayed in the node detail view when a practitioner first selects a node. The current diagnosis path uses a horizontal tree similar to the navigation tree to show the practitioner's current position. Directly underneath the current diagnosis path is a list of the nodes on the current path delimited with vertical bars. This representation of the path exists, so that practitioners can quickly copy a path and paste it into a problem report in the PRT. Future versions of the tool will integrate the Viewer and the PRT, so that the list is no longer necessary.

The node's name is displayed under the current diagnosis path followed by cross references. Not all nodes have cross references. We decided to show the cross references before actual node content because we wanted to immediately redirect practitioners who had incorrectly arrived at a node. Each cross reference contains two pieces of information: the high-level cross reference description of the target node and the rationale. The high-level cross reference description is pulled from the target node for consistency; because each node is cross referenced with the same text, practitioners can quickly identify key nodes and what distinguishes one from another. The rationale is specific to the cross reference and tells the practitioner why the target node may better describe the usability problem. The rationale is hidden and must be displayed with the view rationale button to limit the amount of information that a practitioner must initially process.


Node Description and Examples

structure, alternative way(s) to do task, step

- For cognitive user actions (e.g., thinking, knowing) to determine what physical action to make on what UI object, see Translation (Design helping user know what physical action to make on what UI object)

Node description: How well does the system support users in planning use of the system to accomplish work in the application domain?
- Cognitive user actions to determine **what** tasks or steps to do, i.e., planning for how to use the system to accomplish a work domain goal
- Users knowing what tasks they can do with the system
- How well system supports learning about system for planning

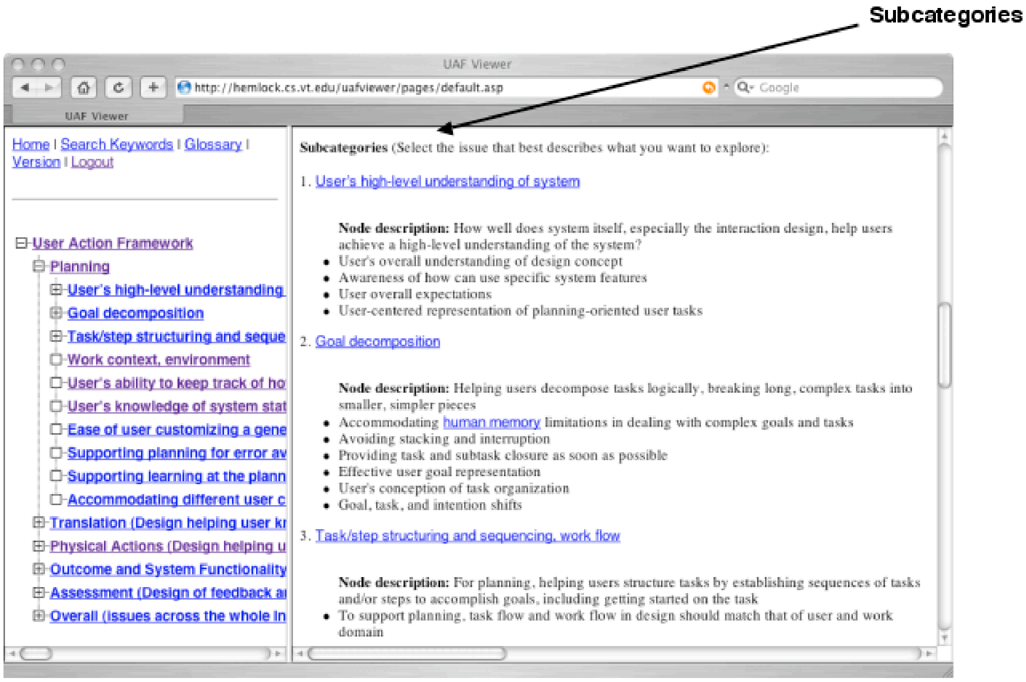
Example(s):
- If a user cannot determine how to organize several related tasks in the work domain, because the system does not help him/her understand exactly how the system can help do these kinds of tasks, the system design could be thought to be the cause of this user's Planning problem.

Keywords:
- goal
- what tasks

>> Edit Keywords

Figure 10: UAF Viewer (2)

Figure 10 shows the node detail view after it has been scrolled. The node description and examples are displayed under the cross references. The node description consists of a brief overview that describes the node at a high level and bullets that contain more detailed descriptions. One of the bullets may be designated as a look-ahead description bullet that is displayed in the subcategory listing of the parent node. The look-ahead description bullet helps to guide practitioners down a particular path to a node. Future versions of the tool will have links to specific paragraphs of HCI literature selected by LSA that describe the usability concept associated with the node. The examples are problems that would be classified in the node. Like description bullets, an example may be classified as a look-ahead example. Future versions of the tool will provide examples by linking to problems that have already been diagnosed and are stored in the PRT. The keywords section shows which keywords have already been associated with the node and provides the option to create and associate new keywords. As discussed, keywords are the only searchable part of the tree. Because the UAF is not a taxonomy, the same usability concept may appear in multiple places. The meaning of the concept in terms of a usability problem is based on the entire path. As a result, it is not logical to allow practitioners to search node descriptions and examples to go directly to a node for diagnosis without following a path.



Subcategories (Select the issue that best describes what you want to explore):

1. User's high-level understanding of system

Node description: How well does system itself, especially the interaction design, help users achieve a high-level understanding of the system?

- User's overall understanding of design concept
- Awareness of how can use specific system features
- User overall expectations
- User-centered representation of planning-oriented user tasks

2. Goal decomposition

Node description: Helping users decompose tasks logically, breaking long, complex tasks into smaller, simpler pieces

- Accommodating human memory limitations in dealing with complex goals and tasks
- Avoiding stacking and interruption
- Providing task and subtask closure as soon as possible
- Effective user goal representation
- User's conception of task organization
- Goal, task, and intention shifts

3. Task/step structuring and sequencing, work flow

Node description: For planning, helping users structure tasks by establishing sequences of tasks and/or steps to accomplish goals, including getting started on the task

- To support planning, task flow and work flow in design should match that of user and work domain

Figure 11: UAF Viewer (3)

Figure 11 shows the node detail view after it has been further scrolled. The subcategories follow the node description and examples. Subcategories are children of the active node. Each child is displayed with the high-level description and description bullets, including any look-ahead description bullets from its children. We do not include examples associated with the children to minimize display space. Each child's name is a hyperlink that displays it in the node detail view.

6.3.1.2 Evaluation and Redesign

The original design of the Viewer displayed a node short description, long description, and cross references. The short description mixed descriptive content, examples, and look-ahead fields. The look-ahead fields could refer to nodes any number of levels deeper in the tree. Practitioners had to click a button to access the long description, similar to how the view rationale button currently functions. The diagnosis study revealed that this initial design was not effective. The analyst subjects never read the long description and rarely reached the cross references because they were displayed at the bottom. Also, analyst subjects often focused on a look-ahead description or example that referred to a node several levels deeper in the tree, but were unable to find the node because of the lack of continuation of the look-ahead.

6.3.1.3 Applications

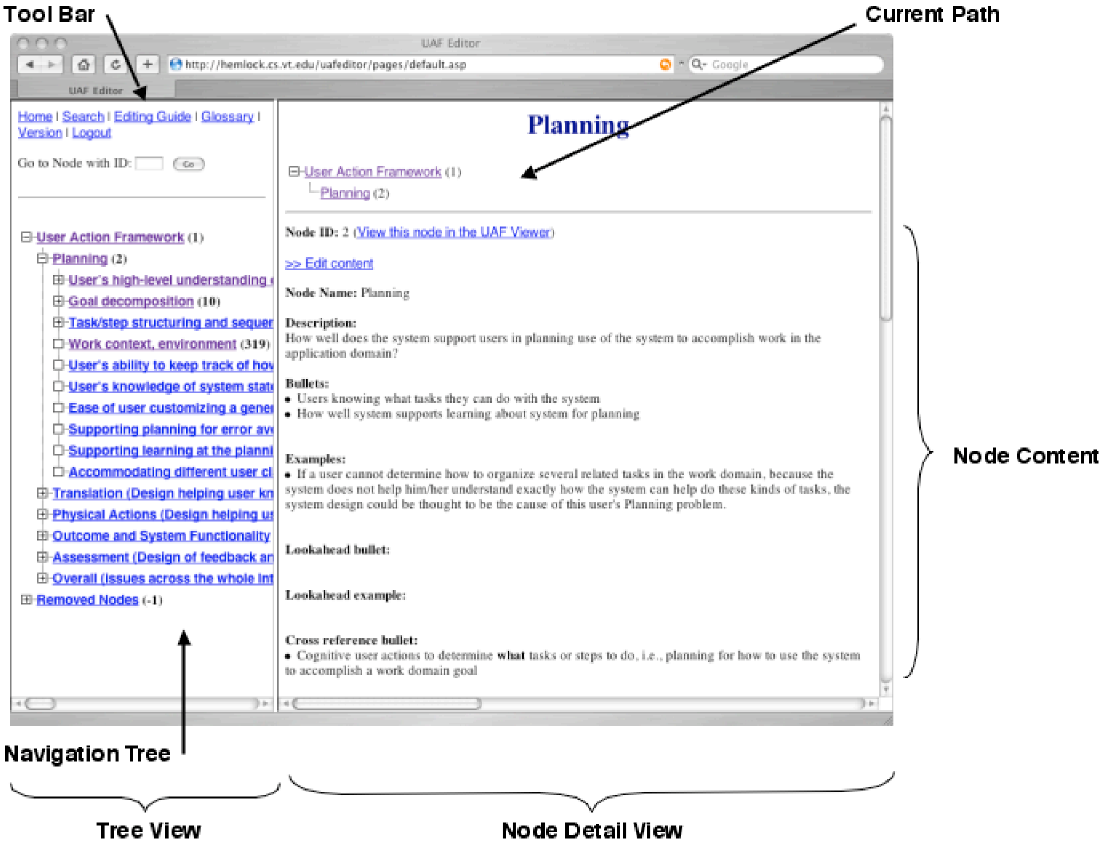
We designed the Viewer to be flexible enough to be used both for diagnosis of problems collected during usability testing and for usability inspection. For diagnosis, it helps practitioners distinguish between usability concepts and guides them to a node that best describes their problem. This node contains enough information to relay to designers the cause of the problem and potential design solutions. For inspection, the Viewer presents the UAF as a number of usability concepts that might be applicable to the system being evaluated. The practitioners can skip nodes that are not applicable, thereby pruning usability concepts.

6.3.2 UAF Editor

The UAF Editor allows practitioners to edit the structure and content of the UAF. The Editor is intended to be a tool for trained, experienced UAF users and therefore places the most emphasis on efficiency. We describe the Editor with the help of a series of screenshots, discuss the evaluation and redesign cycle, and give applications of the tool.

6.3.2.1 Description

Figure 12 and Figure 13 are screenshots of the Planning node in the Editor, the same node as in the Viewer screenshots. The tool is similar in some respects to the Viewer, but it represents a different way of thinking about node content. A practitioner using the Viewer is trying to locate a node that best describes a usability problem, while a practitioner using the Editor is trying to develop or update node content.



Node ID: 2 (View this node in the UAF Viewer)

>> Edit content

Node Name: Planning

Description:
How well does the system support users in planning use of the system to accomplish work in the application domain?

Bullets:
- Users knowing what tasks they can do with the system
- How well system supports learning about system for planning

Examples:
- If a user cannot determine how to organize several related tasks in the work domain, because the system does not help him/her understand exactly how the system can help do these kinds of tasks, the system design could be thought to be the cause of this user's Planning problem.

Lookahead bullet:

Lookahead example:

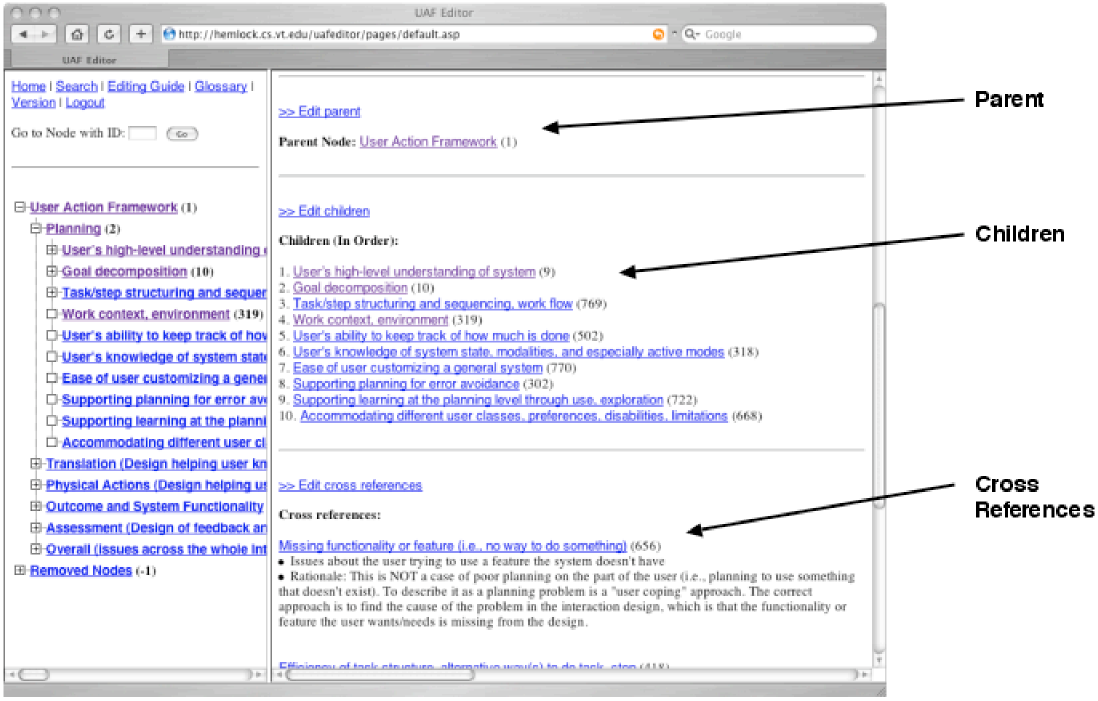
Cross reference bullet:
- Cognitive user actions to determine **what** tasks or steps to do, i.e., planning for how to use the system to accomplish a work domain goal

Figure 12: UAF Editor (1)

Figure 12 shows the tree and node detail views for the Editor. The top left corner of the tree view is a tool bar that has options that apply to the entire tree. The search option allows a practitioner to search the content of the nodes. The Viewer limits searching to keywords because of the nature of diagnosis with the UAF, but the Editor allows searching of all content associated with nodes to help in quickly locating a specific node. The editing guide allows the practitioner to keep notes on the editing process to promote consistency. For example, the practitioner may use this feature to create an informal style guide. The glossary option allows the practitioner to edit the glossary, and the version link allows the practitioner to record major changes and differences with previous versions of the UAF. Finally, the “Go to node with ID” field allows a practitioner to type in a node’s identification number to quickly load that node in the node detail view. The nodes’ numbers are listed beside their names in the navigation tree as well as in the node detail view.

The navigation tree is similar to the one in the Viewer, except that the Editor also contains nodes that have been removed from the tree; “Removed Nodes” appears as a sibling of the “User Action Framework” node. While editing, a practitioner may want to remove a node from the UAF without deleting that node. The node remains as a child of “Removed Nodes” until it is deleted. Removed nodes may be put back in the tree by specifying a parent other than “Removed Nodes.” The content of a removed node also remains, so that it can be copied into other existing nodes or newly created nodes. When a node is removed, the Editor automatically takes care of cross references that would become unattached by the removal of the node.

The node detail view is organized differently from the Viewer. The Editor begins with fields that compromise the majority of the node’s content including a general description, short descriptive bullets, examples, and look-ahead fields. This information is associated with a node to promote consistency. The practitioner only needs to edit the content for a node in one place, and then the Viewer takes care of details such as displaying look-ahead fields in the correct locations in the tree.



>> Edit parent

Parent Node: User Action Framework (1)

>> Edit children

Children (In Order):

1. User's high-level understanding of system (9)
2. Goal decomposition (10)
3. Task/step structuring and sequencing, work flow (769)
4. Work context, environment (319)
5. User's ability to keep track of how much is done (502)
6. User's knowledge of system state, modalities, and especially active modes (318)
7. Ease of user customizing a general system (770)
8. Supporting planning for error avoidance (302)
9. Supporting learning at the planning level through use, exploration (722)
10. Accommodating different user classes, preferences, disabilities, limitations (668)

>> Edit cross references

Cross references:

Missing functionality or feature (i.e., no way to do something) (656)

- Issues about the user trying to use a feature the system doesn't have
- Rationale: This is NOT a case of poor planning on the part of the user (i.e., planning to use something that doesn't exist). To describe it as a planning problem is a "user coping" approach. The correct approach is to find the cause of the problem in the interaction design, which is that the functionality or feature the user wants/needs is missing from the design.

Figure 13: UAF Editor (2)

Figure 13 shows the node detail view after it has been scrolled past the node content. The next section shows the parent node. The practitioner can move the node anywhere in the tree by specifying a new parent. The editor automatically handles updating the tree structure to account for the move. Underneath the parent section is the children section. A practitioner may add new children or reorder existing children. The last section contains cross references. The practitioner may add or remove cross references, reorder cross references, or edit the rationale for cross references.

6.3.2.2 Evaluation and Redesign

Unlike the Viewer, which went through one longer evaluation and redesign cycle, the Editor has gone through several shorter cycles. The earliest versions of the Editor had basic capabilities for editing short and long descriptions and cross references. The results of the diagnosis study led us to reevaluate the structure of content in the UAF, and we updated the Editor to match that structure by breaking up the content into description, description bullet, example, and look-ahead fields. Thereafter, discussions with experts and studies utilizing verbal protocol helped us realize the need for a search over all fields including cross references, an editing guide, a glossary, and a way to locate nodes by node number. Finally, a second evaluation with experts led to small changes such as moving the editing link to the top of a section to better support task flow and automatically handling dangling cross references after node removal.

6.3.2.3 Applications

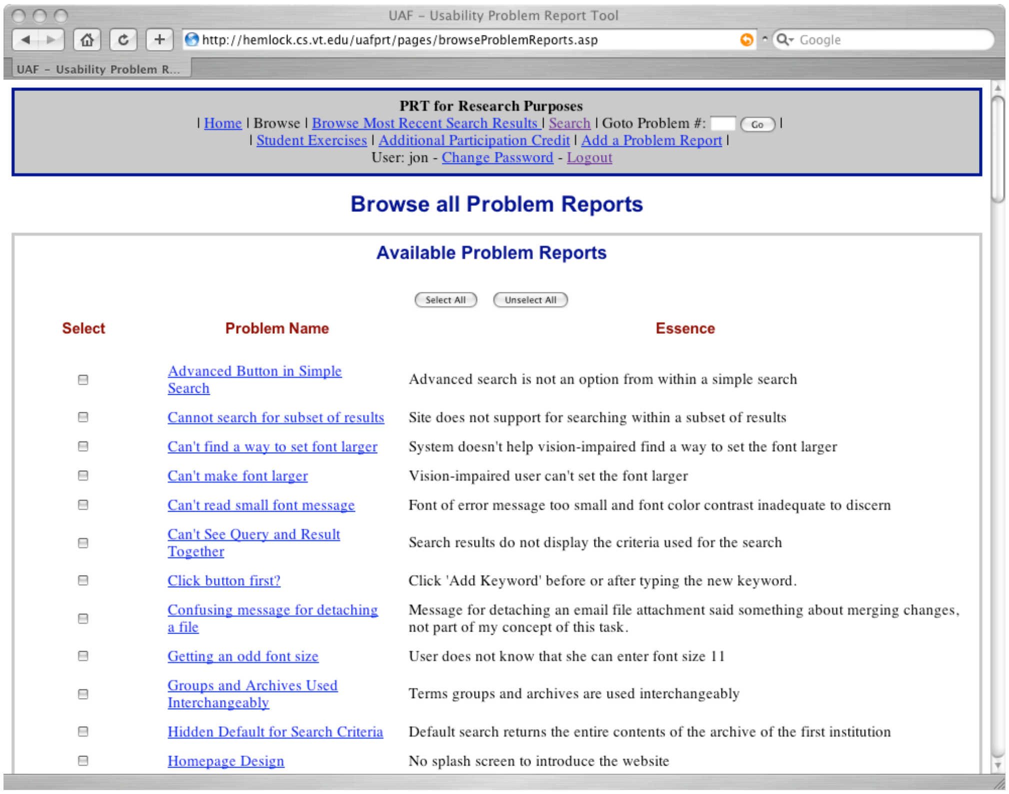
The Editor has primarily been used to perform review and revise tuning of the existing UAF, but practitioners can use it to extend the UAF to new interaction styles. Because the UAF is user-centered and not built around a particular design space, it can be modified to include other interaction styles. For example, the UAF may be extended to work with handheld devices such as PDAs, for which certain usability concepts have different meanings or importance as compared to desktop environments.

6.3.3 UAF PRT

The UAF PRT is intended to facilitate usability data management for analysis and reporting purposes. The tool helps practitioners catalog problem data in a manner that facilitates the identification of patterns or process dependencies both within a development effort and among multiple development efforts. The primary object in the PRT is the problem report, which contains a description of the problem, diagnosis information, and other supporting information such as designer knowledge. In this section, we describe the most important features of the tool and discuss how studies have influenced its development. We conclude with a brief discussion of plans to extend the PRT to better support problem collection.

6.3.3.1 Description

Figure 14 – Figure 16 illustrate three important features of the PRT: browsing existing reports, searching existing reports, and adding a new report. The navigation bar at the top of the screen is available on all pages, and the option for the currently selected page is not an active link. Each problem has a unique identification number and may be quickly loaded with the go to problem # field. The links for student exercises and additional participation credit were added when we modified the PRT to help support the teaching of usability problem diagnosis in graduate level classes. The actual details of these teaching features are outside the scope of this thesis.



PRT for Research Purposes

| Home | Browse | Browse Most Recent Search Results | Search | Goto Problem #: |
| Student Exercises | Additional Participation Credit | Add a Problem Report |
User: jon - Change Password - Logout

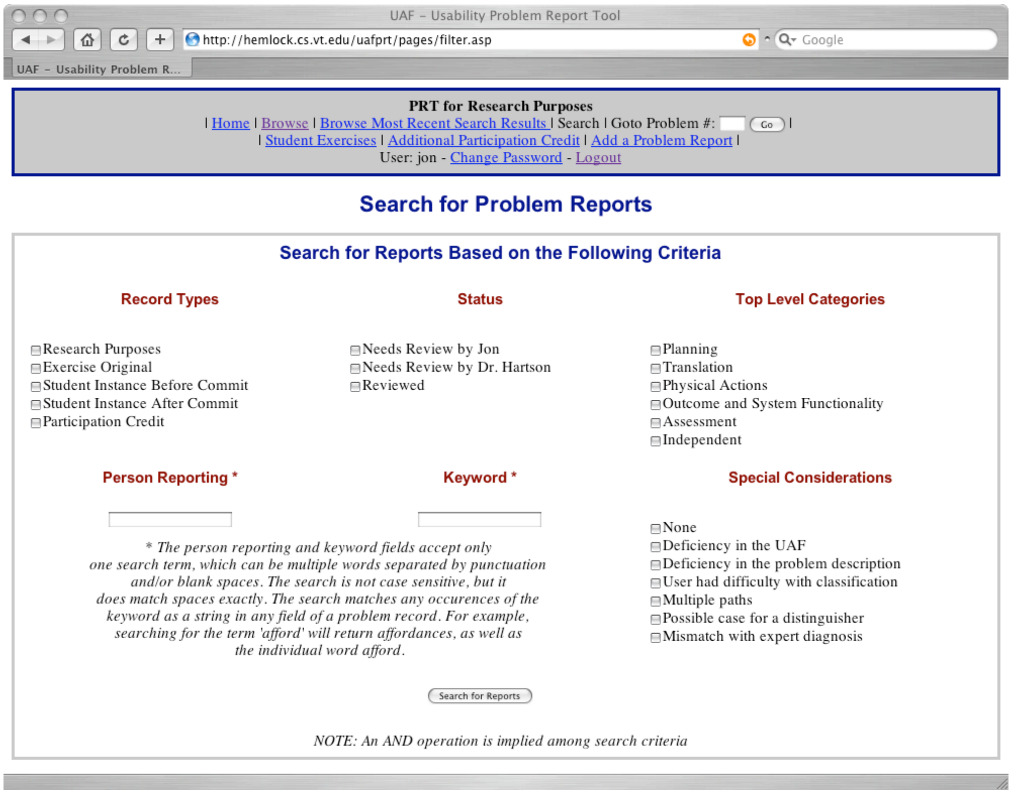
Browse all Problem Reports

Available Problem Reports

Select	Problem Name	Essence
☐	Advanced Button in Simple Search	Advanced search is not an option from within a simple search
☐	Cannot search for subset of results	Site does not support for searching within a subset of results
☐	Can't find a way to set font larger	System doesn't help vision-impaired find a way to set the font larger
☐	Can't make font larger	Vision-impaired user can't set the font larger
☐	Can't read small font message	Font of error message too small and font color contrast inadequate to discern
☐	Can't See Query and Result Together	Search results do not display the criteria used for the search
☐	Click button first?	Click 'Add Keyword' before or after typing the new keyword.
☐	Confusing message for detaching a file	Message for detaching an email file attachment said something about merging changes, not part of my concept of this task.
☐	Getting an odd font size	User does not know that she can enter font size 11
☐	Groups and Archives Used Interchangeably	Terms groups and archives are used interchangeably
☐	Hidden Default for Search Criteria	Default search returns the entire contents of the archive of the first institution
☐	Homepage Design	No splash screen to introduce the website

Figure 14: UAF PRT (1)

Figure 14 is a screenshot of browsing for a problem. Browsing displays the name and essence of all problem reports in the database. The essence is a short description of the problem that contains just enough information to highlight the problem. A practitioner can select multiple problems and generate a summary report of them or delete them. A practitioner can also view a detailed version of an individual problem report by selecting the report's name. In the detail view, the practitioner has the option of editing the report.



PRT for Research Purposes
| Home | Browse | Browse Most Recent Search Results | Search | Goto Problem #: ☐ Go |
| Student Exercises | Additional Participation Credit | Add a Problem Report |
User: jon - Change Password - Logout

Search for Problem Reports

Search for Reports Based on the Following Criteria

Record Types

☐ Research Purposes
☐ Exercise Original
☐ Student Instance Before Commit
☐ Student Instance After Commit
☐ Participation Credit

Status

☐ Needs Review by Jon
☐ Needs Review by Dr. Hartson
☐ Reviewed

Top Level Categories

☐ Planning
☐ Translation
☐ Physical Actions
☐ Outcome and System Functionality
☐ Assessment
☐ Independent

Person Reporting *

Keyword *

** The person reporting and keyword fields accept only one search term, which can be multiple words separated by punctuation and/or blank spaces. The search is not case sensitive, but it does match spaces exactly. The search matches any occurences of the keyword as a string in any field of a problem record. For example, searching for the term 'afford' will return affordances, as well as the individual word afford.*

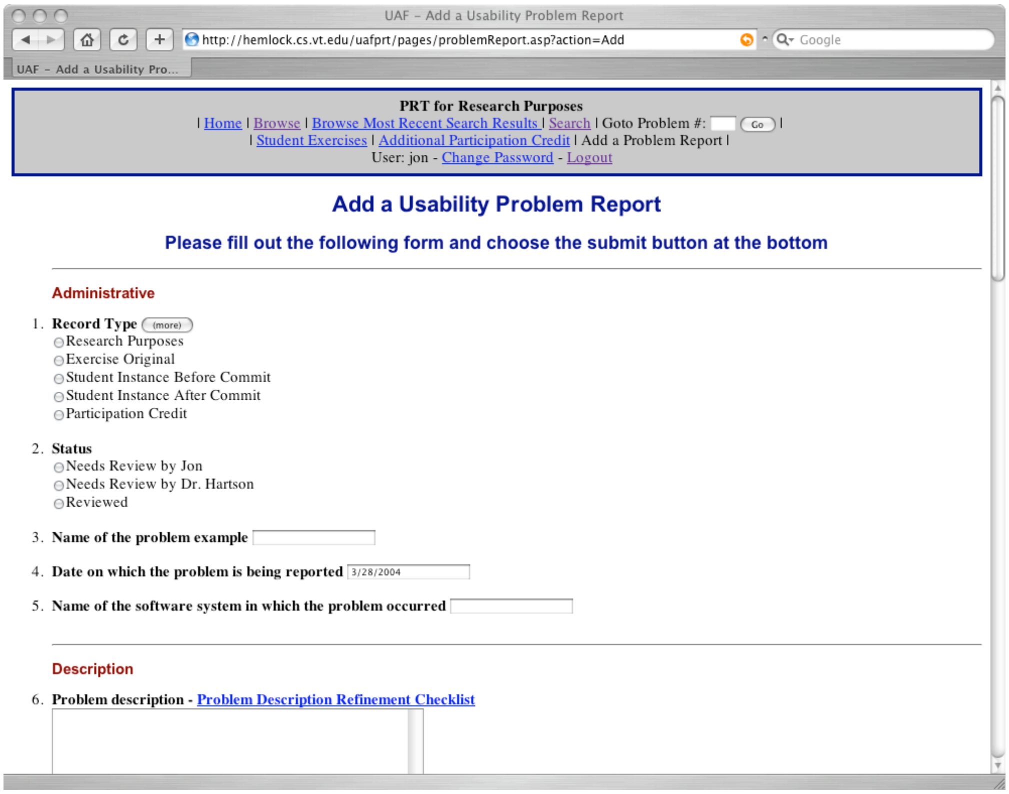
Special Considerations

☐ None
☐ Deficiency in the UAF
☐ Deficiency in the problem description
☐ User had difficulty with classification
☐ Multiple paths
☐ Possible case for a distinguisher
☐ Mismatch with expert diagnosis

NOTE: An AND operation is implied among search criteria

Figure 15: UAF PRT (2)

Figure 15 is a screenshot of the search interface. Because we created the PRT to help support practitioners in the management and analysis of problem data, an efficient and thorough search mechanism was critical. The search is organized, so that the most commonly searched fields are checkboxes. For example, the top level categories group of checkboxes allows a practitioner to rapidly search by stages in the Interaction Cycle. In addition to the checkbox groups, the search interface has a general purpose keyword search, which searches all fields in problem reports. The search interface returns results in a form that is similar to the browse view in Figure 14. After a search has been performed, a practitioner can quickly get back to the results by selecting the “Browse most recent search results” option in the navigation bar.



PRT for Research Purposes
| Home | Browse | Browse Most Recent Search Results | Search | Goto Problem #: ☐ Go |
| Student Exercises | Additional Participation Credit | Add a Problem Report |
User: jon - Change Password - Logout

Add a Usability Problem Report

Please fill out the following form and choose the submit button at the bottom

Administrative

1. **Record Type** (more)
   ○ Research Purposes
   ○ Exercise Original
   ○ Student Instance Before Commit
   ○ Student Instance After Commit
   ○ Participation Credit
2. **Status**
   ○ Needs Review by Jon
   ○ Needs Review by Dr. Hartson
   ○ Reviewed
3. **Name of the problem example** ☐
4. **Date on which the problem is being reported** 3/28/2004
5. **Name of the software system in which the problem occurred** ☐

Description

6. **Problem description** - Problem Description Refinement Checklist

Figure 16: UAF PRT (3)

The screenshot in Figure 16 shows the interface for adding a new problem report. The form has three basic sections: administrative, description, and diagnosis. The administrative section records general information such as the name of the practitioner making the report, the date on which it was reported, the name of the system in which the problem occurred, and who has reviewed the report. The description section contains descriptions of the problem and screenshots or screen action videos. Finally, the diagnosis section contains the actual diagnosis in terms of a path through the UAF and other information such as possible missing usability concepts in the UAF or a lack of specification in the problem description.

6.3.3.2 Evaluation and Redesign

Our early work with diagnosis helped us to realize that the problem reports that came out of industry were inadequate. We hypothesized that they did not include enough information and decided to put fields into the PRT to capture more information. The following fields were included in the original PRT problem report in addition to the fields listed in Table 1:

- Areas for special consideration
- If there are special considerations, please describe them
- Comments on problem type, cause(s), usability principles involved
- Top level category

- Justification for the top level category choice (words/phrases from the problem description)
- Justification for the top level category choice (words/phrases from the top level category descriptions)
- What was the user trying to accomplish?
- What did the user actually do?
- Why the user did what he/she did?
- What was the result, what went wrong?
- Psychological effects
- Physical effects
- Perceptual effects
- Task-related effects
- Associated words
- General comments
- Recommendations for further evaluation

The fields helped us to collect more information, but the information was not translating into more accurate diagnosis. In addition, analyst subjects with little usability engineering experience had trouble filling out fields such as psychological, physical, perceptual, and task-related effects because they simply did not know what to look for during usability testing. We also found that the forms were tedious and that users were less thorough each time they completed a form.

The diagnosis study, however, changed our understanding of what is important in problem reports. We realized that it was not necessary to get more information, but instead it was necessary to get the correct information, namely the user’s intention. We redesigned the PRT and significantly reduced the number of fields in the problem report to those listed in Table 1. We have further evaluations of the PRT scheduled for the fall of 2004, so that we can integrate the findings from our summer evaluations of LSA technology.

Fields included in a problem report

- Session
- Date
- Project
- Version
- Evaluator
- User class
- Associated usability specification
- Associated benchmark task
- Problem description
- Screen images and/or screen action video clips
- Diagnosis and rationale
- Designer knowledge
- Severity rating
- Alternative solutions
- Estimated cost to fix
- Cost/importance analysis
- Priority-to-fix ranking
- Decision to fix or not
- Management approval signoffs
- Actions taken
- Actual cost of fix
- Resulting effect on usability in further testing with computer users

Table 1: Fields included in a problem report

6.4 Wizard and Evaluation

Our work with usability problem data analysis and the results of the diagnosis study confirmed the need for an early process for identifying intention. One of the most serious problems in usability problem analysis is missing usability data – information that is lost between collection and design, but that is needed to answer crucial questions to establish distinguishability. While the usability practitioner gathering usability data during lab-based usability testing may observe the necessary data needed later to make an effective diagnosis, without tool support to elicit this information very soon after observation, that information is lost and will not be available when needed for later complete diagnosis. We propose that the Wizard tool can be used in conjunction with data collection to elicit missing information crucial to diagnosis.

Rather than presenting the user with multiple complex choices at the top level of the UAF (the crucial diagnosis decision point), the Wizard only presents two simple choices at a time and emphasizes the most important distinguishers between the choices. The Wizard is based on a ruling-out strategy that quickly eliminates the unusual and outlier cases, such as the Outcomes and Overall branches of the UAF. These two cases occur rarely during usability testing, and it is convenient for the Wizard to get them out of the way early in the process.

Figure 17 depicts the ruling-out strategy. Each black node represents a decision point where the user chooses between a given stage in the Interaction Cycle and all the remaining stages. The users start the Wizard by choosing between the Outcome and System Functionality stage and the rest of the Interaction Cycle.

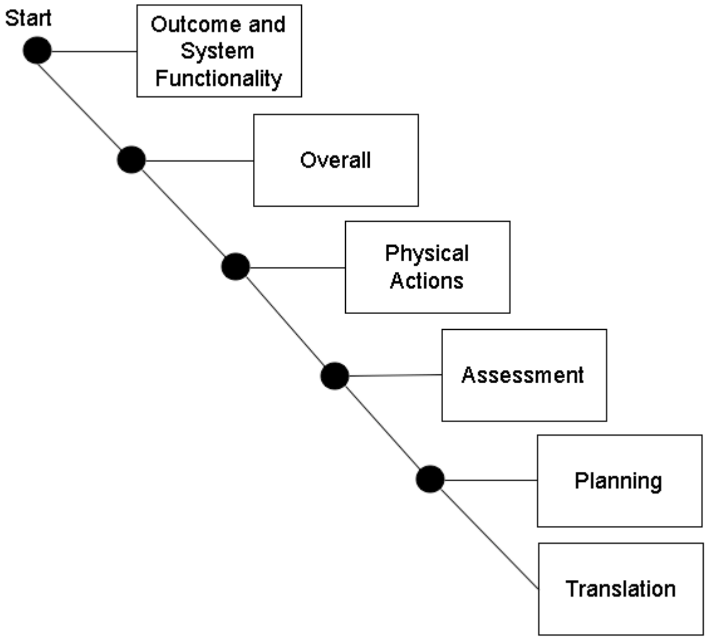


Figure 17: Wizard's ruling-out strategy

The Wizard brings the right distinguisher to bear at the right time and the right place for the usability problem analyst. While the distinguishers needed might exist somewhere among the words in the node descriptions, the Wizard brings them to bear in a direct face-off situation, distilling one step of diagnosis down to an A vs. B question based on a key characteristic of the usability problem.

We developed two static prototype versions of the Wizard (Table 2 and Table 3) in February 2004 to determine feasibility and to verify that it can improve later diagnosis accuracy. The Wizard study, which is described in the next section, is an evaluation of these static versions of the Wizard.

6.4.1 Formative Study of the Wizard

The results of the diagnosis study and the development of the static Wizard prompted us to run a study in March of 2004 to verify the Wizard (heretofore referred to as the Wizard study). The analyst subjects, who all had some general usability knowledge, included an individual who had never used the UAF, two beginning users, one intermediate user, and two experts. We wanted a range of user experience levels that would match the experience levels of all potential users. The analyst subjects were given 10 usability problem descriptions with varying levels of user intention specified. After the analyst subjects read the problem description, we first asked them to choose a top-level category in the UAF and then had them use the Wizard to establish intention. We had two versions of the Wizard: the first version (Table 2) had only abstract descriptions of each stage in the interaction cycle, and the second version (Table 3) combined the abstract descriptions with examples.

The Wizard presented the analyst subject with a screen that had two choices in side-by-side boxes. For this first study, the choice that directly mapped to a stage in the Interaction Cycle was always on the left, and the choice that continued to the remaining stages was on the right. When analyst subjects chose a box on the left, we did not tell them which stage in the Interaction Cycle they had selected.

For clarity, we are including the stages of the Interaction Cycle in parentheses. The analyst subjects did not see how the questions map to these stages. If a user chose a right-hand box, she was presented with the next pair of boxes.

Is your problem one that is internal to the system and invisible to users? (Outcome and System Functionality)	Does your problem concern the user’s interaction with the user interface?

Is your problem independent of the interaction cycle? (Overall)	Does your problem deal with a specific stage in the interaction cycle?

Is your problem about actually performing physical actions on interface objects? (Physical Actions)	Is your problem about cognition or the user’s ability to understand how to use the system?

Is your problem concerned with the user’s understanding after he made an action? (Assessment)	Is your problem concerned with the user’s understanding before he makes an action?

Is your problem about how well the system supports the user in planning use of the system to accomplish a task? (Planning)	Does your problem concern the user's ability to determine (know or not know) how to do a task step? (Translation)

Table 2: First version of the Wizard

Is your problem one that is internal to the system and invisible to users? For example, does the system automate too much and take control away from the user? (Outcome and System Functionality)	Does your problem concern the user's interaction with the user interface? For example, is your problem related to the user's ability to plan for his task, determine appropriate interface elements for that task, manipulate those interface elements, or make sense of the results his actions?

Is your problem independent of the interaction cycle? For example, does the problem deal with interaction flaws that occur throughout the system? (Overall)	Does your problem deal with a specific stage in the interaction cycle? For example, does your problem deal with an interaction flaw that occurs in one place?

Is your problem about actually performing physical actions on interface objects? For example, does the user have problems manipulating interface objects? (Physical Actions)	Is your problem about cognition or the user's ability to understand how to use the system? For example, does the user have trouble determining what interface objects mean?

Is your problem concerned with the user's understanding after he made an action? For example, does the user have trouble understanding feedback from the system? (Assessment)	Is your problem concerned with the user's understanding before he makes an action? For example, does the user have trouble determining how to perform a task?

Is your problem about how well the system supports the user in planning use of the system to accomplish a task? For example, can the user determine what they can do with the system? (Planning)	Does your problem concern the user's ability to determine (know or not know) how to do a task step? For example, does the user know what physical actions to make on which user interface objects? (Translation)

Table 3: Second version of the Wizard

The analyst subjects in the intermediate and expert levels correctly diagnosed all problems and then confirmed their diagnoses with the Wizard. After the first few diagnoses with the Wizard, they begin to focus only on key words in the abstractions and used the Wizard much more rapidly. The result indicates that the Wizard was helping them develop specific words for describing intention. Such words would work well with LSA and provide an opportunity to use the Wizard as training wheels for writing problem descriptions.

On several occasions, the analyst subjects who were new and beginning users chose incorrectly but were then guided to the correct node by the Wizard. For new users of the UAF, the Wizard offers the most potential as a teaching tool. As the users learn more about the Interaction Cycle, they will benefit more from the Wizard as a tool for writing problem descriptions.

The intermediate and expert levels preferred the Wizard with only abstract descriptions because they were already familiar with the categories. In contrast, users new to the UAF preferred the Wizard with the specific examples because they were not yet familiar with the types of problems in each stage of the interaction cycle.

The feedback provided by the analyst subjects led us to develop a third version of the Wizard (Table 4). The most important change from the previous two versions of the Wizard is the wording of the questions and examples. Using verbal protocol, we found that certain words and phrases confused the analyst subjects and lead to misdiagnoses. For example, the question for the Outcome and System Functionality stage in the first and second version read: “Is your problem one that is internal to the system and invisible to the user?” The analyst subjects, particularly those with limited experience with the UAF, did not understand the phrase “internal to the system.” Several times, in fact, these analyst subjects selected this choice when the problem was not related to functional issues because they thought that internal meant any processing by the system. Because most interactions involve processing by the system, they made misdiagnoses. We corrected the problem in the third version by specifying that functional issues are not in the user interface software.

The third version also randomly changes the ordering of pairs, so that the choice that maps to a specific stage in the Interaction Cycle is not always on the left. We noticed that some of the more advanced analyst subjects had learned the pattern and were no longer reading the nodes. The Wizard is only valuable as a training tool if users become very familiar with the concepts that distinguish between top-level nodes.

In addition, the third Wizard summarizes a decision in terms of the Interaction Cycle after the user selects a choice. For example, if a user begins by selecting the choice that does not map to Outcome and System Functionality, she would be notified that the problem is related to the Independent, Physical Actions, Assessment, Planning, or Translation stages of the Interaction Cycle. We believe that the feedback will help users associate concepts from the choices with specific stages.

The results of the study also confirmed the usefulness of examples to accompany the abstract descriptions. The Wizard allows users to turn examples on and off, so that experienced users are not distracted by information that they may not need. Although examples are necessary, the study confirmed that overly specific examples made new and beginning users of the UAF select incorrect choices. For example, the choice in the second version of the Wizard that included the question, “Does your problem deal with consistency as it affects a user’s ability to plan to use the system for a task?” led two analyst subjects to select the other choice that mapped to Overall because their problem did not deal with consistency as it affects planning. We corrected this problem by using more generic examples.

As part of the study, we had the analyst subjects circle key words and phrases in the problem descriptions, which they felt best described the essence of the problem. We eventually plan to have users highlight key words in problem descriptions that they submit to the LSA-based tool. The highlighted words will be weighted more heavily to allow the tool to distinguish problem context from the problem itself. We believe that the weighting has the potential to greatly improve the accuracy of LSA diagnoses.

With this additional part of the study, we wanted to determine how similar the choice of highlighted words was among analyst subjects. Even though the analyst subjects were of all experience levels, they highlighted essentially the same words. We believe that the consistency of highlighting suggests that users are consistent in their understanding of what words constitute the essence of a problem and bodes well for reliable diagnosis through weighting.

Is your problem in the non-user interface software (e.g., a bug in the back end computation)? For example, does the system automate too much and take control away from the user? (Outcome and System Functionality)	Does your problem concern the user’s interaction with the user interface? For example, is your problem related to user planning, determining actions, making actions, or understanding feedback?

Does your problem cut across the whole User Interaction Cycle and not just a particular part? For example, does the problem deal with interaction flaws that occur in several places in the user interface? (Overall)	Does your problem deal with a specific stage in the User Interaction Cycle? For example, is your problem related to user planning, determining actions, making actions, or understanding feedback?

Is your problem about actually performing physical actions on interface objects or with devices? For example, does the user have problems finding or seeing an object to click or actually performing the clicking and dragging? (Physical Actions)	Is your problem about cognition (thinking, knowing) or the user’s ability to understand how to use the system? For example, is your problem related to user planning, determining actions, or understanding feedback?

Is your problem concerned with the user’s ability to understand the outcome of an action after he made the action? For example, does the user have trouble understanding feedback from the system? (Assessment)	Is your problem concerned with the user’s understanding of what action to take and/or how to do an action before he makes the action or the next appropriate action? For example, does the user have trouble determining how to perform a task or the next appropriate task?

Is your problem about how well the system supports the user in high-level planning use of the system to accomplish a task? For example, can the user make an overall general plan for using the system? (Planning)	Does your problem concern the user's ability to determine (know or not know) how to do a specific task step? For example, does the user know what physical action to make on which user interface object? (Translation)

Table 4: Third version of the Wizard

6.5 Training LSA

LSA has great potential for improving the content and structure of the UAF as well as automating the usability problem diagnosis process. LSA can simulate an understanding of concepts by providing measures of semantic similarity between texts. The key to enabling an understanding of concepts by LSA is training it on a sufficiently large number of texts that address the appropriate concepts. For our purposes, LSA must understand concepts related to human interaction with user interfaces. As such, we have chosen to train LSA on HCI literature.

We selected 537 articles from the following journals and conferences (a list of these articles is available in Appendix A):

- Behavior and Information Technology
- ACM Conference on Human Factors in Computing Systems
- Human-Computer Interaction
- International Journal of Human-Computer Interaction
- International Journal of Human-Computer Studies
- Interacting with Computers

The articles from journals and conferences provide good coverage of specific areas in the field of HCI, but we believe HCI textbooks would provide a better overall coverage of HCI concepts and theories. As such, we are currently negotiating with the publishers of the following HCI textbooks for digital copies of their books:

- Baecker, R. M. and W. A. S. Buxton, Eds. (1987). *Readings in Human-computer Interaction: A Multidisciplinary Approach*. Los Altos, CA, Morgan Kaufmann Publishers, Inc.
- Baecker, R. M., J. Grudin, W. A. S. Buxton and S. Greenberg, Eds. (1995). *Readings in Human-computer Interaction: Toward the Year 2000*. San Francisco, CA, Morgan Kaufman Publishers, Inc.
- Carroll, J. M., Ed. (1991). *Designing Interaction: Psychology at the Human-computer Interface*. Cambridge, England, Cambridge University Press.
- Dix, A., J. Finlay, G. Abowd and R. Beale (1998). *Human-computer Interaction*. London, Prentice Hall Europe.
- Helander, M., Ed. (1988). *Handbook of Human-computer Interaction*. Amsterdam, North-Holland.
- Mandel, T. (1997). *Elements of User Interface Design*. New York, John Wiley & Sons, Inc.
- National Research Council (1997). *More than Screen Deep*. Washington, D.C., National Academy Press.
- Nielsen, J. (1989). *Coordinating user Interfaces for Consistency*. Boston, Academic Press, Inc.
- Norman, D. A. and S. W. Draper, Eds. (1986). *User Centered System Design: New Perspectives on Human-computer Interaction*. Hillsdale, NJ, Lawrence Erlbaum Associates.

- Powell, J. E. (1990). *Designing User Interfaces*. San Marcos, CA, Microtrend Books.
- Preece, J., Ed. (1993). *A Guide to Usability: Human Factors in Computing*. Wokingham, England, Addison-Wesley.
- Preece, J., Y. Rogers, H. Sharp, D. Benyon, S. Holland and T. Carey (1994). *Human-Computer Interaction*. Wokingham, England, Addison-Wesley.
- Raskin, J. (2000). *The Humane Interface: New Directions for Designing Interactive Systems*. Boston, Addison-Wesley.
- Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Reading, MA, Addison-Wesley.
- Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Reading, MA, Addison-Wesley.
- Zetie, C. (1995). *Practical user Interface Design: Making Guis Work*. London, McGraw-Hill.

We are reasonably optimistic that we will be able to obtain digital copies of the books by the beginning of phase 2.

We are working with Knowledge Analysis Technologies, a Colorado-based company that specializes in LSA technology. They have performed the first build of the space based on the literature that we have chosen. They have provided us with an early prototype of a tool for exploring the space, and we have performed some early informal tests of the space.

7 Future Work

In phase 1, we developed an infrastructure for usability problem data analysis. In phase 2, we will extend the infrastructure with LSA technology. This extension process consists of the following: integrating the tools and incorporating LSA technology, associating content with UAF nodes, and formalizing and validating our work.

7.1 Integrating the Tools and Incorporating LSA Technology

The tools currently exist independently of one another. To better guide a practitioner through the usability process, we will update and integrate the tools. For example, we will update the PRT to accommodate changes in the data management cycle. More specifically, we will extend the tool to help practitioners collect critical incident data, including information on intention, and organize that data by characteristics such as project, version, session, and trial. The updated PRT will be integrated with the Viewer, so that practitioners can access problem reports based on nodes they encounter during diagnosis.

In addition to updating and integrating the tools, we will integrate LSA technology. For example, we plan to investigate the possibility of a Wizard driven dynamically by LSA that asks questions to elicit missing distinguisher information from the usability problem analyst. On the basis of analyzing the semantic content of the nodes involved in candidate diagnosis paths, LSA will suggest distinguisher-based questions to disambiguate a given diagnosis decision at a given UAF node. As practitioners gain experience using this tool, we expect they will learn how to anticipate the LSA-driven Wizard questions and will adapt by collecting more of the necessary information during data gathering. Thus, the LSA-driven Wizard will act as a kind of training wheels through which practitioners become more expert at both data gathering and analysis.

7.2 Node Content

The UAF is valuable as a conceptual framework, but it is even more valuable when it is used an organizing structure for detailed usability content. We are developing and collecting content to associate with nodes in the UAF, so practitioners will have direct access to information that helps them better understand and relate problems that they have diagnosed. Integrating the tools and incorporating LSA technology will allow us to provide practitioners with the following:

- Actual node content: The content includes usability principles, issues, concepts, and design rationale related to the node's usability concept.
- References to literature: LSA technology will enable us to search the literature that we used to build the LSA space and include references to specific paragraphs that address the usability concept represented by the node.
- Sample usability problem(s): We are currently using the UAF PRT to construct a usability problem library. The construction of the library involves extracting problems from critical incident reports from professional usability labs, diagnosing the problems according to the UAF, and reporting the results. We currently have over 100 analyzed usability problems in our library. We will continue to add to the library until the end of phase 2. The sample problems allow practitioners to confirm diagnoses by comparison. In addition, practitioners can see solutions for related problems.
- Actual project-related usability problem(s): In addition to sample usability problems, practitioners will want to see what previous problems they or members of their team or organization have diagnosed at a given node. Thus, each node will serve as an entry point into the PRT, which will contain records of the life histories of diagnosed usability problems. Each problem will contain the fields listed in Table 1.

7.3 Formalizing and Validating Our Work

A key component of phase 2 will be formalizing our research questions and performing summative studies to validate our work. Potential research activities for phase 2 that might be useful in the formalization and validation process include:

- Computing measures of semantic similarity or distance between nodes and paths in the UAF based on their positions in the LSA space
- Determining the most significant words in determining semantic similarity or distance in the UAF
- Determining if the UAF covers the semantic space defined by the literature used to train LSA
- Determining whether or not a problem statement has enough information to provide immediate intention

- Developing an LSA-based Wizard that can train analyst subjects to provide intention information in problem reports
- Automatically diagnosing a problem based on an appropriate problem statement

8 Summary and Conclusion

We have developed the basis for an infrastructure to support usability problem data analysis. In particular, we have adapted a conceptual framework for organizing usability problem data, created an appropriate process, prototyped tools that leverage the framework and guide practitioners through the process, and begun exploring LSA technology for automating parts of the process. Formative studies have helped us identify and understand the problem and provided a good indication that our approach is appropriate.

We chose the UAF as our conceptual framework and have adapted and embedded it in our infrastructure. Since the beginning of phase 1, we have released three versions of the UAF. For each version, we used the results of exploratory and formative studies to improve the content of nodes and refine the overall structure.

Our proposed process for usability data management improves upon the existing process by moving extraction and initial diagnosis to the usability data collection stage. Our process allows analysts to capture intention while the user participant is still available for questioning. Although our process requires more resources in the collection stage than the existing process, it is our working hypothesis that it will save resources later in the usability data management cycle and result in the creation of problem reports that are more useful for developers working on redesign and implementation.

We have also developed working prototypes of three tools: the UAF Viewer, the UAF Editor, and the UAF PRT. These tools give us the ability to update the UAF, view the contents of the UAF for diagnosis purposes, and store and organize large numbers of usability problem reports. We have also constructed a paper prototype of the Wizard, a tool that facilitates the identification of intention. We have performed several exploratory studies to improve the interaction design of these tools.

Finally, we have trained LSA on HCI literature. We have done exploratory evaluation of the resulting semantic space, and are ready to begin more formally evaluating the space and its potential in phase 2. Our initial work with LSA indicates that it has great potential for automating parts of the analysis process.

Our work contributes to the limited group of existing methods and techniques for usability problem data analysis. In particular, our work offers a practical, complete approach that guides practitioners through a repeatable process and helps them organize and reuse collected data.

References

Allen, B. G. and E. Buie (2002). "What's in a word? The semantics of usability." *interactions*, 9 (2), 17-21.

Andre, T. S., H. R. Hartson, S. M. Belz and F. A. McCreary (2001). "The User Action Framework: A Reliable Foundation for Usability Engineering Support Tools." *International Journal of Human-Computer Studies*, 54 (1), 107-136.

Andre, T. S., H. R. Hartson and R. C. Williges (2002). "Determining the effectiveness of the usability problem inspector: A theory-based model and tool for finding usability problems." *Human Factors*, 45 (3), 455-482.

Aucella, A. F. (1997). "Ensuring success with usability engineering." *interactions*, 4 (3), 19-22.

Brooks, P. (1994). "Adding value to usability testing." *Usability Inspection Methods*. J. Nielsen and R. L. Mack. New York, John Wiley & Sons. 255-271.

Brooks, R. (1990). "The Contribution of Practitioner Case Studies to Human-Computer Interaction Science." *Interacting with Computers*, 2 (1), 3-7.

Butler, K. A. (1996). "Usability engineering turns 10." *interactions*, 3 (1), 58-75.

Capra, M. G., H. R. Hartson and R. C. Williges (2002). *End-user Critical Incident Classification with the User Action Framework*. Master's thesis, Blacksburg, VA, Virginia Tech.

Castillo, J. C., J. R. Hartson and D. Hix (1998). "Remote usability evaluation: can users report their own critical incidents?" *ACM Computer-Human Interaction*, 253-254.

Cockton, G. and D. Lavery (1999). "A framework for usability problem extraction." *Proceedings of the IFIP Seventh International Conference on Human-Computer Interaction - INTERACT '99*, 344-352.

Colaso, V., R. H. Hartson, M. Perez-Quinonez and S. McCrickard (2003). *A Usability Problem Inspection Tool: Development and Formative Evaluation*. Master's thesis, Blacksburg, VA, Virginia Tech.

Deerwester, S. C., S. T. Dumais, T. K. Landauer, G. W. Furnas and R. A. Harshman (1990). "Indexing by latent semantic analysis." *Journal of the American Society of Information Science*, 41 (6), 391-407.

Desurvire, H. W. (1994). "Faster, Cheaper! Are usability inspection methods as effective as empirical testing?" *Usability Inspection Methods*. J. Nielsen and R. L. Mack. New York, John Wiley & Sons. 173-202.

Dumais, S. T., G. W. Furnas, T. K. Landauer, S. Deerwester and R. Harshman (1988). "Using Latent Semantic Analysis to Improve Access to Textual Information." *ACM CHI'88 Conference on Human Factors in Computing Systems*, 281-285.

Dumais, S. T. and J. Nielsen (1992). "Automating the assignment of submitted manuscripts to reviewers." *Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 233-244.

Foltz, P. W., M. A. Britt and C. A. Perfetti (1994). "Where did you learn that? Matching student essays to the texts they have read." *Fourth Annual Conference of the Society for Text and Discourse*,

Foltz, P. W., M. A. Britt and C. A. Perfetti (1996). "Reasoning from multiple texts: An automatic analysis of readers' situation models." *18th Annual Cognitive Science Conference*, 110-115.

Foltz, P. W., W. Kintsch and T. K. Landauer (1998). "The measurement of textual coherence with latent semantic analysis." *Discourse Processes*, 25 (2), 285-307.

Gould, J. D. and C. Lewis (1985). "Designing for usability: key principles and what designers think." *Communications of the ACM*, 28 (3), 300-311.

Gray, W. D. and M. C. Salzman (1998). "Damaged Merchandise? A Review of Experiments That Compare Usability Evaluation Methods." *Human-Computer Interaction*, 13 (3), 203-261.

Griffin, T., S. Schwartz and K. Sofronoff (1998). "Implicit processes in medical diagnosis." *Implicit and Explicit Mental Processes*. K. Kirsner, C. Speelman, M. Maybery, A. O'Brien-Malone, M. Anderson and C. MacLeod. Mahwah, New Jersey, Lawrence Erlbaum Associates. 329-342.

Harmon, D. (1986). "An experimental study of factors important in document ranking." *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 186-193.

Hartson, H. R. (2003). "Cognitive, physical, sensory, and functional affordances in interaction design." *Behavior and Information Technology*, 22 (5), 315-338.

Hartson, H. R., T. S. Andre, R. C. Willeges and L. v. Rens (1999). "The User Action Framework: A theory-based foundation for inspection and classification of usability problems." *Human-Computer Interaction*, 1058-1062.

Hartson, H. R., T. S. Andre and R. C. Williges (2003). "Criteria for evaluating usability evaluation methods." *International Journal of Human-Computer Interaction*, 15 (1), 145-181.

Hilbert, D. M. and D. F. Redmiles (2000). "Extracting usability information from user interface events." *ACM Computing Survey*, 32 (4), 384-421.

Hix, D. and H. R. Hartson (1993). *Developing User Interfaces: Ensuring Usability Through Product & Process*. New York, John Wiley & Sons, Inc.

Jeffries, R. (1994). "Usability problem reports: Helping evaluators communicate effectively with developers." *Usability Inspection Methods*. J. Nielsen and R. L. Mack. New York, John Wiley and Sons. 273-294.

Kaur, K., N. Maiden and A. Sutcliffe (1999). "Interacting with Virtual Environments: An Evaluation of a Model of Interaction." *Interacting with Computers*, 11 (4), 403-426.

Keenan, S. L. (1996). *Product usability and process improvement based on usability problem classification*. Ph.D. dissertation, Blacksburg, VA, Virginia Tech.

Keenan, S. L., H. R. Hartson, D. G. Kafura and R. S. Schulman (1999). "The Usability problem taxonomy: A framework for classification and analysis." *Empirical Software Engineering*, 4 (1), 71-104.

Landauer, T. K., P. W. Foltz and D. Laham (1998). "Introduction to latent semantic analysis." *Discourse Processes*, 25, 259-284.

Lavery, D. and G. Cockton (1997). "Representing predicted and actual usability problems." *Proceedings of the International Workshop on Representations in Interactive Software Development*, 97-108.

Lavery, D., G. Cockton and M. P. Atkinson (1997). "Comparison of evaluation methods using structured usability problem reports." *Behaviour and Information Technology*, 16 (4-5), 246-266.

Lim, K. H., I. Benbasat and P. Todd (1996). "An experimental investigation of the interactive effects of interface style, instructions, and task familiarity on user performance." *ACM Transactions of Computer-Human Interaction*, 3, 1-37.

Mack, R. and F. Montaniz (1994). "Observing, predicting, and analyzing usability problems." *Usability Inspection Methods*. J. Nielsen and R. L. Mack. New York, John Wiley & Sons. 295-339.

Mayhew, D. J. (1992). *Principles and Guidelines in Software User Interface Design*. Englewood Cliffs, NJ, Prentice-Hall.

Molich, R., N. Bevan, I. Curson, S. Butler, E. Kindlund, D. Miller and J. Kirakowski (1998). "Comparative evaluation of usability tests." *Usability Professional's Association*, 83-84.

Muller, M. J., T. Dayton and R. Root (1993). "Comparing Studies that Compare Usability Assessment Methods: An Unsuccessful Search for Stable Criteria." *ACM INTERCHI'93 Conference on Human Factors in Computing Systems -- Adjunct Proceedings*, 185-186.

Nayak, N. P., D. Mrazek and D. R. Smith (1995). "Analyzing and communicating usability data: Now that you have the data what do you do?" *ACM SIGCHI Bulletin*, 22-30.

Nielsen, J. (1993). *Usability Engineering*. Boston, Academic Press.

Nielsen, J. and R. Molich (1990). "Heuristic Evaluation of User Interfaces." *ACM CHI'90 Conference on Human Factors in Computing Systems*, 249-256.

Norman, D. A. (1983). "Design principles for human-computer interfaces." *SIGCHI conference on Human Factors in Computing Systems*, 1-10.

Norman, D. A. and S. W. Draper, Ed.^Eds. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, Erlbaum.

Ostrand, T. J. and E. J. Weyuker (1984). "Collecting and categorizing software error data in an industrial environment." *The Journal of Systems and Software*, 4 (4), 289-300.

Pernice, K. and M. B. Butler (1995). "Database support for usability testing." *interactions*, 2 (1), 27-31.

Polson, P. G., C. Lewis, J. Rieman and C. Wharton (1992). "Cognitive walkthroughs: a method for theory-based evaluation of user interfaces." *International Journal of Man-Machine Studies*, 36 (5), 741-773.

Rizzo, A., E. Marchigiani and A. Andreadis (1997). "The AVANTI project: prototyping and evaluation with a cognitive walkthrough based on Norman's model of action." *Designing Interactive Systems Conference*, 305-309.

Rubin, J. (1994). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. New York, Wiley.

Scapin, D. L. (1990). "Organizing human factors knowledge for the evaluation and design of interfaces." *International Journal of Human-Computer Interaction*, 2 (3), 203-229.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Reading, MA, Addison-Wesley.

Soto, R. (1999). "Learning and Performing by Exploration: Label Quality Measured by Latent Semantic Analysis." *ACM CHI 99 Conference on Human Factors in Computing Systems*, 418-425.

Springett, M. (1998). "Linking surface error characteristics to root problems in user-based evaluation studies." *Working Conference on Advanced Visual Interfaces - AVI '98*, 102-113.

Vora, P. R. (1995). "Classifying user errors in human-computer interactive tasks." *Common Ground: The Newsletter of Usability Professionals*, 5 (2), 15.

Wharton, C., J. Bradford, R. Jeffries and M. Franzke (1992). "Applying cognitive walkthroughs to more complex user interfaces: experiences, issues, and recommendations." *Conference on Human Factors and Computing Systems*, 381-388.

Wilson, C. E. and K. P. Coyne (2001). "Tracking usability issues: To bug or not to bug?" *interactions*, 8 (3), 15-19.

Appendix

A Articles Used to Build the LSA Space

The following is a list of HCI articles that were included in the first build of the LSA space:

Aalst, J. W. v., T. T. Carey and D. L. McKerlie (1995). "Design Space Analysis as "Training Wheels" in a Framework for Learning User Interface Design." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1, 154-161.

Aaltonen, A., A. Hyrskykari and K.-J. Raiha (1998). "101 Spots, or How do Users Read Menus?" *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1, 132-139.

Accot, J. and S. Zhai (1997). "Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1, 295-302.

Accot, J. and S. Zhai (2003). "Refining Fitts' law models for bivariate pointing." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 193-200.

Adelson, B. (1992). "Evocative Agents and Multi-Media Interface Design." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 351-356.

Agarwal, R., J. Prasad and M. C. Zanino (1996). "Training Experiences and Usage Intentions: A Field Study of a Graphical User Interface." *International Journal of Human-Computer Studies*, 45 (2), 215-241.

Ahlberg, C. and B. Shneiderman (1994). "The Alphaslider: A Compact and Rapid Selector." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1, 365-371.

Ahlberg, C., C. Williamson and B. Shneiderman (1992). "Dynamic Queries for Information Exploration: An Implementation and Evaluation." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 619-626.

Akamatsu, M. and S. Sato (1994). "A Multi-Modal Mouse with Tactile and Force Feedback." *International Journal of Human-Computer Studies*, 40 (3), 443-453.

Al-Gahtani, S. S. and M. King (1999). "Attitudes, Satisfaction and Usage: Factors Contributing to Each in the Acceptance of Information Technology." *Behaviour and Information Technology*, 18 (4), 277-297.

Albinsson, P.-A. and S. Zhai (2003). "High precision touch screen interaction." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 105-112.

Allen, B. G. and E. Buie (2002). "What's in a word? The semantics of usability." *Interactions*, 9 (2), 17-21.

Allwood, C. M. and L. Hedelin (1996). "Information Administrative Support of Decision Processes in Organizations." *Behaviour and Information Technology*, 15 (6), 352-362.

Altmann, E. M., J. H. Larkin and B. E. John (1995). "Display Navigation by an Expert Programmer: A Preliminary Model of Memory." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 3-10.

Amant, R. S. (1999). "User Interface Affordances in a Planning Representation." *Human-Computer Interaction*, 14 (3), 317-354.

Amento, B., W. Hill, L. Terveen, P. Ju and D. Hix (1999). "An Empirical Evaluation of User Interfaces for Topic Management of Web Sites." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 552-559.

Andre, T. S., H. R. Hartson, S. M. Belz and F. A. McCreary (2001). "The User Action Framework: A Reliable Foundation for Usability Engineering Support Tools." *International Journal of Human-Computer Studies*, 54 (1), 107-136.

Arango, G. (1989). "Domain analysis: from art form to engineering discipline." *Proceedings of International Workshop on Software Specifications & Design*, 152-159.

Archer, J. E., R. Conway and F. B. Schneider (1984). "User Recovery and Reversal in Interactive Systems." *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 6 (1), 1-19.

Ashlund, S. and K. J. Horwitz (1996). "Usability Improvements in Lotus cc:Mail for Windows." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 481-488.

Aucella, A. F. (1997). "Ensuring success with usability engineering." *ACM Interactions*, 4 (3), 19-22.

Baar, D. J. M. J. d., J. D. Foley and K. E. Mullet (1992). "Coupling Application Design and User Interface Design." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 259-266.

Bailey, G. S. (1993). "Iterative Methodology and Designer Training in Human-Computer Interface Design." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 198-205.

Bailey, W. A., S. T. Knox and E. F. Lynch (1988). "Effects of Interface Design Upon User Productivity." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 207-212.

Baldonado, M. Q. W. and T. Winograd (1997). "SenseMaker: An Information-Exploration Interface Supporting the Contextual Evolution of a User's Interests."

Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems, 1**,** 11-18.

Ballas, J. A., C. L. Heitmeyer and M. A. Perez (1992). "Evaluating Two Aspects of Direct Manipulation in Advanced Cockpits." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 127-134.

Bannon, L., A. Cypher, S. Greenspan and M. L. Monty (1983). "Evaluation and Analysis of Users' Activity Organization." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 54-57.

Barfield, L., E. Boeve and S. Pemberton (1991). "The Views User-Interface System." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 415-416.

Barnard, P., N. Hammond, A. MacLean and J. Morton (1982). "Learning and Remembering Interactive Commands." *Proceedings of Human Factors in Computer Systems*, 2-7.

Barnard, P., M. Wilson and A. MacLean (1987). "Approximate Modelling of Cognitive Activity: Towards an Expert System Design Aid." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 21-26.

Basden, A. and P. R. Hibberd (1996). "User Interface Issues Raised by Knowledge Refinement." *International Journal of Human-Computer Studies*, 45 (2), 135-155.

Bass, L. and B. E. John (2001). "Supporting usability through software architecture." *IEEE Computer*, 113-115.

Bastien, J. M. and D. L. Scapin (1995). "Evaluating a user interface with ergonomic criteria." *International Journal of Human-Computer Interaction*, 7 (2), 105-121.

Bauersfeld, P. F. and J. L. Slater (1991). "User-Oriented Color Interface Design: Direct Manipulation of Color in Context." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 417-418.

Bayles, M. E. (2002). "Designing online banner advertisements: should we animate?" *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 363-366.

Beier, B. and M. W. Vaughan (2003). "The bull's-eye: a framework for web application user interface design guidelines." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 489-496.

Bell, B., J. Rieman and C. Lewis (1991). "Usability Testing of a Graphical Programming System: Things We Missed in a Programming Walkthrough." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 7-12.

Bellotti, V. (1993). "Integrating Theoreticians' and Practitioners' Perspectives with Design Rationale." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 101-106.

Bellotti, V., M. Back, W. K. Edwards, R. E. Grinter, A. Henderson and C. Lopes (2002). "Making sense of sensing systems: five questions for designers and researchers." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 415-422.

Bellotti, V. and K. Edwards (2001). "Intelligibility and Accountability: Human Considerations in Context Aware Systems." *Human-Computer Interaction*, 16 (2/4), 193-212.

Bellotti, V., S. B. Shum, A. MacLean and N. Hammond (1995). "Multidisciplinary Modeling in HCI Design ...In Theory and in Practice." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 146-153.

Berlin, L. M., R. Jeffries, V. L. O'Day, A. Paepcke and C. Wharton (1993). "WHERE Did You Put It? Issues in the Design and Use of a Group Memory." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 23-30.

Bhavnani, S. K., B. K. Christopher, T. M. Johnson, R. J. Little, F. A. Peck, J. L. Schwartz and V. J. Strecher (2003). "Strategy hubs: next-generation domain portals with search procedures." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 393-400.

Bhavnani, S. K. and B. E. John (1997). "From Sufficient to Efficient Usage: An Analysis of Strategic Knowledge." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 91-98.

Bhavnani, S. K., F. Reif and B. E. John (2001). "Beyond Command Knowledge: Identifying and Teaching Strategic Knowledge for Using Complex Computer Applications." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 229-236.

Bier, E. A. and K. Pier (1991). "Documents as User Interfaces." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 443-444.

Birdi, K. S. and D. Zapf (1997). "Age Differences in Reactions to Errors in Computer-Based Work." *Behaviour and Information Technology*, 16 (6), 309-319.

Black, J. B., J. S. Bechtold, M. Mitrani and J. M. Carroll (1989). "On-Line Tutorials: What Kind of Inference Leads to the Most Effective Learning?" *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 81-83.

Blackmon, M. H., M. Kitajima and P. G. Polson (2003). "Repairing usability problems identified by the cognitive walkthrough for the web." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 497-504.

Blandford, A. E., M. D. Harrison and P. J. Barnard (1995). "Using Interaction Framework to Guide the Design of Interactive Systems." *International Journal of Human-Computer Studies*, 43 (1), 101-130.

Blandford, A. E., S. J. B. Shum and R. M. Young (1998). "Training Software Engineers in a Novel Usability Evaluation Technique." *International Journal of Human-Computer Studies*, 49 (3), 245-279.

Blomberg, J. L. and A. Henderson (1990). "Reflections on Participatory Design: Lessons from the Trillium Experience." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 353-359.

Bodker, S. (1998). "Understanding Representation in Design." *Human-Computer Interaction*, 13 (2), 107-125.

Bohringer, K.-F. and F. N. Paulisch (1990). "Using Constraints to Achieve Stability in Automatic Graph Layout Algorithms." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 43-51.

Borgman, C. L., N. J. Belkin, W. B. Croft, M. E. Lesk and T. K. Landauer (1988). "Retrieval Systems for the Information Seeker: Can the Role of the Intermediary be Automated?" *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 51-53.

Borning, A. and M. Travers (1991). "Two Approaches to Casual Interaction Over Computer and Video Networks." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 13-19.

Boy, G. A. (1998). "Cognitive Function Analysis for Human-Centered Automation of Safety-Critical Systems." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 265-272.

Boyarski, D., C. Neuwirth, J. Forlizzi and S. H. Regli (1998). "A Study of Fonts Designed for Screen Display." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 87-94.

Branting, L. K. and P. S. Broos (1997). "Automated Acquisition of User Preferences." *International Journal of Human-Computer Studies*, 46 (1), 55-77.

Brennan, P., G. Deffner, D. Lawrence, M. Marics, E. Schwab and M. Franzke (1991). "Should We or Shouldn't We Use Spoken Commands in Voice Interfaces?" *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 369-372.

Brooks, R. (1990). "The Contribution of Practitioner Case Studies to Human-Computer Interaction Science." *Interacting with Computers*, 2 (1), 3-7.

Brown, J., T. C. N. Graham and T. Wright (1998). "The Vista Environment for the Coevolutinary Design of User Interfaces." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 376-383.

Bury, K. F., J. M. Boyle, R. J. Evey and A. S. Neal (1982). "Windowing vs Scrolling on a Visual Display Terminal." *Proceedings of Human Factors in Computer Systems*, 41-44.

Butler, K. A. (1985). "Connecting Theory and Practice: A Case Study of Achieving Usability Goals." *Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems*, 85-88.

Butler, K. A. (1996). "Usability engineering turns 10." *Interactions*, 3 (1), 58-75.

Byrne, M. D. (1993). "Using Icons to Find Documents: Simplicity is Critical." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 446-453.

Byrne, M. D., J. R. Anderson, S. Douglass and M. Matessa (1999). "Eye Tracking the Visual Search of Click-Down Menus." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 402-409.

Byrne, M. D., B. E. John, N. S. Wehrle and D. C. Crow (1999). "The Tangled Web We Wove: A Taskonomy of W\NW Use." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 544-551.

Byrne, M. D., S. D. Wood, P. N. Sukaviriya, J. D. Foley and D. Kieras (1994). "Automating Interface Evaluation." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 232-237.

Callahan, J., D. Hopkins, M. Weiser and B. Shneiderman (1988). "An Empirical Comparison of Pie vs. Linear Menus." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 95-100.

Campbell, C. and P. Maglio (1999). "Facilitating Navigation in Information Spaces: Road-Signs on the World Wide Web." *International Journal of Human-Computer Studies*, 50 (4), 309-327.

Card, S. K. and J. Austin Henderson (1987). "A Multiple, Virtual-Workspace Interface to Support User Task Switching." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 53-59.

Card, S. K., J. D. Mackinlay and G. G. Robertson (1990). "The Design Space of Input Devices." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 117-124.

Card, S. K., P. Pirolli and J. D. Mackinlay (1994). "The Cost-of-Knowledge Characteristic Function: Display Evaluation for Direct-Walk Dynamic Information Visualizations." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 238-244.

Card, S. K., G. G. Robertson and J. D. Mackinlay (1991). "The Information Visualizer, An Information Workspace." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 181-188.

Card, S. K., G. G. Robertson and W. York (1996). "The WebBook and the Web Forager: An Information Workspace for the World-Wide Web." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 111-117.

Carey, J. M., P. J. Mizzi and L. C. Lindstrom (1996). "Pull-Down versus Traditional Menu Types: An Empirical Comparison." *Behaviour and Information Technology*, 15 (2), 84-95.

Carey, T., S. Mitchell, D. Peerenboom and M. Lytwyn (1998). "Design Evolution in a Multimedia Tutorial on User-Centered Design." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 109-116.

Carr, D. A. (1994). "Specification of Interface Interaction Objects." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 372-378.

Carroll, J. M. (1990). "Infinite Detail and Emulation in an Ontologically Minimized HCI." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 321-327.

Carroll, J. M. (1997). "Human-Computer Interaction: Psychology as a Science of Design." *International Journal of Human-Computer Studies*, 46 (4), 501-522.

Carroll, J. M. and D. S. Kay (1985). "Prompting, Feedback and Error Correction in the Design of a Scenario Machine." *Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems*, 149-153.

Carroll, J. M. and W. A. Kellogg (1989). "Artifact as Theory-Nexus: Hermeneutics Meet Theory-Based Design." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 7-14.

Casner, S. and C. Lewis (1987). "Learning about Hidden Events in System Interactions." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 197-203.

Castillo, J. C., H. R. Hartson and D. Hix (1998). "Remote usability evaluation: can users report their own critical incidents?" *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 253-254.

Chalfonte, B. L., R. S. Fish and R. E. Kraut (1991). "Expressive Richness: A Comparison of Speech and Text as Media for Revision." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 21-26.

Chandra, S. and D. I. Blockley (1995). "Cognitive and Computer Models of Physical Systems." *International Journal of Human-Computer Studies*, 43 (4), 539-559.

Chi, E. H., P. Pirolli, K. Chen and J. Pitkow (2001). "Using Information Scent to Model User Information Needs and Actions and the Web." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 490-497.

Chi, E. H., J. Pitkow, J. Mackinlay, P. Pirolli, R. Gossweiler and S. K. Card (1998). "Visualizing the Evolution of Web Ecologies." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 400-407.

Chi, E. H., A. Rosien, G. Supattanasiri, A. Williams, C. Royer, C. Chow, E. Robles, B. Dalal, J. Chen and S. Cousins (2003). "The bloodhound project: automating discovery of web usability issues using the InfoScentp simulator." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 505-512.

Chimera, R. (1992). "Value Bars: An Information Visualization and Navigation Tool for Multi-Attribute Listings." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 293-294.

Chin, J. P., V. A. Diehl and K. L. Norman (1988). "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 213-218.

Cockburn, A. and B. McKenzie (2001). "What do Web Users Do? An Empirical Analysis of Web Use." *International Journal of Human-Computer Studies*, 54 (6), 903-922.

Cockburn, A. and B. McKenzie (2002). "Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 203-210.

Comstock, E. M. and W. M. Duane (1996). "Embed User Values in System Architecture: The Declaration of System Usability." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 420-427.

Corbett, A. T. and J. R. Anderson (2001). "Locus of Feedback Control in Computer-Based Tutoring: Impact on Learning Rate, Achievement and Attitudes." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 245-252.

Costagliola, G., S. Orefice, G. Polese, M. Tucci and G. Tortora (1999). "On the Generation of Interactive Iconic Environments." *International Journal of Human-Computer Studies*, 50 (5), 363-389.

Coutaz, J. and S. Balbo (1991). "Applications: A Dimension Space for User Interface Management Systems." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 27-32.

Coventry, L., A. D. Angeli and G. Johnson (2003). "Usability and biometric verification at the ATM interface." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 153-160.

Curtis, P., T. Heiserman, D. Jobusch, M. Notess and J. Webb (1999). "Customer-Focused Design Data in a Large, Multi-Site Organization." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 608-615.

Cypher, A. and D. C. Smith (1995). "KidSim: End User Programming of Simulations." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 27-34.

Czaja, S. J., J. Sharit, S. Nair and M. Rubert (1998). "Understanding Sources of User Variability in Computer-Based Data Entry Performance." *Behaviour and Information Technology*, 17 (5), 282-293.

Dalal, N. P. and G. M. Kasper (1994). "The Design of Joint Cognitive Systems: The Effect of Cognitive Coupling on Performance." *International Journal of Human-Computer Studies*, 40 (4), 677-702.

Danielson, D. R. (2002). "Web navigation and the behavioral effects of constantly visible site maps." *Interacting with Computers*, 14 (5), 601-618.

Darken, R. P. and J. L. Sibert (1996). "Wayfinding Strategies and Behaviors in Large Virtual Worlds." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 142-149.

Dervin, B. (1995). "The relationship of user-centered evaluation to design: addressing issues of productivity and power." *ACM SIGOIS Bulletin*, 16 (2), 42-46.

Diaper, D. (1989). "The Discipline of HCI." *Interacting with Computers*, 1 (1), 3-5.

Diaper, D. and P. Waelend (2000). "World Wide Web Working whilst Ignoring Graphics: Good News for Web Page Designers." *Interacting with Computers*, 13 (2), 163-181.

Dillon, A. and C. Watson (1996). "User Analysis in HCI -- The Historical Lessons from Individual Differences Research." *International Journal of Human-Computer Studies*, 45 (6), 619-637.

Dillon, R. F., J. D. Eday and J. W. Tombaugh (1990). "Measuring the True Cost of Command Selection: Techniques and Results." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 19-25.

Douglas, S. and T. Kirkpatrick (1996). "Do Color Models Really Make a Difference?" *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 399-405.

Dumais, S., E. Cutrell and H. Chen (2001). "Optimizing Search by Showing Results in Context." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 277-284.

Edwards, A. D. N. (1988). "The Design of Auditory Interfaces for Visually Disabled Users." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 83-88.

Edwards, W. K., V. Bellotti, A. K. Dey and M. W. Newman (2003). "The challenges of user-centered design and evaluation for infrastructure." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 297-304.

Egan, D. E., J. R. Remde, T. K. Landauer, C. C. Lochbaum and L. M. Gomez (1989). "Behavioral Evaluation and Analysis of a Hypertext Browser." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 205-210.

Ehret, B. D. (2002). "Learning where to look: location learning in graphical user interfaces." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 211-218.

Elrod, S., R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, E. Pedersen, K. Pier, J. Tang and B. Welch (1992). "Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations and Remote Collaboration." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 599-607.

Engel, F. L., P. Goossens and R. Haakma (1994). "Improved Efficiency through I- and E-Feedback: A Trackball with Contextual Force Feedback." *International Journal of Human-Computer Studies*, 41 (6), 949-974.

Erickson, T. and G. Salomon (1991). "Designing a Desktop Information System: Observations and Issues." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 49-54.

Fahlen, L. E., O. Stahl, C. G. Brown and C. Carlsson (1993). "A Space Based Model for User Interaction in Shared Synthetic Environments." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 43-48.

Fahlen, L. E., O. Stahl, C. G. Brown and C. Carlsson (1993). "A Space Based Model for User Interaction in Shared Synthetic Environments." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 43-48.

Fallman, D. (2003). "Design-oriented human: computer interaction." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 225-232.

Feldman, M. B. and G. T. Rogers (1982). "Toward the Design and Development of Style-Independent Interactive Systems." *Proceedings of Human Factors in Computer Systems*, 111-116.

Fischer, G. and A. Girgensohn (1990). "End-User Modifiability in Design Environments." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 183-191.

Fischer, G., S. Henninger and D. Redmiles (1991). "Intertwining Query Construction and Relevance Evaluation." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 55-62.

Fischer, G., A. Lemke and T. Schwab (1985). "Knowledge-Based Help Systems." *Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems*, 161-167.

Fischer, G., A. C. Lemke, T. Mastaglio and A. I. Morch (1990). "Using Critics to Empower Users." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 337-347.

Fischer, G., R. McCall and A. Morch (1989). "Design Environments for Constructive and Argumentative Design." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 269-275.

Fischer, G., K. Nakakoji, J. Ostwald, G. Stahl and T. Sumner (1993). "Embedding Computer-Based Critics in the Contexts of Design." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 157-164.

Fischer, G., S. A. Weyer, W. P. Jones, A. C. Kay, W. Kintsch and R. H. Trigg (1988). "A Critical Assessment of Hypertext Systems." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 223-227.

Foley, J., C. Gibbs, W. C. Kim and S. Kovacevic (1988). "A Knowledge-Based User Interface Management System." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 67-72.

Foltz, P. W., S. E. Davies, P. G. Polson and D. E. Kieras (1988). "Transfer Between Menu Systems." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 107-112.

Frank M. Shipman, I., C. C. Marshall and T. P. Moran (1995). "Finding and Using Implicit Structure in Human-Organized Spatial Layouts of Information." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 346-353.

Franzke, M. (1995). "Turning Research into Practice: Characteristics of Display-Based Interaction." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 421-428.

Frederick P. Brooks, J. (1988). "Grasping Reality Through Illusion -- Interactive Graphics Serving Science." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 1-11.

Frohlich, D. M. and P. Luff (1989). "Conversational Resources for Situated Action." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 253-258.

Furnas, G. W. (1991). "New Graphical Reasoning Models for Understanding Graphical Interfaces." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 71-78.

Furnas, G. W. (1997). "Effective View Navigation." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 367-374.

Gale, S. (1996). "A Collaborative Approach to Developing Style Guides." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 362-367.

Gantt, M. and B. A. Nardi (1992). "Gardeners and Gurus: Patterns of Cooperation among CAD Users." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 107-117.

Gaver, W., A. Sellen, C. Heath and P. Luff (1993). "One is Not Enough: Multiple Views in a Media Space." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 335-341.

Gaver, W. W. (1991). "Technology Affordances." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 79-84.

Gaver, W. W., J. Beaver and S. Benford (2003). "Ambiguity as a resource for design." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 233-240.

Gaylin, K. B. (1986). "How are Windows Used? Some Notes on Creating an Empirically-Based Windowing Benchmark Task." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 96-100.

Geiselman, R. E. and M. G. Samet (1982). "Notetaking and Comprehension for Computer-Displayed Messages: Personalized versus Fixed Formats." *Proceedings of Human Factors in Computer Systems*, 45-50.

Geist, R., R. Allen and R. Nowaczyk (1987). "Towards a Model of User Perception of Computer System Response Time." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 249-253.

Gentner, D. R. and J. Grudin (1990). "Why Good Engineers (Sometimes) Create Bad Interfaces." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 277-282.

George Chin, J. and M. B. Rosson (1998). "Progressive Design: Staged Evolution of Scenarios in the Design of a Collaborative Science Learning Environment." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 611-618.

Gieskens, D. F. and J. D. Foley (1992). "Controlling User Interface Objects Through Pre- and Postconditions." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 189-194.

Gillan, D. J. and S. D. Breedin (1990). "Designers' Models of the Human-Computer Interface." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 391-398.

Gillan, D. J., K. Holden, S. Adam, M. Rudisill and L. Magee (1990). "How Does Fitts' Law Fit Pointing and Dragging?" *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 227-234.

Gong, R. and J. Elkerton (1990). "Designing Minimal Documentation Using a GOMS Model: A Usability Evaluation of an Engineering Approach." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 99-106.

Good, M., R. Campbell, G. Lynch and P. Wright (1989). "Experience with Contextual Field Research." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 21-24.

Goodwin, N. (1987). "Functionality and usability." *Communications of the ACM*, 30 (3), 229-233.

Gould, J. D., L. Alfaro, R. Finn, B. Haupt, A. Minuto and J. Salaun (1987). "Why Reading Was Slower from CRT Displays Than from Paper." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 7-11.

Gould, J. D. and C. Lewis (1985). "Designing for Usability -- Key Principles and What Designers Think." *Communications of the ACM*, 28 (3), 300-311.

Graham, J. (1999). "The Reader's Helper: A Personalized Document Reading Environment." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 481-488.

Graham, T. C. N., L. A. Watts, G. Calvary, J. Coutaz, E. Dubois and L. Nigay (2000). "A dimension space for the design of interactive systems within their physical environments." *Proceedings of the Conference on Designing Interactive Systems*, 406-416.

Greenberg, S. and M. Rounding (2001). "The Notification Collage: Posting Information to Public and Personal Displays." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 514-521.

Greenberg, S. and I. H. Witten (1988). "How Users Repeat Their Actions on Computers: Principles for Design of History Mechanisms." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 171-178.

Grisedale, S., M. Graves and A. Grunsteidl (1997). "Designing a Graphical User Interface for Healthcare Workers in Rural India." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 471-478.

Grudin, J. (1989). "The case against user interface consistency." *Communications of the ACM*, 32 (10), 1164-1173.

Grudin, J. (1990). "Interface." *Proceedings of the 1990 ACM Conference on Computer-supported Cooperative Work*, 269-278.

Grudin, J. (1993). "Interface: an evolving concept." *Communications of the ACM*, 36 (4), 110-119.

Grudin, J., S. F. Ehrlich and R. Shriner (1987). "Positioning Human Factors in the User Interface Development Chain." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 125-131.

Guiard, Y., M. Beaudouin-Lafon and D. Mottet (1999). "Navigation as Multiscale Pointing: Extending Fitts' Model to Very High Precision Tasks." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 450-457.

Guindon, R. (1988). "How to Interface to Advisory Systems? Users Request Help with a Very Simple Language." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 191-196.

Gupta, S. M., L. H. Geyer and J. A. Maalouf (1983). "Effect of Font and Medium on Recognition/Confusion." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 144-149.

Gutierrez, O. (1989). "Prototyping Techniques for Different Problem Contexts." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 259-264.

Gutwin, C. and S. Greenberg (1998). "Effects of Awareness Support on Groupware Usability." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 511-518.

Halasz, F. G. and T. P. Moran (1983). "Mental Models and Problem Solving in Using a Calculator." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 212-216.

Hammond, N., P. Barnard, J. Coutaz, M. Harrison, A. MacLean and R. M. Young (1991). "Modelling User, System and Design: Results of a Scenarios Matrix Exercise." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 377-380.

Hammond, N., A. Jorgensen, A. MacLean, P. Barnard and J. Long (1983). "Design Practice and Interface Usability: Evidence from Interviews with Designers." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 40-44.

Hammontree, M. L., J. J. Hendrickson and B. W. Hensley (1992). "Integrated Data Capture and Analysis Tools for Research and Testing on Graphical User Interfaces." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 431-432.

Han, Y. and I. Zukerman (1997). "A Mechanism for Multimodal Presentation Planning Based on Agent Cooperation and Negotiation." *Human-Computer Interaction*, 12 (1/2), 187-226.

Hansen, A. L. (1997). "Reflections on I/Design: User Interface Design at a Startup." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 487-493.

Hargreaves, W., D. Rempel, N. M. Halpern, R. Markison, K. Kroemer and J. Litewka (1992). "Toward a More Humane Keyboard." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 365-368.

Harrison, B. L., K. P. Fishkin, A. Gujar, C. Mochon and R. Want (1998). "Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 17-24.

Harrison, B. L., H. Ishii, K. J. Vicente and W. A. S. Buxton (1995). "Transparent Layered User Interfaces: An Evaluation of a Display Design to Enhance Focused and Divided Attention." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 317-324.

Harrison, B. L. and K. J. Vicente (1996). "An Experimental Evaluation of Transparent Menu Usage." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 391-398.

Harrison, S. M. (1995). "A Comparison of Still, Animated, or Nonillustrated On-Line Help with Written or Spoken Instructions in a Graphical User Interface." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 82-89.

Hartson, H. R. (1998). "Human-Computer Interaction: Interdisciplinary Roots and Trends." *Journal of Systems and Software*, 43 (2), 103-118.

Hartson, H. R. (2003). "Cognitive, physical, sensory, and functional affordances in iteraction design." *Behavior and Information Technology*, 22 (5), 315-338.

Hartson, H. R., T. S. Andre, R. C. Willeges and L. van Rens (1999). "The User Action Framework: A theory-based foundation for inspection and classification of usability problems." *Proceedings of the International Conference on Human-Computer Interaction*, 1058-1062.

Hartson, H. R., J. C. Castillo, J. Kelso and W. C. Neale (1996). "Remote evaluation: the network as an extension of the usability laboratory." *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, 228-235.

Hartson, H. R. and D. Hix (1989). "Human-computer interface development: concepts and systems for its management." *ACM Computing Survey*, 21 (1), 5-92.

Hartswood, M. and R. Procter (2000). "Design Guidelines for Dealing with Breakdowns and Repairs in Collaborative Work Settings." *International Journal of Human-Computer Studies*, 53 (1), 91-120.

Hearst, M. A. (1995). "TileBars: Visualization of Term Distribution Information in Full Text Information Access." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 59-66.

Heer, J. and E. H. Chi (2002). "Separating the swarm: categorization methods for user sessions on the web." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 243-250.

Heinbokel, T., S. Sonnentag, M. Frese, W. Stolte and F. C. Brodbeck (1996). "Don't Underestimate the Problems of User Centredness in Software Development Projects -- There Are Many!" *Behaviour and Information Technology*, 15 (4), 226-236.

Hemenway, K. (1982). "Psychological Issues in the Use of Icons in Command Menus." *Proceedings of Human Factors in Computer Systems*, 20-23.

Hendrickson, J. J. (1989). "Performance, Preference, and Visual Scan Patterns on a Menu-Based System: Implications for Interface Design." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 217-222.

Herbsleb, J. D. and E. Kuwana (1993). "Preserving Knowledge in Design Projects: What Designers Need to Know." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 7-14.

Hershmann, D. (1995). "The Effects of Practical Business Constraints on User Interface Design." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 531-537.

Hewett, T. T. and C. T. Meadow (1986). "On Designing for Usability: An Application of Four Key Principles." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 247-252.

Hilbert, D. M. and D. F. Redmiles (2000). "Extracting usability information from user interface events." *ACM Computing Survey*, 32 (4), 384-421.

Hill, R. D. (1992). "The Abstraction-Link-View Paradigm: Using Constraints to Connect User Interfaces to Applications." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 335-342.

Hill, W. C. and J. D. Hollan (1992). "Pointing and Visualization." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 665-666.

Hill, W. C. and J. R. Miller (1988). "Justified Advice: A Semi-Naturalistic Study of Advisory Strategies." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 185-190.

Hinckley, K., E. Cutrell, S. Bathiche and T. Muss (2002). "Quantitative analysis of scrolling techniques." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 65-72.

Hochheiser, H. and B. Shneiderman (2000). "Performance Benefits of Simultaneous Over Sequential Menus as Task Complexity Increases." *International Journal of Human-Computer Interaction*, 12 (2), 173-192.

Hofmeester, G. H., J. A. M. Kemp and A. C. M. Blankendaal (1996). "Sensuality in Product Design: A Structured Approach." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 428-435.

Holden, P. (1989). "'Working-to-Rules': A Case of Taylor-Made Expert Systems." *Interacting with Computers*, 1 (2), 197-219.

Holt, R. W., D. A. Boehm-Davis and A. C. Schultz (1986). "The Effects of Structured, Multi-Level Documentation." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 122-128.

Hoogeveen, M. (1997). "Toward a Theory of the Effectiveness of Multimedia Systems." *International Journal of Human-Computer Interaction*, 9 (2), 151-168.

Hook, K., P. Sengers and G. Andersson (2003). "Sense and sensibility: evaluation and interactive art." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 241-248.

Hoppe, H. U. (1988). "Task-Oriented Parsing - A Diagnostic Method to be Used by Adaptive Systems." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 241-247.

Hoppe, H. U. and F. Schiele (1992). "Towards Task Models for Embedded Information Retrieval." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 173-180.

Hopper, S., H. Hambrose and P. Kanevsky (1996). "Real World Design in the Corporate Environment: Designing an Interface for the Technically Challenged." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 489-495.

Hornof, A. J. and T. Halverson (2003). "Cognitive strategies and eye movements for searching hierarchical computer displays." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 249-256.

Hornof, A. J. and D. E. Kieras (1999). "Cognitive Modeling Demonstrates How People Use Anticipated Location Knowledge of Menu Items." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 410-417.

Horvitz, E. (1999). "Principles of Mixed-Initiative User Interfaces." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 159-166.

Houde, S. (1992). "Iterative Design of an Interface for Easy 3-D Direct Manipulation." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 135-142.

Howard, S. (1997). "Trade-Off Decision Making in User Interface Design." *Behaviour and Information Technology*, 16 (2), 98-109.

Howes, A. (1994). "A Model of the Acquisition of Menu Knowledge by Exploration." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1, 445-451.

Howes, A. and S. J. Payne (1990). "Semantic Analysis During Exploratory Learning." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 399-405.

Howes, A. and R. M. Young (1991). "Predicting the Learnability of Task-Action Mappings." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 113-118.

Howie, D. E. and K. J. Vicente (1998). "Making the Most of Ecological Interface Design: The Role of Self-Explanation." *International Journal of Human-Computer Studies*, 49 (5), 651-674.

Hrebec, D. G. and M. Stiber (2001). "A survey of system administrator mental models and situation awareness." *Proceedings of the 2001 ACM SIGCPR Conference on Computer Personnel Research*, 166-172.

Hubona, G. S. and J. E. Blanton (1996). "Evaluating System Design Features." *International Journal of Human-Computer Studies*, 44 (1), 93-118.

Hudson, S. E. (1990). "Adaptive Semantic Snapping - A Technique for Semantic Feedback at the Lexical Level." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 65-70.

Huguenard, B. R., F. J. Lerch, B. W. Junker, R. J. Patz and R. E. Kass (1997). "Working-memory failure in phone-based interaction." *ACM Transactions on Computer-Human Interaction*, 4 (2), 67-102.

Hwang, F., S. Keates, P. Langdon and P. J. Clarkson (2003). "Multiple haptic targets for motion-impaired computer users." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 41-48.

Isa, B. S., J. M. Boyle, A. S. Neal and R. M. Simons (1983). "A Methodology for Objectively Evaluating Error Messages." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 68-71.

Ivory, M. Y. and M. A. Hearst (2002). "Statistical profiles of highly-rated web sites." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 367-374.

Ivory, M. Y., R. R. Sinha and M. A. Hearst (2001). "Empirically Validated Web Page Design Metrics." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 53-60.

Jacko, J. A., I. U. Scott, F. Sainfort, L. Barnard, P. J. Edwards, V. K. Emery, T. Kongnakorn, K. P. Moloney and B. S. Zorich (2003). "Older adults and visual impairment: what do exposure times and accuracy tell us about performance gains associated with multimodal feedback?" *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 33-40.

Jackson, S. L., J. Krajcik and E. Soloway (1998). "The Design of Guided Learner-Adaptable Scaffolding in Interactive Learning Environments." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 187-194.

Jacob, R. J. K. (1990). "What You Look At is What You Get: Eye Movement-Based Interaction Techniques." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 11-18.

Jacob, R. J. K., H. Ishii, G. Pangaro and J. Patten (2002). "A tangible interface for organizing information using a grid." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 339-346.

Jacob, R. J. K. and L. E. Sibert (1992). "The Perceptual Structure of Multidimensional Input Device Selection." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 211-218.

James, A. and J. C. Spohrer (1992). "Simulation-Based Learning Systems: Prototypes and Experiences." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 523-524.

Janssen, C., A. Weisbecker and J. Ziegler (1993). "Generating User Interfaces from Data Models and Dialogue Net Specifications." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 418-423.

Jeffries, R. and J. Rosenberg (1987). "Comparing a Form-Based and a Language-Based User Interface for Instructing a Mail Program." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 261-266.

Jellinek, H. D. and S. K. Card (1990). "Powermice and User Performance." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 213-220.

Jettmar, E. and C. Nass (2002). "Adaptive testing: effects on user performance." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 129-134.

John, B. E. (1990). "Extensions of GOMS Analyses to Expert Performance Requiring Perception of Dynamic Visual and Auditory Information." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 107-115.

John, B. E. and D. E. Kieras (1996). "Using GOMS for user interface design and evaluation: which technique?" *ACM Transactions on Computer-Human Interaction*, 3 (4), 287-219.

John, B. E. and A. Newell (1987). "Predicting the Time to Recall Computer Command Abbreviations." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 33-40.

John, B. E. and H. Packer (1995). "Learning and Using the Cognitive Walkthrough Method: A Case Study Approach." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 429-436.

John, B. E. and A. H. Vera (1992). "A GOMS Analysis of a Graphic, Machine-Paced, Highly Interactive Task." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 251-258.

Johnson, B. (1992). "TreeViz: Treemap Visualization of Hierarchically Structured Information." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 369-370.

Johnson, J. (1992). "Selectors: Going Beyond User-Interface Widgets." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 273-279.

Johnson, W., H. Jellinek, J. Leigh Klotz, R. Rao and S. Card (1993). "Bridging the Paper and Electronic Worlds: The Paper User Interface." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 507-512.

Jokela, T., N. Livari, J. Matero and M. Karukka (2003). "The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11." *Proceedings of the Latin American Conference on Human-computer Interaction*, 53-60.

Kamba, T., S. Elson, T. Harpold, T. Stamper and P. N. Sukaviriya (1996). "Using Small Screen Space More Efficiently." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 383-390.

Kambouri, M., M. Koppen, M. Villano and J.-C. Falmagne (1994). "Knowledge Assessment: Tapping Human Expertise by the QUERY Routine." *International Journal of Human-Computer Studies*, 40 (1), 119-151.

Kaplan, S. M., W. J. Tolone, D. P. Bogia and T. A. Phelps (1993). "Flexible, Active Support for Collaboration with ConversationBuilder." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 248.

Kaptelinin, V. (2003). "UMEA: translating interaction histories into project contexts." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 353-360.

Karat, J. and T. Dayton (1995). "Practical Education for Improving Software Usability." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 162-169.

Karsenty, L. (1999). "Cooperative Work and Shared Visual Context: An Empirical Study of Comprehension Problems in Side-by-Side and Remote Help Dialogues." *Human-Computer Interaction*, 14 (3), 283-315.

Keeler, M. A. and S. M. Denning (1991). "The Challenge of Interface Design for Communication Theory: From Interaction Metaphor to Contexts of Discovery." *Interacting with Computers*, 3 (3), 283-301.

Kellogg, W. A. and T. J. Breen (1987). "Evaluating User and System Models: Applying Scaling Techniques to Problems in Human-Computer Interaction." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 303-308.

Kellogg, W. A., C. Lewis and P. Polson (2000). "Introduction to This Special Issue on New Agendas for Human-Computer Interaction." *Human-Computer Interaction*, 15 (2/3), 69-74.

Kemp, R. H. and S. P. Smith (1994). "Domain and Task Representation for Tutorial Process Models." *International Journal of Human-Computer Studies*, 41 (3), 363-383.

Kieras, D., D. Meyer and J. Ballas (2001). "Towards Demystification of Direct Manipulation: Cognitive Modeling Charts the Gulf of Execution." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 128-135.

Kiesler, S., R. Kraut, V. Lundmark, W. Scherlis and T. Mukhopadhyay (1997). "Usability, Help Desk Calls, and Residential Internet Usage." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 536-537.

Kim, J. (1999). "An Empirical Study of Navigation Aids in Customer Interfaces." *Behaviour and Information Technology*, 18 (3), 213-224.

Kim, W. C. and J. D. Foley (1993). "Providing High-Level Control and Expert Assistance in the User Interface Presentation Design." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 430-437.

Kimerer, B. S. (1986). "A User Interface for Multiple-Process, Turnkey Systems Targeted for the Novice User." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 144-148.

Kirsh, D. (2001). "The Context of Work." *Human-Computer Interaction*, 16 (2/4), 306-322.

Kitajima, M. and P. G. Polson (1992). "A Computational Model of Skilled Use of a Graphical User Interface." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 241-249.

Kitajima, M. and P. G. Polson (1994). "A model-based analysis of errors in display-based HCI." *Conference Companion on Human Factors in Computing Systems*, 301-302.

Kitajima, M. and P. G. Polson (1996). "A Comprehension-Based Model of Exploration." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 324-331.

Kitajima, M. and P. G. Polson (1997). "A Comprehension-Based Model of Exploration." *Human-Computer Interaction*, 12 (4), 345-389.

Kitamura, Y., Y. Yamaguchi, I. Hiroshi, F. Kishino and M. Kawato (2003). "Things happening in the brain while humans learn to use new tools." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 417-424.

Klein, G., G. L. Kaempf, S. Wolf, M. Thorsden and T. Miller (1997). "Applying Decision Requirements to User-Centered Design." *International Journal of Human-Computer Studies*, 46 (1), 1-15.

Klemmer, S. R., J. Graham, G. J. Wolff and J. A. Landay (2003). "Books with voices: paper transcripts as a physical interface to oral histories." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 89-96.

Klemmer, S. R., M. Thomsen, E. Phelps-Goodman, R. Lee and J. A. Landay (2002). "Where do web sites come from?: capturing and interacting with design history." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 1-8.

Kline, R. L. and E. P. Glinert (1995). "Improving GUI Accessibility for People with Low Vision." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 114-121.

Knox, S. T., W. A. Bailey and E. F. Lynch (1989). "Directed Dialogue Protocols: Verbal Data for User Interface Design." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 283-287.

Koenemann, J. and N. J. Belkin (1996). "A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 205-212.

Koenemann-Belliveau, J., J. M. Carroll, M. B. Rosson and M. K. Singley (1994). "Comparative Usability Evaluation: Critical Incidents and Critical Threads." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 245-251.

Kontogiannis, T. and N. Linou (2001). "Making Instructions "Visible" on the Interface: An Approach to Learning Fault Diagnosis Skills through Guided Discovery." *International Journal of Human-Computer Studies*, 54 (1), 53-79.

Kraut, R. E., S. J. Hanson and J. M. Farber (1983). "Command Use and Interface Design." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 120-124.

Kurlander, D. and S. Feiner (1992). "Interactive Constraint-Based Search and Replace." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 609-618.

Kurniawan, S., A. King, D. G. Evans and P. Blenkhorn (2003). "Design and user evaluation of a joystick-operated full-screen magnifier." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 25-32.

Lai, K.-Y. and T. W. Malone (1991). "Object Lens: Letting End-Users Create Cooperative Work Applications." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 425-426.

Larson, K. and M. Czerwinski (1998). "Web Page Design: Implications of Memory, Structure and Scent for Information Retrieval." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 25-32.

Laurel, B., T. Oren and A. Don (1990). "Issues in Multimedia Interface Design: Media Integration and Interface Agents." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 133-139.

Lazonder, A. W. and H. v. d. Meij (1995). "Error-Information in Tutorial Documentation: Supporting Users' Errors to Facilitate Initial Skill Learning." *International Journal of Human-Computer Studies*, 42 (2), 185-206.

Lee, A. and N. Pennington (1994). "The Effects of Paradigm on Cognitive Activities in Design." *International Journal of Human-Computer Studies*, 40 (4), 577-601.

Lee, A. Y., P. W. Foltz and P. G. Polson (1994). "Memory for Task-Action Mappings: Mnemonics, Regularity and Consistency." *International Journal of Human-Computer Studies*, 40 (5), 771-794.

Lee, A. Y., P. G. Polson and W. A. Bailey (1989). "Learning and Transfer of Measurement Tasks." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 115-120.

Levy, E., J. Zacks, B. Tversky and D. Schiano (1996). "Gratuitous Graphics? Putting Preferences in Perspective." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 42-49.

Lewis, C., D. C. Hair and V. Schoenberg (1989). "Generalization, Consistency, and Control." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 1-5.

Lewis, C., P. Polson, C. Wharton and J. Rieman (1990). "Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 235-242.

Liao, C. and P. C. Palvia (2000). "The Impact of Data Models and Task Complexity on End-User Performance: An Experimental Investigation." *International Journal of Human-Computer Studies*, 52 (5), 831-845.

Liberman, H., S. Romberger and K. S. Eklundh (1993). "Mondrian: A Teachable Graphical Editor." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 144.

Lif, M. (1999). "User-Interface Modelling -- Adding Usability to Use Cases." *International Journal of Human-Computer Studies*, 50 (3), 243-262.

Lim, K. Y. and J. B. Long (1992). "A Method for (Recruiting) Methods: Facilitating Human Factors Input to System Design." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 549-556.

Lindeman, R. W., J. L. Sibert and J. K. Hahn (1999). "Towards Usable VR: An Empirical Study of User Interfaces for Immersive Virtual Environments." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 64-71.

Lindgaard, G. (1995). "Human Performance in Fault Diagnosis: Can Expert Systems Help?" *Interacting with Computers*, 7 (3), 254-272.

Lohse, J. (1991). "A Cognitive Model for the Perception and Understanding of Graphs." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 137-144.

Long, J. (1996). "Specifying Relations between Research and the Design of Human-Computer Interactions." *International Journal of Human-Computer Studies*, 44 (6), 875-920.

Lowgren, J. and T. Nordqvist (1992). "Knowledge-Based Evaluation as Design Support for Graphical User Interfaces." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 181-188.

Lueg, C. (1998). "Supporting Situated Actions in High Volume Conversational Data Situations." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 472-479.

Lundell, J. and M. Notess (1991). "Human Factors in Software Development: Models, Techniques, and Outcomes." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 145-151.

Mackay, W. E. (1991). "Triggers and Barriers to Customizing Software." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 153-160.

Mackay, W. E., R. Guindon, M. M. Mantei, L. Suchman and D. G. Tatar (1988). "Video: Data for Studying Human-Computer Interaction." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 133-137.

MacKenzie, I. S. and W. Buxton (1992). "Extending Fitts' Law to Two-Dimensional Tasks." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 219-226.

MacKenzie, I. S., T. Kauppinen and M. Silfverberg (2001). "Accuracy Measures for Evaluating Computer Pointing Devices." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 9-16.

MacKenzie, I. S. and A. Oniszczak (1998). "A Comparison of Three Selection Techniques for Touchpads." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 336-343.

MacKenzie, I. S., A. Sellen and W. Buxton (1991). "A Comparison of Input Devices in Elemental Pointing and Dragging Tasks." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 161-166.

MacLean, A., V. Bellotti, R. Young and T. Moran (1991). "Reaching Through Analogy: A Design Rationale Perspective on Roles of Analogy." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 167-172.

MacLean, A., K. Carter, L. Lovstrand and T. Moran (1990). "User-Tailorable Systems: Pressing the Issues with Buttons." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 175-182.

MacLean, A., R. M. Young and T. P. Moran (1989). "Design Rationale: The Argument Behind the Artifact." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 247-252.

Maguire, M. (1999). "A Review of User-Interface Design Guidelines for Public Information Kiosk Systems." *International Journal of Human-Computer Studies*, 50 (3), 263-286.

Malone, T. W. (1982). "Heuristics for Designing Enjoyable User Interface: Lessons from Computer Games." *Proceedings of Human Factors in Computer Systems*, 63-68.

Mankoff, J., A. K. Dey, G. Hsieh, J. Kientz, S. Lederer and M. Ames (2003). "Heuristic evaluation of ambient displays." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 169-176.

Mantei, M. M., R. M. Baecker, A. J. Sellen, W. A. S. Buxton, T. Milligan and B. Wellman (1991). "Experiences in the Use of a Media Space." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 203-208.

Marcus, A., W. B. Cowan and W. Smith (1989). "Color in User Interface Design: Functionality and Aesthetics." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 25-27.

Marti, P. (1996). "Task-centred design." *ACM SIGCHI Bulletin*, 28 (3), 65-70.

Maulsby, D. L. (1992). "Prototyping an Instructible Interface: Moctec." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 153-154.

McCarthy, J. C., V. C. Miles and A. F. Monk (1991). "An Experimental Study of Common Ground in Text-Based Communication." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 209-215.

McCoy, K. F. (1983). "Correcting Misconceptions: What to Say when the User is Mistaken." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 197-201.

McDonald, J. E., M. E. Molander and R. W. Noel (1988). "Color-Coding Categories in Menus." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 101-106.

McGee, D. R., P. R. Cohen, R. M. Wesson and S. Horman (2002). "Comparing paper and tangible, multimodal tools." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 407-414.

McGee, D. R., M. Pavel and P. R. Cohen (2001). "Context Shifts: Extending the Meanings of Physical Objects with Language." *Human-Computer Interaction*, 16 (2/4), 351-362.

McGraw, K. L. (1997). "Defining and Designing the Performance-Centered Interface: Moving Beyond the User-Centered Interface." *Interactions*, 4 (2), 19-26.

McKendree, J. and J. Zaback (1988). "Planning for Advising." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 179-184.

McMillan, W. W. (1992). "Computing for Users with Special Needs and Models of Computer-Human Interaction." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 143-148.

Mcroy, S. (1998). "Preface: Detecting, Repairing and Preventing Human -- Machine Miscommunication." *International Journal of Human-Computer Studies*, 48 (5), 547-552.

Mcroy, S. W. (1998). "Achieving Robust Human-Computer Communication." *International Journal of Human-Computer Studies*, 48 (5), 681-704.

Miles, G. E., A. Howes and A. Davies (2000). "A Framework for Understanding Human Factors in Web-Based Electronic Commerce." *International Journal of Human-Computer Studies*, 52 (1), 131-163.

Miller, G. A. (1956). "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capability for Processing Information." *Pyschological Review*, 63, 81-97.

Miller, L. A. and J. John C. Thomas (1999). "Behavioral Issues in the Use of Interactive Systems." *International Journal of Human-Computer Studies*, 51 (2), 169-196.

Miller, L. A. and K. M. Stanney (1997). "The Effect of Pictogram-Based Interface Design on Human-Computer Performance." *International Journal of Human-Computer Interaction*, 9 (2), 119-131.

Mills, C. B., K. F. Bury, P. Reed, T. L. Roberts, B. Tognazzini, A. Wichansky and J. Gould (1986). " Usability Testing in the Real World." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 212-215.

Mills, S. (2000). "The Importance of Task Analysis in Usability Context Analysis -- Designing for Fitness for Purpose." *Behaviour and Information Technology*, 19 (1), 57-68.

Minneman, S. L. and S. A. Bly (1991). "Managing a trois: A Study of a Multi-User Drawing Tool in Distributed Design Work." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 217-224.

Moll-Carrillo, H. J., G. Salomon, M. Marsh, J. F. Suri and P. Spreenberg (1995). "Articulating a Metaphor through User-Centered Design." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 566-572.

Monk, A. (1999). "Modelling Cyclic Interaction." *Behaviour and Information Technology*, 18 (2), 127-139.

Monk, A. and S. Howard (1998). "Methods & tools: the rich picture: a tool for reasoning about work context." *Interactions*, 5 (2), 21-30.

Moore, J. D. (1989). "Responding to "Huh?": Answering Vaguely Articulated Follow-Up Questions." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 91-96.

Moran, T. P. (1983). "Getting Into a System: External-Internal Task Mapping Analysis." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 45-49.

Moriyon, R., P. Szekely and R. Neches (1994). "Automatic Generation of Help from Interface Design Models." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 225-231.

Morley, S., H. Petrie, A.-M. O'Neill and P. Mcnally (1999). "Auditory Navigation in Hyperspace: Design and Evaluation of a Non-Visual Hypermedia System for Blind Users." *Behaviour and Information Technology*, 18 (1), 18-26.

Morris, M. G. and A. P. Dillon (1996). "The Importance of Usability in the Establishment of Organizational Software Standards for End User Computing." *International Journal of Human-Computer Studies*, 45 (2), 243-258.

Mountford, S. J., B. Buxton, M. Krueger, B. Laurel and L. Vertelney (1989). "Drama and Personality in User Interface Design." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 105-108.

Mrazek, D. and M. Rafeld (1992). "Integrating Human Factors on a Large Scale: "Product Usability Champions"." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 565-570.

Mrenak, G. (1990). "Evolving concepts, or why users often don't recognize the software they asked for." *Proceedings of the Washington Ada Symposium*, 17-22.

Muller, M. J. (1988). "Multifunctional Cursor for Direct Manipulation User Interfaces." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 89-94.

Muller, M. J. (1991). "PICTIVE - An Exploration in Participatory Design." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 225-231.

Mulligan, R. M., M. W. Altom and D. K. Simkin (1991). "User Interface Design in the Trenches: Some Tips on Shooting from the Hip." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 232-236.

Mulligan, R. M., M. Dieli, J. Nielsen, S. Poltrock, D. Rosenberg and S. E. Rudman (1992). "Designing Usable Systems Under Real-World Constraints: A Practitioners Forum." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 149-152.

Murata, A. (1998). "Improvement of Pointing Time by Predicting Targets in Pointing with a PC Mouse." *International Journal of Human-Computer Interaction*, 10 (1), 23-32.

Myers, B. A. (1985). "The Importance of Percent-Done Progress Indicators for Computer-Human Interfaces." *Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems*, 11-17.

Myers, B. A., R. Wolf, K. Potosnak and C. Graham (1993). "Heuristics in Real User Interfaces." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 304-307.

Mynatt, B. T., L. M. Leventhal, K. Instone, J. Farhat and D. S. Rohlman (1992). "Hypertext or Book: Which is Better for Answering Questions?" *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 19-25.

Mynatt, E. D. (1997). "Transforming Graphical Interfaces Into Auditory Interfaces for Blind Users." *Human-Computer Interaction*, 12 (1/2), 7-45.

Mynatt, E. D. and G. Weber (1994). "Nonvisual Presentation of Graphical User Interfaces: Contrasting Two Approaches." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 166-172.

Nakatani, L. H. and J. A. Rohrlich (1983). "Soft Machines: A Philosophy of User-Computer Interface Design." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 19-23.

Nardi, B. A. and J. A. Johnson (1994). "User Preferences for Task-Specific vs. Generic Application Software." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 392-398.

Nathan, M. J. (1990). "Empowering the Student: Prospects for an Unintelligent Tutoring System." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 407-414.

Neerincx, M. and P. d. Greef (1993). "How to Aid Non-Experts." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 165-171.

Neerincx, M. A. and H. P. d. Greef (1998). "Cognitive Support: Extending Human Knowledge and Processing Capacities." *Human-Computer Interaction*, 13 (1), 73-106.

Neilson, I. and J. Lee (1994). "Conversations with Graphics: Implications for the Design of Natural Language/Graphics Interfaces." *International Journal of Human-Computer Studies*, 40 (3), 509-541.

Nichols, S. and F. E. Ritter (1995). "A Theoretically Motivated Tool for Automatically Generating Command Aliases." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 393-400.

Nielsen, J. (1992). "Finding Usability Problems Through Heuristic Evaluation." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 373-380.

Nielsen, J. (1994). "Enhancing the Explanatory Power of Usability Heuristics." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 152-158.

Nielsen, J. (1999). "User interface directions for the Web." *Communications of the ACM*, 42 (1), 65-72.

Nielsen, J., R. M. Bush, T. Dayton, N. E. Mond, M. J. Muller and R. W. Root (1992). "Teaching Experienced Developers to Design Graphical User Interfaces." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 557-564.

Nielsen, J., R. Kerr, D. Rosenberg, G. Salomon, H. Desurvire, R. Molich and T. Stewart (1993). "Comparative Design Review: An Exercise in Parallel Design." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 414-417.

Nielsen, J. and R. Molich (1990). "Heuristic Evaluation of User Interfaces." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 249-256.

Nielson, J. (1993). "Noncommand user interfaces." *Communications of the ACM*, 36 (4), 83-99.

Nigay, L. and J. Coutaz (1993). "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 172-178.

Noirhomme-Fraiture, M. and J. M. Vanderdonckt (1993). "Screen usability guidelines for persons with disabilities." *INTERACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems*, 25-26.

Norman, D. A. (1982). "Steps toward a cognitive engineering: Design rules based on analyses of human error." *Proceedings of the SIGCHI Conference on Human Factors in Computing System*, 378-382.

Norman, D. A. (1983). "Design Principles for Human-Computer Interfaces." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 1-10.

O'Day, V. L. and R. Jeffries (1993). "Orienteering in an Information Landscape: How Information Seekers Get from Here to There." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 438-445.

O'Hara, K. P. and S. J. Payne (1999). "Planning and the User Interface: The Effects of Lockout Time and Error Recovery Cost." *International Journal of Human-Computer Studies*, 50 (1), 41-59.

O'Keefe, R. M., M. Cole, P. Y. K. Chau, A. Massey, M. Montoya-Weiss and M. Perry (2000). "From the User Interface to the Consumer Interface: Results from a Global Experiment." *International Journal of Human-Computer Studies*, 53 (4), 611-628.

Olsson, E. and J. Gulliksen (1999). "A Corporate Style Guide That Includes Domain Knowledge." *International Journal of Human-Computer Interaction*, 11 (4), 317-338.

Oostendorp, H. v., J. Preece and A. G. Arnold (1999). "Designing Multimedia for Human Needs and Capabilities." *Interacting with Computers*, 12 (1), 1-5.

Oosterholt, R., M. Kusano and G. d. Vries (1996). "Interaction Design and Human Factors Support in the Development of a Personal Communicator for Children." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 450-457.

Otter, M. and H. Johnson (2000). "Lost in Hyperspace: Metrics and Mental Models." *Interacting with Computers*, 13 (1), 1-40.

Oviatt, S. (1997). "Multimodal Interactive Maps: Designing for Human Performance." *Human-Computer Interaction*, 12 (1/2), 93-129.

Oviatt, S. and W. Wahlster (1997). "Introduction to This Special Issue on Multimodal Interfaces." *Human-Computer Interaction*, 12 (1/2), 1-5.

Papstein, P. V. and M. Frese (1988). "Transferring Skills from Training to the Actual Work Situation: The Role of Task Application Knowledge, Action Styles and Job Decision Latitude." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 55-60.

Parush, A., R. Nadir and A. Shtub (1998). "Evaluating the Layout of Graphical User Interface Screens: Validation of a Numerical Computerized Model." *International Journal of Human-Computer Interaction*, 10 (4), 343-360.

Patel, S. C., C. G. Drury and V. L. Shalin (1998). "Effectiveness of Expert Semantic Knowledge as a Navigational Aid within Hypertext." *Behaviour and Information Technology*, 17 (6), 313-324.

Pausch, R., L. Vogtle and M. Conway (1992). "One Dimensional Motion Tailoring for the Disabled: A User Study." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 405-411.

Pawlak, W. S. and K. J. Vicente (1996). "Inducing Effective Operator Control through Ecological Interface Design." *International Journal of Human-Computer Studies*, 44 (5), 653-688.

Perez-Quinones, M. A. and J. L. Sibert (1996). "Negotiating User-Initiated Cancellation and Interruption Requests." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 2**,** 267-268.

Perkins, R. (1995). "The Interchange Online Network: Simplifying Information Access." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 558-565.

Petersen, R. J., W. W. Banks and D. I. Gertman (1982). "Performance-Based Evaluation of Graphic Displays for Nuclear Power Plant Control Rooms." *Proceedings of Human Factors in Computer Systems*, 182-189.

Philips, B. H., M. Rahman and J. Jarvinen (2001). "Building a Human Factors "Knowledge Shelf" as a Collaborative Information Tool for Designers." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 98-103.

Pierce, J. S. and R. Pausch (2002). "Comparing voodoo dolls and HOMER: exploring the importance of feedback in virtual environments." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 105-112.

Pirhonen, A., S. Brewster and C. Holguin (2002). "Gestural and audio metaphors as a means of control for mobile devices." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 291-298.

Pirolli, P. and S. Card (1995). "Information Foraging in Information Access Environments." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 51-58.

Plaisant, C. and B. Shneiderman (1991). "Scheduling ON-OFF Home Control Devices." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 459-460.

Plowman, L., R. Luckin, D. Laurillard, M. Stratfold and J. Taylor (1999). "Designing Multimedia for Learning: Narrative Guidance and Narrative Construction." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 310-317.

Pollack, M. E. (1985). "Information Sought and Information Provided: An Empirical Study of User/Expert Dialogues." *Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems*, 155-159.

Polson, P. G., S. Bovair and D. Kieras (1987). "Transfer Between Text Editors." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 27-32.

Poltrock, S. E., D. D. Steiner and P. N. Tarlton (1986). "Graphic Interfaces for Knowledge-Based System Development." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 9-15.

Potosnak, K. M., P. J. Hayes, M. B. Rosson, M. L. Schneider and J. A. Whiteside (1986). "Classifying Users: A Hard Look at Some Controversial Issues." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 84-88.

Quek, F. K. H. and M. C. Petro (1993). "Human-Machine Perceptual Cooperation." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 123-130.

Quintana, C., J. Eng, A. Carra, H.-K. Wu and E. Soloway (1999). "Symphony: A Case Study in Extending Learner-Centered Design Through Process Space Analysis." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 473-480.

Rappin, N., M. Guzdial, M. Realff and P. Ludovice (1997). "Balancing Usability and Learning in an Interface." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 479-486.

Reed, A. V. (1982). "Error-Correcting Strategies and Human Interaction with Computer Systems." *Proceedings of Human Factors in Computer Systems*, 236-238.

Rekimoto, J., B. Ullmer and H. Oba (2001). "DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 269-276.

Revesman, M. E. and J. S. Greenstein (1983). "Application of a Model of Human Decision Making for Human/Computer Communication." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 107-111.

Rice, J., A. Farquhar, P. Piernot and T. Gruber (1996). "Using the Web Instead of a Window System." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 103-110.

Rich, E. (1999). "Users are Individuals: Individualizing User Models." *International Journal of Human-Computer Studies*, 51 (2), 323-338.

Rieman, J. (1996). "A field study of exploratory learning strategies." *ACM Transactions on Computer-Human Interaction*, 3 (3), 189-218.

Rieman, J., S. Davies, D. C. Hair, M. Esemplare, P. Polson and C. Lewis (1991). "An Automated Cognitive Walkthrough." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 427-428.

Risden, K., M. Czerwinski, S. Worley, L. Hamilton, J. Kubiniec, H. Hoffman, N. Mickel and E. Loftus (1998). "Interactive Advertising: Patterns of Use and Effectiveness." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 219-224.

Robertson, S. P. and J. B. Black (1983). "Planning Units in Text Editing Behavior." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 217-221.

Rodden, T., Y. Rogers, J. Halloran and I. Taylor (2003). "Designing novel interactional workspaces to support face to face consultations." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 57-64.

Roesler, A. W. and S. G. McLellan (1995). "What Help Do Users Need?: Taxonomies for On-Line Information Needs and Access Methods." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 437-441.

Ropa, A. and B. Ahlstrom (1992). "A Case Study of a Multimedia Co-Working Task and the Resulting Interface Design of a Collaborative Communication Tool." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 649-650.

Ross, B. H. and T. P. Moran (1983). "Remindings and Their Effects in Learning a Text Editor." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 222-225.

Rosson, M. B. and J. M. Carroll (1995). "Integrating Task and Software Development for Object-Oriented Applications." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 377-384.

Rosson, M. B., S. Maass and W. A. Kellogg (1987). "Designing for Designers: An Analysis of Design Practice in the Real World." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 137-142.

Rosson, M. B. and C. D. Seals (2001). "Teachers as Simulation Programmers: Minimalist Learning and Reuse." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 237-244.

Roth, S. F., J. Kolojejchick, J. Mattis and J. Goldstein (1994). "Interactive Graphic Design Using Automatic Presentation Knowledge." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 112-117.

Roth, S. F. and J. Mattis (1990). "Data Characterization for Intelligent Graphics Presentation." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 193-200.

Rousseau, G. K., B. A. Jamieson, W. Rogers, S. E. Mead and R. A. Sit (1998). "Assessing the Usability of On-Line Library Systems." *Behaviour and Information Technology*, 17 (5), 274-281.

Rowley, D. E. and D. G. Rhoades (1992). "The Cognitive Jogthrough: A Fast-Paced User Interface Evaluation Procedure." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 389-395.

Rubine, D. (1992). "Combining Gestures and Direct Manipulation." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 659-660.

Rudnicky, A. I. and A. G. Hauptmann (1991). "Models for Evaluating Interaction Protocols in Speech Recognition." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 285-291.

Russo, P. and S. Boor (1993). "How fluent is your interface?: designing for international users." *Proceedings of the SIGCHI Conference on Human Factors in Computing System*, 342-347.

Salber, D., A. K. Day and G. D. Abowd (1999). "The Context Toolkit: Aiding the Development of Context-Enabled Applications." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 434-441.

Salvucci, D. D. and F. J. Lee (2003). "Simple cognitive modeling in a complex cognitive architecture." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 265-272.

Satzinger, J. and L. Olfman (1992). "A Research Program to Assess User Perceptions of Group Work Support." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 99-106.

Savage, R. E., J. K. Habinek and T. W. Barnhart (1982). "The Design, Simulation, and Evaluation of a Menu Driven User Interface." *Proceedings of Human Factors in Computer Systems*, 36-40.

Savidis, A. and C. Stephanidis (1995). "Developing Dual Interfaces for Integrating Blind and Sighted Users: The HOMER UIMS." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 106-113.

Scapin, D. L. (1982). "Computer Commands Labelled by Users versus Imposed Commands and the Effect of Structuring Rules on Recall." *Proceedings of Human Factors in Computer Systems*, 17-19.

Scapin, D. L. (1990). "Organizing human factors knowledge for the evaluation and design of interfaces." *International Journal of Human-Computer Interaction*, 2 (3), 203-229.

Schaik, P. v. and J. Ling (2001). "The Effects of Frame Layout and Differential Background Contrast on Visual Search Performance in Web Pages." *Interacting with Computers*, 13 (5), 513-525.

Scholtz, J. C., T. Salvador, P. Lockhart and J. Newbery (1997). "Design: No Job too Small." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 447-454.

Schrier, J. R., E. L. Williams, K. S. MacDonell, L. A. Peterson, P. F. Strijland, A. M. Wichansky and J. R. Williams (1992). "HCI Standards on Trial: You be the Jury." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 635-638.

Scott, S. D., N. Lesh and G. W. Klau (2002). "Investigating human-computer optimization." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 155-162.

Sebrechts, M. M. and M. L. Swartz (1991). "Question Asking as a Tool for Novice Computer Skill Acquisition." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 293-299.

Seffah, A., R. Djouab and H. Antunes (2001). "Comparing and reconciling usability-centered and use case-driven requirements engineering processes." *Proceedings of the 2nd Australasian Conference on User Interface*, 132-139.

Sellen, A. J., R. Murphy and K. L. Shaw (2002). "How knowledge workers use the web." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 227-234.

Sharlin, E., Y. Itoh, B. Watson, Y. Kitamura, S. Sutphen and L. Liu (2002). "Cognitive cubes: a tangible user interface for cognitive assessment." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 347-354.

Shaw, M. (1986). "An Input-Output Model for Interactive Systems." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 261-273.

Shaw, M. L. G. and B. R. Gaines (1991). "Supporting Personal Networking Through Computer Networking." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 437-438.

Shneiderman, B. (1986). "Seven Plus or Minus Two Central Issues in Human-Computer Interaction." *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, 343-349.

Shneiderman, B. (1997). "Designing Information-Abundant Web Sites: Issues and Recommendations." *International Journal of Human-Computer Studies*, 47 (1), 5-29.

Shneiderman, B. (2000). "Universal Usability." *Communications of the ACM*, 43 (5), 84-91.

Shneiderman, B. and P. Maes (1997). "Direct Manipulation vs Interface Agents." *Interactions*, 4 (6), Shneiderman, B., C. Williamson and C. Ahlberg (1992). "Dynamic Queries: Database Searching by Direct Manipulation." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 669-670.

Shrager, J. and D. Klahr (1983). "Learning in an Instructionless Environment: Observation and Analysis." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 226-229.

Shubin, H., D. Falck and A. G. Johansen (1996). "Exploring color in interface design." *Interactions*, 3 (4), 36-48.

Shum, S. B. and C. McKnight (1997). "World Wide Web Usability: Introduction to This Special Issue." *International Journal of Human-Computer Studies*, 47 (1), 1-4.

Sidhu, J. and C. J. Hinde (1997). "An Analysis of the Use of Natural Language Processing Systems in Business." *Behaviour and Information Technology*, 16 (3), 145-157.

Simpson, N. (1999). "Managing the Use of Style Guides in an Organisational Setting: Practical Lessons in Ensuring UI Consistency." *Interacting with Computers*, 11 (3), 323-351.

Siochi, A. C. and H. R. Hartson (1989). "Task-Oriented Representation of Asynchronous User Interfaces." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 183-188.

Siochi, A. C. and D. Hix (1991). "A Study of Computer-Supported User Interface Evaluation Using Maximal Repeating Pattern Analysis." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 301-305.

Smets, G., K. Overbeeke and W. Gaver (1994). "Form-Giving: Expressing the Nonobvious." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 79-84.

Smith, A. and L. Dunckley (2002). "Prototype evaluation and redesign: structuring the design space through contextual techniques." *Interacting with Computers*, 14 (6), 821-843.

Soloway, E., S. L. Jackson, J. Klein, C. Quintana, J. Reed, J. Spitulnik, S. J. Stratford, S. Studer, J. Eng and N. Scala (1996). "Learning Theory in Practice: Case Studies of Learner-Centered Design." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 189-196.

Somberg, B. L. (1987). "A Comparison of Rule-Based and Positionally Constant Arrangements of Computer Menu Items." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 255-260.

Sonnentag, S. (1996). "Planning and Knowledge about Strategies: Their Relationship to Work Characteristics in Software Design." *Behaviour and Information Technology*, 15 (4), 213-225.

Soto, R. (1999). "Learning and Performing by Exploration: Label Quality Measured by Latent Semantic Analysis." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 418-425.

Spool, J., C. D. Allen, D. Ballman, V. Begg, H. H. Miller-Jacobs, M. Muller and J. Nielsen (1993). "User Involvement in the Design Process: Why, When and How?" *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 251-254.

Springett, M. (1998). "Linking surface error characteristics to root problems in user-based evaluation studies." *Proceedings of the Working Conference on Advanced Visual Interfaces*, 102-113.

Staples, L. (1993). "Representation in Virtual Space: Visual Convention in the Graphical User Interface." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 348-354.

Stary, C. and M. F. Peschl (1998). "Representation Still Matters: Cognitive Engineering and User Interface Design." *Behaviour and Information Technology*, 17 (6), 338-360.

Stasko, J. T. (1991). "Using Direct Manipulation to Build Algorithm Animations by Demonstration." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 307-314.

Steiner, K. E. and T. G. Moher (1992). "Graphic StoryWriter: An Interactive Environment for Emergent Storytelling." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 357-364.

Stewart, J., B. B. Bederson and A. Druin (1999). "Single Display Groupware: A Model for Co-Present Collaboration." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 286-293.

Stubblefield, W. A. (1998). "Patterns of Change in Design Metaphor: A Case Study." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 73-80.

Suh, B., A. Woodruff, R. Rosenholtz and A. Glass (2002). "Popout prism: adding perceptual principles to overview+detail document interfaces." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 251-258.

Suhm, B., A. Waibel and B. Myers (1999). "Model-Based and Empirical Evaluation of Multimodal Interactive Error Correction." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 584-591.

Sukaviriya, P. N., E. Isaacs and K. Bharat (1992). "Multimedia Help: A Prototype and an Experiment." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 433-434.

Sullivan, K. (1996). "The Windows 95 User Interface: A Case Study in Usability Engineering." *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, 1**,** 473-480.

Sumner, T., N. Bonnardel and B. H. Kallak (1997). "The Cognitive Ergonomics of Knowledge-Based Design Support Systems." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 83-90.

Sutcliffe, A. (1997). "Task-Related Information Analysis." *International Journal of Human-Computer Studies*, 47 (2), 223-257.

Sutcliffe, A. and J. Carroll (1999). "Designing Claims for Reuse in Interactive Systems Design." *International Journal of Human-Computer Studies*, 50 (3), 213-241.

Sutcliffe, A. and P. Faraday (1994). "Designing Presentation in Multimedia Interfaces." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 92-98.

Sutcliffe, A., M. Ryan, A. Doubleday and M. Springett (2000). "Model Mismatch Analysis: Towards a Deeper Explanation of Users' Usability Problems." *Behaviour and Information Technology*, 19 (1), 43-55.

Sutcliffe, A. G., M. Ennis and J. Hu (2000). "Evaluating the Effectiveness of Visual User Interfaces for Information Retrieval." *International Journal of Human-Computer Studies*, 53 (5), 741-763.

Sutcliffe, A. G. and K. D. Kaur (2000). "Evaluating the Usability of Virtual Reality User Interfaces." *Behaviour and Information Technology*, 19 (6), 415-426.

Tan, D. S., D. Gergle, P. Scupelli and R. Pausch (2003). "With similar visual angles, larger displays improve spatial performance." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 217-224.

Te'eni, D. (1996). "Teaching Human-Computer Interaction in Context: An Illustrative Lesson on Windows." *Behaviour and Information Technology*, 15 (2), 101-112.

Teal, S. L. and A. I. Rudnicky (1992). "A Performance Model of System Delay and User Strategy Selection." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 295-305.

Teasley, B. E. (1994). "The Effects of Naming Style and Expertise on Program Comprehension." *International Journal of Human-Computer Studies*, 40 (5), 757-770.

Teitelbaum, R. C. and R. E. Granda (1983). "The Effects of Positional Constancy on Searching Menus for Information." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 150-153.

Telles, M. (1990). "Updating an Older Interface." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 243-247.

Tero, A. and P. Briggs (1994). "Consistency versus Compatibility: A Question of Levels?" *International Journal of Human-Computer Studies*, 40 (5), 879-894.

Terrier, P. and J.-M. Cellier (1999). "Depth of Processing and Design-Assessment of Ecological Interfaces: Task Analysis." *International Journal of Human-Computer Studies*, 50 (4), 287-307.

Terveen, L., J. McMackin, B. Amento and W. Hill (2002). "Specifying preferences based on user history." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 315-322.

Terwilliger, R. B. and P. G. Polson (1997). "Relationships between users' and interfaces' task representations." *Proceedings of the SIGCHI conference on Human factors in computing systems*, 99-106.

Tetzlaff, L. (1987). "Transfer of Learning: Beyond Common Elements." *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 205-210.

Tetzlaff, L. and D. R. Schwartz (1991). "The Use of Guidelines in Interface Design." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 329-333.

Thiel, D. D. (1991). "The Cue Ball as Part of a Gestural Interface." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 463.

Thomas, I. (1990). "The software process as a goal-directed activity." *Proceedings of the International Software Process Workshop*, 137-139.

Thovtrup, H. and J. Nielsen (1991). "Assessing the Usability of a User Interface Standard." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 335-341.

Tognazzini, B. (1993). "Principles, Techniques, and Ethics of Stage Magic and Their Potential Application to Human Interface Design." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 355-362.

Tognazzini, B. (1994). "The "Starfire" Video Prototype Project: A Case History." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 99-105.

Tolmie, P., J. Pycock, T. Diggins, A. MacLean and A. Karsenty (2002). "Unremarkable computing." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 399-406.

Tractinsky, N. (1997). "Aesthetics and Apparent Usability: Empirically Assessing Cultural and Methodological Issues." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 115-122.

Trewin, S. and H. Pain (1999). "Keyboard and Mouse Errors Due to Motor Disabilities." *International Journal of Human-Computer Studies*, 50 (2), 109-144.

Trudel, C.-I. and S. J. Payne (1996). "Self-Monitoring During Exploration of an Interactive Device." *International Journal of Human-Computer Studies*, 45 (6), 723-747.

Trumbly, J. E., K. P. Arnett and P. C. Johnson (1994). "Productivity Gains via an Adaptive User Interface: An Empirical Analysis." *International Journal of Human-Computer Studies*, 40 (1), 63-81.

van Setten, M., G. C. van der Veer and S. Brinkkemper (1997). "Comparing interaction design techniques: a method for objective comparison to find the conceptual basis for interaction design." *Proceedings of the Conference on Designing Interactive Systems*, 349-357.

Vanderdonckt, J. (1999). "Development Milestones towards a Tool for Working with Guidelines." *Interacting with Computers*, 12 (2), 81-118.

Vanderheiden, G. (2000). "Fundamental principles and priority setting for universal usability." *Proceedings of the ACM Conference on Universal Usability*, 32-37.

Venolia, G. D. and C. Neustaedter (2003). "Understanding sequence and reply relationships within email conversations: a mixed-model visualization." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 361-368.

Vinson, N. G. (1999). "Design Guidelines for Landmarks to Support Navigation in Virtual Environments." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 278-285.

Vora, P. R. (1998). "Designing for the web: A survey." *Interactions*, 5 (3), 13-30.

Vora, P. R., M. G. Helander and V. L. Shalin (1994). "Evaluating the Influence of Interface Styles and Multiple Access Paths in Hypertext." *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1**,** 323-329.

Vredenburg, K., J.-Y. Mao, P. W. Smith and T. Carey (2002). "A survey of user-centered design practice." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 471-478.

Wagner, A., P. Curran and R. O'Brien (1995). "Drag Me, Drop Me, Treat Me Like an Object." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 525-530.

Walker, N. and J. R. Olson (1988). "Designing Keybindings to be Easy to Learn and Resistant to Forgetting Even When the Set of Commands is Large." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 201-206.

Walker, N. and J. B. Smelcer (1990). "A Comparison of Selection Times from Walking and Pull-Down Menus." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 221-225.

Wang, Y. and C. L. MacKenzie (1999). "Object Manipulation in Virtual Environments: Relative Size Matters." *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1**,** 48-55.

Wann, J. and M. Mon-Williams (1996). "What Does Virtual Reality NEED?: Human Factors Issues in the Design of Three-Dimensional Computer Environments." *International Journal of Human-Computer Studies*, 44 (6), 829-847.

Ware, C. and W. Knight (1992). "Orderable Dimensions of Visual Texture Useful for Data Display: Orientation, Size, and Contrast." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 203-209.

Wenzel, E. M., W. W. Gaver, S. H. Foster, H. Levkowitz and R. Powell (1993). "Perceptual vs. Hardware Performance in Advanced Acoustic Interface Design." *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 363-366.

Wharton, C., J. Bradford, R. Jeffries and M. Franzke (1992). "Applying Cognitive Walkthroughs to More Complex User Interfaces: Experiences, Issues, and Recommendations." *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 381-388.

Whiteside, J., S. Jones, P. S. Levy and D. Wixon (1985). "User Performance with Command, Menu, and Iconic Interfaces." *Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems*, 185-191.

Whittaker, S., S. E. Brennan and H. H. Clark (1991). "Co-Ordinating Activity: An Analysis of Interaction in Computer-Supported Co-Operative Work." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 361-367.

Whittaker, S., J. Hirschberg, B. Amento, L. Stark, M. Bacchiani, P. Isenhour, L. Stead, G. Zamchick and A. Rosenberg (2002). "SCANMail: a voicemail interface that makes speech browsable, readable and searchable." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 275-282.

Wiecha, C., W. Bennett, S. Boies and J. Gould (1989). "Generating Highly Interactive User Interfaces." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 277-282.

Wiedenbeck, S. (1999). "The Use of Icons and Labels in an End User Application Program: An Empirical Study of Learning and Retention." *Behaviour and Information Technology*, 18 (2), 68-82.

Wiedenbeck, S., P. L. Zila and D. S. McConnell (1995). "End-User Training: An Empirical Study Comparing On-Line Practice Methods." *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1**,** 74-81.

Wilcox, E. M., J. W. Atwood, M. M. Burnett, J. J. Cadiz and C. R. Cook (1997). "Does Continuous Visual Feedback Aid Debugging in Direct-Manipulation Programming Systems?" *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 258-265.

Wilde, N. and C. Lewis (1990). "Spreadsheet-Based Interactive Graphics: From Prototype to Tool." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 153-159.

Wilson, S., M. Bekker, P. Johnson and H. Johnson (1997). "Helping and Hindering User Involvement -- A Tale of Everyday Design." *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, 1**,** 178-185.

Wilson, S. and P. Johnson (1995). "Empowering users in a task-based approach to design." *Proceedings of the Symposium on Designing Interactive Systems*, 25-31.

Wixon, D., K. Holtzblatt and S. Knox (1990). "Contextual Design: An Emergent View of System Design." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 329-336.

Wixon, D., J. Whiteside, M. Good and S. Jones (1983). "Building a User-Defined Interface." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 24-27.

Woodruff, A., A. Faulring, R. Rosenholtz, J. Morrsion and P. Pirolli (2001). "Using Thumbnails to Search the Web." *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, 198-205.

Wright, P. (1983). "Manual Dexterity: A User-Oriented Approach to Creating Computer Documentation." *Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems*, 11-18.

Wright, P. and A. Lickorish (1994). "Menus and Memory Load: Navigation Strategies in Interactive Search Tasks." *International Journal of Human-Computer Studies*, 40 (6), 965-1008.

Wroblewski, D. A., T. P. McCandless and W. C. Hill (1991). "DETENTE: Practical Support for Practical Action." *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 195-202.

Yang, J., R. Stiefelhagen, U. Meier and A. Waibel (1998). "Visual Tracking for Multimodal Human Computer Interaction." *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1**,** 140-147.

Yang, Y. (1988). "A New Conceptual Model for Interactive User Recovery and Command Reuse Facilities." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 165-170.

Young, R. M., T. R. G. Green and T. Simon (1989). "Programmable User Models for Predictive Evaluation of Interface Designs." *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, 15-19.

Young, R. M. and A. MacLean (1988). "Choosing Between Methods: Analysing the User's Decision Space in Terms of Schemas and Linear Models." *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 139-143.

Young, R. M. and J. Whittington (1990). "Using a Knowledge Analysis to Predict Conceptual Errors in Text-Editor Usage." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 91-97.

Zanden, B. V. and B. A. Myers (1990). "Automatic, Look-and-Feel Independent Dialog Creation for Graphical User Interfaces." *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, 27-34.

Zhai, S., S. Conversy, M. Beaudouin-Lafon and Y. Guiard (2003). "Human on-line response to target expansion." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 177-184.

Zhai, S. and P.-O. Kristensson (2003). "Shorthand writing on stylus keyboard." *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, 97-104.

Zhai, S., A. Sue and J. Accot (2002). "Movement model, hits distribution and learning in virtual keyboarding." *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, 17-24.

Vita

Jonathan R. Howarth

Contact Information

jhowarth@vt.edu

http://csgrad.cs.vt.edu/~jhowarth

660 McBryde Hall (0106)
Virginia Tech
Blacksburg, VA 24061

Education

- Furman University, Greenville, SC (2002)
  - Bachelor of Science, Computer Science
  - Bachelor of Science, German Literature
- Virginia Tech, Blacksburg, VA (2004)
  - Master of Science, Computer Science and Applications

Employment

- Virginia Tech, Research Assistant (Spring 2004)
- Virginia Tech, Research Assistant (Summer 2003)
- Virginia Tech, Teaching Assistant (Fall 2002 – Fall 2003)
- Liberty Insurance Services, Programmer (Summer, 2001)