Supporting Novice Usability Practitioners with Usability Engineering Tools

Jonathan Randall Howarth

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

Doctor of Philosophy
in
Computer Science and Applications

Dr. Rex Hartson, Virginia Tech, Chair
Dr. Tonya Smith-Jackson, Virginia Tech
Dr. Manuel Pérez-Quiñones, Virginia Tech
Dr. Terence Andre, United States Air Force Academy
Dr. Andrea Kavanaugh, Virginia Tech

April 13, 2007
Blacksburg, Virginia

Keywords: Usability Engineering, Usability Engineering Tool Support, Usability Evaluation, Usability Problem Instances

Supporting Novice Usability Practitioners with Usability Engineering Tools

Jonathan Randall Howarth

Abstract

The usability of an application often plays an important role in determining its success. Accordingly, organizations that develop software have realized the need to integrate usability engineering into their development lifecycles. Although usability practitioners have successfully applied usability engineering processes to increase the usability of user-interaction designs, the literature suggests that usability practitioners experience a number of difficulties that negatively impact their effectiveness. These difficulties include identifying and recording critical usability data, understanding and relating usability data, and communicating usability information. These difficulties are particularly pronounced for novice usability practitioners.

With this dissertation, I explored approaches to address these difficulties through tool support for novice usability practitioners. Through an analysis of features provided by existing tools with respect to documented difficulties, I determined a set of desirable tool features including usability problem instance records, usability problem diagnosis, and a structured process for combining and associating usability problem data. I developed a usability engineering tool, the Data Collection, Analysis, and Reporting Tool (DCART), which contains these desirable tool features, and used it as a platform for studies of how these desirable features address the documented difficulties.

The results of the studies suggest that appropriate tool support can improve the effectiveness with which novice usability practitioners perform usability evaluations. More specifically, tool support for usability problem instance records helped novice usability practitioners more reliably identify and better describe instances of usability problems experienced by participants. Additionally, tool support for a structured process for combining and associating usability data helped novice usability practitioners create usability evaluation reports that were of higher quality as rated by usability practitioners and developers.

The results highlight key contributions of this dissertation, showing how tools can support usability practitioners. They demonstrate the value of a structured process for transforming raw usability data into usability information based on usability problem instances. Additionally, they show that appropriate tool support is a mechanism for further integrating usability engineering into the overall software development lifecycle; tool support addresses the documented need for more usability practitioners by helping novices perform more like experts.

Dedication

I dedicate this dissertation to my family. I love you Mom, Dad, and Liz. Thanks for all your support.

Additionally, I dedicate this dissertation to my friends who encouraged me and occasionally made me tasty home-cooked meals to help keep me going during late nights of work.

Acknowledgements

I would to thank the members of my committee for their time and effort: Dr. Rex Hartson, Dr. Tonya Smith-Jackson, Dr. Manuel Pérez-Quiñones, Dr. Terence Andre, and Dr. Andrea Kavanaugh. In particular, I would to thank Dr. Rex Hartson, my mentor and friend, for his input and guidance on all aspects of this work and for the countless hours that he spent “word smithing” this dissertation.

I owe a special thanks to Dr. Miranda Capra for the many conversations that I had with her concerning the design of my studies and the analysis and interpretation of the results.

I also owe a special thanks to Pardha Pyla for many in-depth talks about usability and the universe in general.

Additionally, I would like to thank Dr. Miranda Capra, Laurian Vega, Pardha Pyla, and Dr. Beth Yost for the work that they did helping me collect data for my studies. Each one contributed at least 20 hours of his or her time.

I would also like to thank the Air Force Office of Scientific Research for its support with funding for a Small Business Technology Transfer Topic grant and Dr. Randall Sparks of Knowledge Analysis Technologies with whom Dr. Rex Hartson and I collaborated on the grant. The ideas for this dissertation developed in part from work done on this grant.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

For several years, usability has no longer required justification in most quarters. Butler [1996] states that "usability has become a competitive necessity for the . . . success of software" (p. 59). Because of the growing awareness of its importance, organizations that produce software products have been expending resources for “doing usability”– building enviable usability laboratories, buying usability equipment, training developers in usability engineering (UE) methods [Hix & Hartson, 1993a], and conducting usability testing. These investments have helped to make UE an important part of the overall software development lifecycle. Accordingly, organizations want to maximize the effectiveness of their UE processes. The literature, however, suggests that usability practitioners experience a number of difficulties that negatively impact the effectiveness with which they are able to work, which in turn impacts the effectiveness of the UE process within which they work. These difficulties are particularly pronounced for novice usability practitioners. Determining how to address these difficulties represents an interesting research opportunity, the results of which can be applied to improve the effectiveness with which usability practitioners work.

1.2 Problem

The concept of an iterative, evaluation-centered life cycle process is now an established and proven approach to improving the usability of a user-interaction design. However, the typical UE process is not as effective as it has the potential to be in improving product usability through design iteration because of difficulties experienced by usability practitioners.

1.3 Background

1.3.1 Abstract Representation of Effectiveness

A key term in the problem statement (Section 1.2) is “effective”, which can have many meanings and interpretations. For the purposes of this dissertation, I use the following general definition provided by Sink [1985]: the degree to which a system accomplishes what it should accomplish. Sink provided this definition in the context of organization performance measurement, but it also is appropriate in terms of the contribution of the UE process in improving the usability of an interaction design.

Figure 1 is an abstract representation of two different processes (P and P’) that take one set of inputs (I) and produce two different sets of outputs (O and O’) with the objective of achieving some goal G. This general setup is similar in nature to the core organizational system described by Sink and Tuttle [1989], which transforms inputs into useful outputs. Assume that there is a function Effectiveness that takes as its parameters a set of inputs and a set of outputs and returns an effectiveness measurement with respect to G. If Effectiveness(I,O’) > Effectiveness(I,O), then P’ is defined as being more effective than P.

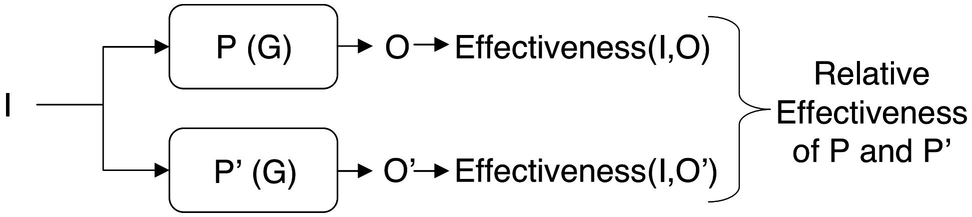


Figure 1: An abstract representation of the relative effectiveness of outputs (O and O’) produced for a given set of inputs (I) by two different processes (P and P’) with respect to some goal (G).

1.3.2 Usability Engineering Process

To put the abstraction of Section 1.3.1 in the context of UE, the process component (P) of Figure 1 is the UE process, and the goal (G) is increased usability for a software application. The usability of a software application is the effectiveness, efficiency, and satisfaction with which users of the application are able to achieve specific goals [ISO, 1998]. A number of different depictions of the UE process exist in the literature, but they all share the same basic sub-processes shown in Figure 2: systems analysis, design, implementation, and usability evaluation [Butler, 1996]. Each sub-process contains a role in italics, which refers to the person or people who participate in the sub-process. The UE process is iterative and can be continued until user and organizational requirements are satisfied or until budget or time limitations are reached.

Systems analysis involves understanding the needs of the users who will use the system and the context in which they will use it. Sources such as the ISO 13407 standard provide guidance on the types of information that should be collected during systems analysis [ISO, 1999]. Examples include the characteristics of users such as knowledge, skill, and experience; the tasks users will perform with the system and the information objects needed for these tasks; and required performance characteristics of the new system relative to any existing systems. On successive iterations, systems analysis includes comparing data obtained from evaluation with the user and organizational requirements to determine what to improve and how to improve it.

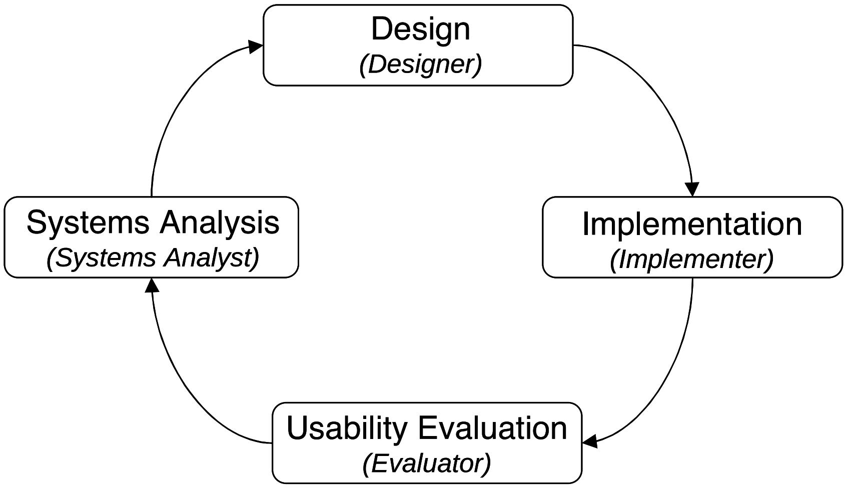


Figure 2: Usability engineering process

Design involves creating an interaction design of how a user communicates or interacts with an application that meets the requirements developed during systems analysis. The focus on interaction is important and is discussed in a variety of sources such as [Hix & Hartson, 1993a] and [Shneiderman, 1998]. This focus distinguishes design in UE from design in other disciplines, such as systems engineering, in which issues such as the system architecture design are given precedence. During the first iteration, an initial interaction design is developed, which is then modified on successive iterations.

The implementation sub-process involves constructing a prototype or real software product embodying the interaction design and then modifying the prototype and improving it on subsequent iterations. The eventual output of the implementation sub-process is a release quality system. Sources such as Mayhew [1999] describe an approach to prototyping that begins with low fidelity prototypes and progresses to more high fidelity prototypes and eventually to a release product. Low fidelity prototypes such as paper prototypes are useful in evaluating the interaction design of a system at an early stage. After eliminating major flaws, increasingly higher fidelity prototypes are useful in subsequent iterations to test usability specifications and user satisfaction.

During the usability evaluation sub-process, evaluators apply usability evaluation methods to gather data on the suitability of an interaction design as it is represented in a prototype. Usability evaluation methods include analytical methods such as heuristic evaluation [Nielsen & Molich, 1990] or cognitive walkthroughs [Polson *et al.*, 1992] and empirical methods such as lab-based usability testing [Hix & Hartson, 1993b]. These methods collect both subjective and objective data, which are analyzed and then recorded in usability problem (UP) reports. These UP descriptions are collected in a usability evaluation report, which is then used in the systems analysis sub-process to compare the current state of an interaction design to the needs of the intended users and organizational requirements and in the design sub-process to improve the interaction design.

The set of inputs (I) for the UE process is information inputs including rationale documenting the need for the new system, initial design ideas, and existing similar designs. The output (O) of the UE process is usually an interaction design meeting various usability goals (e.g. high customer satisfaction, safety for life-critical systems, learnability for new users) that will increase target users' performance and satisfaction levels. The interaction design may be embodied as a high fidelity prototype to be used as a proof of concept, a final product design, or recommendations for improving a number of related products.

For the purposes of this dissertation, information inputs are held constant by focusing on the usability evaluation sub-process (see Section 1.3.3). Additionally, we are assuming a fixed resource environment, meaning that people and time resources are relatively fixed for a given project.

1.3.3 Usability Evaluation Sub-Process

The UE process described in Section 1.3.2 contains four sub-processes; in principle, increasing the effectiveness of any of these will contribute to increasing the effectiveness of the overall process. The usability evaluation sub-process is an important part of the UE process because it generates the UP information that is used to make decisions in the rest of the process.

There are many sources that describe the usability evaluation sub-process using a variety of techniques and methods; for examples the reader is referred to [Hix & Hartson, 1993a, Rubin, 1994]. All usability evaluation sub-processes whether they use empirical or analytical techniques have three basic stages: usability data collection, UP analysis, and usability evaluation reporting. Figure 3 shows these stages. As in Figure 2, each stage contains a role in italics, which refers to the person or people participating in the stage. Each stage has a different role associated with it because a number of individuals may work together to complete the activities in the usability evaluation sub-process. In addition, text accompanying each of the connecting arrows describes what is produced by each of the stages.

The facilitator does usability testing (or any kind of usability data collection) during the usability data collection stage and produces raw usability data in the form of notes with associated video and audio clips, screen images, etc. The ultimate goal of the usability evaluation sub-process is to transform this data into usability information that can be used to improve an interaction design. Current approaches rely on the expertise of problem analysts to extract UPs from the raw data in the usability problem analysis stage. The extraction of UPs, however, is not straightforward, particularly for novices. Raw usability data is typically very specific and detailed while usability problems are necessarily general. I introduce the concept of UP instances to serve as a bridge between raw data and usability problems.

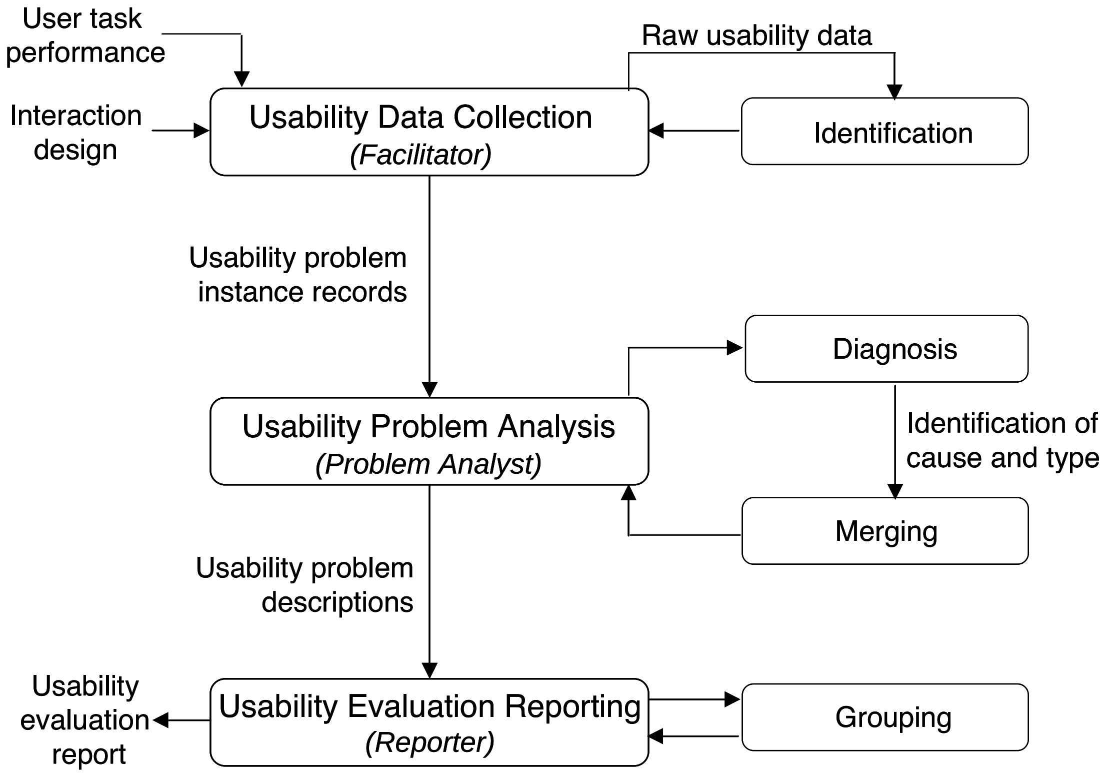

User task performance
Raw usability data
Interaction design
Usability Data Collection
(Facilitator)
Identification
Usability problem instance records
Diagnosis
Usability Problem Analysis
(Problem Analyst)
Identification of cause and type
Merging
Usability problem descriptions
Usability evaluation report
Usability Evaluation Reporting
(Reporter)
Grouping

Figure 3: Usability evaluation sub-process of the usability engineering process

Each occurrence of a UP as encountered by a participant and observed by the evaluator is a UP instance. The same UP may be experienced by multiple participants or multiple times by one participant. Figure 3 includes the identification of instances in the usability data collection stage. The facilitator produces brief UP instance records that contain just enough information to describe the UP instance.

During the UP analysis stage, the problem analyst fills in the UP instance records from the usability data collection stage with more details as necessary. The problem analyst then diagnoses and merges the UP instance records. Diagnosis provides a clear, complete, and unambiguous statement of the design flaw associated with each UP instance. Diagnosis also normalizes UP instances for comparison and evaluation (e.g., to determine if two seemingly different UP instances are actually about the same UP). Merging involves combining UP instances that map to the same UP. A UP necessarily has the same diagnosis or a generalized version of the diagnosis of the merged UP instances that it represents. A UP describes the effect that an interaction design flaw has on the user; UPs are documented with UP descriptions.

The reporter in the usability evaluation reporting stage uses the UP descriptions generated during the UP analysis stage to generate usability evaluation reports to guide subsequent fixing in the design stage of the UE process. Grouping involves associating UPs in a manner that is most appropriate for the target audience of the usability evaluation report. For example, implementers may want to know specific areas of an interface that are involved in a UP while managers may want an executive summary of an interaction design's strengths and weaknesses.

1.3.4 Formative Usability Evaluations

During the usability evaluation sub-process, usability practitioners primarily conduct formative usability evaluations. As described in a Usability Professional's Association workshop report [Theofanos *et al.*, 2005], formative usability evaluations are conducted to "guide the improvement in design of future iterations" (p. 3). Usability practitioners conduct formative usability evaluations to understand the strengths and weaknesses of a given interaction design. During formative evaluations, usability practitioners collect a variety of qualitative data such as verbal protocol and subjective ratings with the goal of producing UP descriptions.

Summative usability studies represent a different type of usability evaluation that is typically performed after a product is released. Summative usability studies provide proof in the form of statistical significance that one given interaction design is better than other designs in specific ways. Usability practitioners may collect quantitative data such as measures of time on task and error counts that they later use in metrics.

Summative evaluations certainly have their value, but formative evaluations are the focus of this dissertation because of the emphasis on usability practitioners' abilities to understand and critique the usability of an interaction design. Improving these abilities will help to increase the effectiveness with which they work in the usability evaluation sub-process.

1.3.5 Usability Engineering Tool Support

There are a number of research issues associated with conducting formative usability evaluations in the usability evaluation sub-process. For example, much work in the 1990's focused on understanding the relative strengths and weaknesses of usability evaluation methods. More recent work has focused on understanding and fixing the UPs identified through usability evaluation methods. The focus of this dissertation is the use of software tools to support usability practitioners during formative usability evaluations. Appropriate tool support can provide a number of benefits including helping usability practitioners collect and analyze usability data and report usability problems in a structured and efficient manner.

1.3.6 Usability Practitioner Skill

The literature suggests that skill plays an important part in usability evaluation. For example, a study by Nielsen [1992] found that usability specialists were better than non-specialists at using heuristic evaluation to evaluate an interface. Also, in a study comparing the iterative development of designs by human factors specialists and programmers, Bailey [1993] concludes that "the training and background of designers can have a large effect on user interface design" (p. 204).

The focus of this dissertation is the use of usability engineering tools to support novice usability practitioners. I chose novice usability practitioners as the target audience because they can benefit most from appropriate tool support. Experts typically have developed methods and strategies that work for them; although they may benefit from tool support, they do not require it. Novice practitioners, on the other, may fail to recognize important usability data or interpret data incorrectly without the guidance and support that can be provided by a usability engineering tool.

1.4 Scope

Sections 1.3.1 to 1.3.5 provide background that scopes my work. This dissertation is limited to the usability evaluation sub-process of the overall UE process. Within the usability evaluation sub-process, the focus is on formative usability evaluations that have the goal of producing usability evaluation reports. Within the context of formative usability evaluations, the focus is on tool support for UE, in particular the effectiveness of such tool support for novice usability practitioners.

1.5 Research Goals, Research Questions, and Approach

Table 1 shows the research goals, research questions, and steps of the approach. Table 2 maps steps to mechanisms and principle outputs.

Table 1: Goals, questions, steps of the approach mapped to phases

Research Goal	Research Questions	Approach
RG1 - Investigate difficulties experienced by usability practitioners and how these difficulties are addressed (or not) by state-of-the-art UE tools.	**RQ1a** - What difficulties do usability practitioners experience when they perform usability evaluations? **RQ1b** - What features do state-of-the-art tools provide (or not) to address these difficulties?	**Step 1** - Review the literature to develop an understanding of difficulties that usability practitioners encounter during formative usability evaluations. This review synthesizes the anecdotal evidence that supports my statement of the problem. **Step 2** - Review existing UE tools to reveal features that are used to support usability practitioners. While the first step explores the problem space, this step explores the solution space as it is embodied in state-of-the-art tools.
RG2 - Develop a set of desirable features for UE tools targeted at difficulties that are either unaddressed or poorly addressed by existing state-of-the-art tools.	**RQ2a** - What difficulties are not adequately addressed by features of existing state-of-the-art tools? **RQ2b** - What are some desirable tool features that target difficulties that are unaddressed or poorly addressed by existing state-of-the-art tools?	**Step 3** - Analyze the features identified in Step 2 in terms of how well they address the difficulties identified in Step 1. The analysis yields a set of desirable features for a UE tool. **Step 4** - Develop specific instances of the desirable features identified in Step 3. **Step 5** - Design and implement a tool that includes the specific instances of the desirable features developed in Step 4.
RG3 - Evaluate these desirable features with respect to how they affect the effectiveness of novice evaluators.	**RQ3a** - How does tool support for UP instance records affect the effectiveness of novice evaluators? **RQ3b** - How does tool support for diagnosis affect the effectiveness of novice evaluators? **RQ3c** - How does tool support for merging UP instances and grouping UPs affect the effectiveness of novice evaluators?	**Step 6** - Use the tool developed in Step 5 to study each of the desirable features individually to determine how well each addresses the difficulties identified in Step 1 for novice usability practitioners.

Table 2: Research mechanisms, outputs, and completion dates by phase

Step	Mechanism(s)	Principle Output(s)
1	• Literature review • Analysis	• Synthesis of difficulties experienced by usability practitioners
2	• Literature review • Analysis • Tool testing	• Categorization of existing UE tools • Identification of features in existing UE tools
3	• Analysis	• Abstract descriptions of desirable features
4	• Analysis	• Descriptions of specific instances of desirable features
5	• Design • Implementation • Evaluation (walkthroughs, formative testing, and field testing)	• The Data Collection, Analysis, and Reporting Tool (DCART)
6	• Study 1 • Study 2 • Study 3	• Measures of novice evaluator effectiveness in usability evaluations • Method based on UP instances • Data concerning tradeoffs associated with desirable features • Analysis of data • Discussion of tradeoffs

1.6 Contribution of Research

With this research, I make two primary contributions. The first contribution is to the academic field. I provide a synthesis of difficulties encountered by usability practitioners during the evaluation sub-process. These difficulties are documented in the literature, but no previous research has related them with a focus on how they can be moderated with tool support. I also provide a comparison of approaches and concepts implicit in existing UE tools. There are informal comparisons of small numbers of tools in the research literature and in trade publications for practitioners and consumers. My work differs in that I review a large number of both commercial and academic tools and introduce a categorization scheme for organizing and relating them. Another academic contribution is a formal method for analyzing how tools support usability practitioners. Specifically, I define the concept of effectiveness for evaluators in usability evaluations and develop quantitative measures of effectiveness. My approach to measuring effectiveness is novel in that it is based on UP instances; all other research efforts have been based on UPs. UP instances allow for finer granularity and more precise measurements. Although I have applied the method to evaluating UE tools, it would also be appropriate for researching other aspects of UE such as comparisons of usability evaluation methods.

The second contribution of my work is more applied. My review of existing UE tools has benefits for the academic field in that I describe concepts implicit in these tools, but it also has practical value in that I identify leading state-of-the-art tools and describe their features. Additionally, this dissertation suggests needed features that can be incorporated in commercial tools. Also, my work addresses Nielsen's concern about how the usability engineering will scale up to impact more interaction designs in more products [2005]. The desirable features in this dissertation can help novice usability practitioners produce usability evaluation reports of better quality thereby helping to "expand usability beyond the usability professionals" (p. 3).

2 Related Work

2.1 Difficulties Experienced by Usability Practitioners

The UE process described in Section 1.3.2 is known to be successful in improving software usability. For example, Szcuzur [1994] describes a UE process that was effective for improving the usability of an application used at the Goddard Space Flight Center. Another example is a case study by Hertzum [1999], which successfully employed a UE process that included both formal laboratory tests and informal workshop tests. Literature also exists that demonstrates the cost benefits of a UE process and provides methods for calculating a UE process' contribution [Bias & Mayhew, 1994, Lund, 1997]. However, difficulties experienced by usability practitioners that are documented in the literature indicate that the UE process, especially the usability evaluation sub-process, is not as effective as it could be.

2.1.1 Evaluator Effect

The evaluator effect is the tendency of usability practitioners with differing knowledge and experience to find different types and numbers of UPs during usability evaluation. Work by Rowe et al. [1994] demonstrated that different usability evaluation teams studying the same interface will find different issues. Jacobsen et al. [1998] documented and named the evaluator effect in a study in which four usability experts were given the same video tapes of four participants performing tasks in a multimedia authoring system. Each expert identified about half of the UPs, but about half of those were unique to the individual expert. A related study by Hertzum and Jacobsen [2003] provided more evidence of the evaluator effect by reviewing 11 studies that used one of the following usability evaluation methods: cognitive walkthroughs, heuristic evaluation, or thinking-aloud study. The authors proposed that the evaluator effect occurs because usability evaluation involves interpretation and that usability evaluation methods do not provide the guidance that usability practitioners need to perform reliable evaluations. Vermeeren et al. [2003] conducted a study that found evidence of the evaluator effect in different domains and also proposed reasons for the evaluator effect related to interpretation, such as guessing user intentions.

2.1.2 Content of Usability Problem Descriptions

UP descriptions document interaction design flaws that cause UPs for users. They are used, in the context of a usability evaluation report, to help system analysts and designers identify specific features of an interaction design to change, add, or remove in subsequent iterations.

There have been a limited number of UP description formats documented in the literature. Jeffries developed recommendations for what to include in a UP description while performing a review of UP descriptions to determine their shortcomings [Jeffries, 1994]. While the recommendations represent an improvement over ad hoc reporting, they do not provide a definite format and focus on solutions without addressing causes. A study by John and Packer [1995] on the learnability and applicability of the cognitive walkthrough method contained a UP description form with a unique reference number and fields for describing the UP, estimating its severity, and assessing the source of its discovery. This form, much like Jeffries' recommendations, did not specifically address the causes within the interaction design of problems. In a study comparing empirical testing with usability inspections, Mack and Montaniz [1994] describe a UP description structure that includes descriptions of goal-directed behavior, interface interactions, possible causes, and severity. This report structure does address the causes of UPs, but it relies heavily on interpretation and is subject to the difficulties discussed in Section 2.1.1.

To enable comparative studies of usability evaluation methods, a more standard way to describe UPs was needed. Lavery et al. [1997] developed a structured UP description format that addressed the shortcomings of previous UP description formats. The method captures the problem context, cause, outcomes, and solutions. Cockton and Lavery [1999] leverage this structured UP description format in the Structured Usability Problem Extraction (SUPEX) framework, which separates problem context, cause, and recommendations. The SUPEX framework provides a rigorous approach to extracting problems that distinguishes among multiple levels of abstraction and handles relationships among user actions to reduce under- and over-reporting of UPs. While SUPEX is thorough, its application represents an investment in terms of time and effort that is too large to be practical for use in non-academic settings. The authors describe modifications to the framework that would reduce the time and effort requirements, but these, as would be expected, negatively affect the quality of the results. In addition, later work by Cockton et al. [2003] supports the use of structured UP descriptions by demonstrating that they help to improve analysts' performance with the heuristic evaluation method.

More recently Capra [2006] developed guidelines for the content of UP descriptions through a series of three studies with usability practitioners. These guidelines represent an important step towards structuring the content of UP descriptions. They, however, are subject to a major limitation of guidelines in that they may be difficult to apply consistently [Borges *et al.*, 1996, Smith, 1986].

In sum, the literature does not provide a clear answer as to what to include in a UP description. As a result, UP descriptions are often ad hoc [Andre *et al.*, 2001]. Without a specific format, usability practitioners may not be aware of what to look for during usability data collection or what to clarify with participants during empirical testing. In addition, even if necessary usability data are observed and recorded during usability data collection, the lack of a consistent report format may make it difficult for problem analysts to understand and relate the data.

2.1.3 Content of Usability Evaluation Reports

As illustrated in Figure 3, the output of the usability evaluation sub-process is a usability evaluation report. The UP descriptions discussed in Section 2.1.2 differ from usability evaluation reports; the former is used to document an individual UP while the latter is used to convey results of an entire usability evaluation. In a paper concerning redesign proposals, Hornbæk & Frøkjær [2005] show the need for usability evaluation reports that summarize and convey usability information by discussing how lists of UP descriptions, by themselves, have limited use in practical contexts. Usability evaluation reports are used throughout the subsequent iterations of the UE process to make decisions about what UPs to fix and how to fix them. It is therefore important that these usability evaluation reports contain information in a format that is useful to other individuals involved in the usability engineering process. Hornbæk & Stage [2006], for example, identify providing feedback from usability evaluation to design as a challenge for usability research.

As discussed in Section 2.1.2, the challenge associated with UP descriptions is getting the necessary data to completely specify the UP. The challenge associated with usability evaluation reports is conveying the necessary information associated with the UP descriptions to a given audience. Nayak et al. [1995] discuss some of the difficulties of conveying usability information, such as explaining observation-based data and understanding the needs of the target audience. Dumas et al. [2004] provide further evidence that practitioners have difficulty writing effective usability evaluation reports in a study of reports generated for the fourth Comparative Usability Evaluation. The authors demonstrate that practitioners who write reports often emphasize the negative aspects of a design, express annoyance, use usability jargon, and are not specific with respect to UPs and how to fix them.

The information included in a usability evaluation report depends on the purpose of the report and the intended audience. For example, if the report is being produced for management, the focus may be on what UPs can be fixed within the number of people hours allocated in the budget. In such cases, a cost-importance analysis that prioritizes UPs based on a ratio of estimated cost to perceived importance may be a key element [Hix & Hartson, 1993a]. A report for implementers, however, might present UPs with appropriate solutions as they are related to the software modules or components.

Theofanus [2005] lists several potential elements of a usability evaluation report including an executive summary, a description of participants, a description of tasks and scenarios, and a collection of UP descriptions. There have been few papers, however, addressing the best format for usability evaluation reports. In 1997, the Industry Usability Reporting Project initiated by the National Institute of Standards and Technology developed the Common Industry Format (CIF), which is currently the most well known format for usability evaluation reports. The CIF became an American National Standard for Information Technology Standard in 2001 [ANSI, 2001]. By standardizing the reporting of usability tests, the CIF hoped to encourage the consideration of usability in purchasing software products; customer organizations that were interested could evaluate different products based on their CIF reports. The CIF includes sections for describing the product, the method used to evaluate the product, and the results of the evaluation. The CIF is intended for summative usability evaluations, but usability practitioners most frequently perform formative usability evaluations. Theofanos [2005] and Theofanus and Quesenbery [2005] describe efforts to develop a new CIF that would provide practitioners with guidance for performing and reporting formative studies.

The usability evaluation report consolidates usability information and provides the context for understanding UP descriptions. Without this context, UP descriptions may be misunderstood or overlooked.

2.2 Existing Usability Engineering Tools

There are a number of tools for use in UE efforts that represent a variety of focuses and development activities. I present a survey of these tools using a categorization scheme to structure the discussion of the state of these tools. I include tools that I found through a combination of a literature and a web search. The list of tools is not exhaustive; instead it provides examples of each basic category of tool.

2.2.1 Tools not Included in the Survey

The following basic types were excluded from the survey: custom tools, tools that facilitate the construction of interfaces, and tools with a business or social research focus.

Custom tools are created by an organization specifically to fit the needs of a particular usability process. As a result, these tools are generally not documented and not made available for use outside of the organization. The case studies of usability efforts in commercial organizations (for examples, see [Hertzum, 1999, Szczur, 1994]) discuss the processes used to perform usability testing and the resulting impact but not the tools used to support the processes. One exception is a panel discussion about in-house usability tools that included representatives from several major companies as panel members [Weiler, 1993]. This panel discussion provided general information about custom tools used in companies such as Microsoft and Apple, but there is no practical way for me to evaluate these tools. In addition, these tools are highly specialized and may be difficult to use in different contexts or may not be adaptable to different processes.

Two basic types of tools are used to facilitate the construction of interfaces. One type is tools that help programmers write the code for graphical user interfaces. Throughout the 1980’s and into the early 1990’s user interface management systems received a considerable amount of attention (see [Olsen, 1992, Olsen *et al.*, 1985, Olsen *et al.*, 1987] for examples and issues). User interface management systems have been replaced by integrated development environments that have GUI layout capabilities. Another type of tools is used to help with creating interfaces quickly for prototyping. One example is Ludi's tutorial on using Macromedia Director as a rapid prototyping tool [2000]. There are also a number of tools that exist online for prototyping. For example, there is a look and feel for Java that simulates sketching an interface on a napkin to give interfaces a more informal feel [Arnold, 2005]. Both types of tools are excluded because of the focus on the usability evaluation sub-process instead of the design sub-process.

Some tools are specifically created for UE processes, but do not address usability evaluation activities in any detail. For example, Agility helps individuals involved in the UE process collaborate with one another and plan activities and deliverables [Classic System Solutions Inc., 2005]. The tool, however, has a business focus and is not appropriate for this survey. Other tools are capable of logging data and could be used for UE, but are tailored for other types of research. One example is Observer, which is primarily intended for collecting observational data for social research [Noldus, 2005].

2.2.2 Categorization Scheme

My categorization scheme is based on a taxonomy of usability evaluation tools developed by Ivory and Hearst [Ivory & Hearst, 2001] and the stages of the usability evaluation sub-process shown in Figure 3. The three levels to this scheme are as follows:

- Evaluation method class – How usability evaluations are conducted using the tool
    - Analytical – Evaluations involve inspections by experts such as heuristic evaluations or cognitive walkthroughs or static analysis of an interaction design.
    - Empirical – Evaluations involve observing a participant using a tool.
- Application class – What type of application can be evaluated using the tool
    - Desktop – The tool can only be used to evaluate desktop applications.
    - Web – The tool can only be used to evaluate websites.
    - Both – The tool can evaluate both desktop applications and websites.

- Supported stages of the usability evaluation sub-process – Which stages of the usability evaluation sub-process are supported by the tool (Section 1.3.3)
    - Usability data collection
    - UP analysis
    - Usability evaluation reporting

2.2.3 Tools Included in the Survey

The tools included in the survey along with their evaluation method class, application class, and supported stages of the usability evaluation sub-process are shown in Table 3. Tools are discussed in subsequent sections based on their evaluation methods, application classes, and supported stages.

Table 3: Tools included in the survey

Tool Name	Evaluation Method Class	Application Class	Supported Stages
A tool for computing the complexity of dialog boxes [Parush *et al.*, 1998]	Analytical	Desktop	UP analysis
KRI/AG [Lowgren & Nordqvist, 1992]	Analytical	Desktop	UP analysis
semi-Automated Interface Designer and Evaluator (AIDE) [Sears, 1995]	Analytical	Desktop	UP analysis
SHERLOCK [Mahajan & Shneiderman, 1997]	Analytical	Desktop	UP analysis
LIFT [Usable Net, 2005]	Analytical	Web	UP analysis Usability evaluation reporting
NIST Webmetrics – Static Analyzer Tool (WebSAT) [Scholtz & Laskowski, 1998]	Analytical	Web	UP analysis
Web page critiquing tool [Faraday, 2000]	Analytical	Web	UP analysis

Integrated Data Capture and Analysis Tool (IDCAT) [Hammontree *et al.*, 1992]	Empirical	Desktop	Usability data collection UP analysis
User Action Graphic Effort (UsAGE) [Uehling & Wolf, 1995]	Empirical	Desktop	Usability data collection UP analysis
NIST Webmetrics – Category Analysis Tool (WebCAT) [Scholtz & Laskowski, 1998]	Empirical	Web	Usability data collection
NIST Webmetrics – Visual Instrumenter Program (WebVIP) [Scholtz & Laskowski, 1998]	Empirical	Web	Usability data collection
Usability Testing Environment (UTE) [Mind Design Systems, 2005]	Empirical	Web	Usability data collection UP analysis
Usability Testing Suite [Uzilla, 2005]	Empirical	Web	Usability data collection UP analysis
Usability Testing Tool [Working Web, 2005]	Empirical	Web	Usability data collection UP analysis Usability evaluation reporting
Web Event-logging Tool (WET) [Etgen & Cantor, 1999]	Empirical	Web	Usability data collection
Diagnostic Recorder for Usability Measurement (DRUM) [Macleod & Rengger, 1993]	Empirical	Both	Usability data collection UP analysis Usability evaluation reporting
Morae [TechSmith, 2005]	Empirical	Both	Usability data collection UP analysis Usability evaluation reporting
Ovo Logger [Ovo Studios, 2005]	Empirical	Both	Usability data collection UP analysis Usability evaluation reporting
Spectator [Biobserve, 2005]	Empirical	Both	Usability data collection UP analysis Usability evaluation reporting

Usability Activity Log [Bit Debris Solutions, 2005]	Empirical	Both	Usability data collection UP analysis
Visual Mark [Users First, 2005]	Empirical	Both	Usability data collection Usability evaluation reporting

Tools in the table are sorted first by method class and then by application class. Thereafter, they appear in alphabetical order.

2.2.3.1 Analytical, Desktop

The tools in this category provide usability practitioners with a way to statically evaluate interface designs. As a result, these tools only support the UP analysis stage of the usability evaluation sub-process.

Parush et al. [1998] developed a tool for computing the complexity of dialog boxes that analyzes a given dialog based on the screen factors of element size, location density, alignment, and grouping to produce a complexity score for a dialog. The authors performed a study with the tool and found that poor alignment and local density have effects on search time, and alignment and grouping affected participants' subjective ratings of dialogs. AIDE uses a different set of metrics (efficiency, alignment, horizontal balance, vertical balance, and constraints) to help usability practitioners analyze layouts for dialogs [Sears, 1995]. AIDE differs from the tool developed by Parush et al. in that it allows designers to interactively develop a design.

As described in Section 2.1.2, guidelines are often difficult to apply. The goal of the development of KRI/AG was to make general interface design knowledge more accessible to designers [Lowgren & Nordqvist, 1992]. KRI/AG represents the knowledge contained in guidelines, such as those by Smith and Mosier [1986], and style guides, such as the Motif Style Guide [Open Software Foundation, 1991], in a series of rules. A representation of a design is passed to KRI/AG, and it produces critiques based on these rules.

SHERLOCK, the final tool in this section facilitates evaluating the task-independent aspects of consistency of software applications, including layout, visual design, and terminology [Mahajan & Shneiderman, 1997]. Usability practitioners convert an interface description to a standard format and then submit it to the SHERLOCK suite for processing.

2.2.3.2 Analytical, Web

NIST Webmetrics WebSAT [Scholtz & Laskowski, 1998] and LIFT [Usable Net, 2005] check static html code for violations of web specifications such as Section 508 and W3C-WCAG priority 1 and 2. WebSAT supports only UP analysis, but

LIFT allows usability practitioners to generate a number of different reports, from very detailed reports on coding issues for developers to more general executive-level summaries for managers.

A web page critiquing tool by Faraday [2000] is based on the empirical results from eye tracking studies that determine how a user will search and scan a web page. The tool helps designers determine how users will view their webpage and allows them to change the layout of the page if important objects will be missed.

2.2.3.3 Empirical, Desktop

The tools in this section take two different approaches to collecting usability data and analyzing it. UsAGE is based on a specific user interface management system and has the ability to record user interface actions [Uehling & Wolf, 1995]. UsAGE is used to record both expert and novice interactions with the user interface. The novices' interface actions are compared to those of the expert using an action graph. The expert's actions are shown as a linear path, while the novices' are shown as deviations from the path.

The Event Capture component of IDCAT works at the system level to capture system events and write them to log files [Hammontree et al., 1992]. The Event Filters component allows the usability practitioner to filter the log files by various criteria such as object type and event type. Usability practitioners can then use the Multimedia Data Analyzer component to automatically scroll to a specific point in video captured during the usability evaluation and add additional comments to the video using the Retrospective Verbal Protocol Recorder.

2.2.3.4 Empirical, Web

NIST Webmetrics WebCAT is unique in this category in terms of the type of data it collects [Scholtz & Laskowski, 1998]. It enables card sorting of content in a website; participants group objects in the interface and provide names for the groups. The usability practitioner can then use this information to create categories of content for a website.

NIST Webmetrics WebVIP [Scholtz & Laskowski, 1998] and WET [Etgen & Cantor, 1999] provide for basic collection of usability data. WebVIP allows usability practitioners to instrument a website, so that they are able to log link clicks and times. A major drawback, however, is that the tool requires the usability practitioner to create a copy of the website, which can be difficult with dynamic websites, and add code to the underlying html. WET logs events in web browsers such as page loads, button clicks, or link clicks. WET only requires the addition of a single call on each page to a Javascript file.

UTE [Mind Design Systems, 2005], the Usability Testing Suite [Uzilla, 2005], and the Usability Testing Tool [Working Web, 2005] are similar tools. Each allows usability practitioners to create scenarios for participants and capture data such as the number and order of pages visited, the time spent on each page, and page load times. Each tool also provides facilities for analyzing the collected data and computing metrics such as average completion time over a group of participants. The Usability Testing Tool also generates html reports with appropriate graphs of the collected data.

2.2.3.5 Empirical, Both

The tools in this category are all commercial tools. With the exception of DRUM [Macleod & Rengger, 1993], these tools represent the state of the art for UE tool support. Also, the tools provide support for all stages of the usability evaluation sub-process with the exception of the Usability Activity Log [Bit Debris Solutions, 2005], which does not provide usability evaluation reporting functionality, and Visual Mark [Users First, 2005], which does not provide UP analysis functionality.

DRUM was the earliest of all tools in this category. It is no longer available for use, but many of its ideas and features are present in newer tools. The DRUM Recording Logger allows usability practitioners to perform real-time and retrospective logging. It also includes the ability to control video-recorders by allowing usability practitioners to jump to the correct place in a video for a given event. The DRUM Scheme Manager allows usability practitioners to create event types and organize them into hierarchies that represent task-analytic schemes. Usability practitioners can use the DRUM Log Processor to generate metrics based on the MuSIC Performance Measurement Method from log data [Macleod *et al.*, 1997]; example metrics include the amount of time participants spend having problems and the efficiency with which participants can accomplish a task. The DRUM system provides support for managing the usability evaluation process that goes beyond UP analysis support. In particular, the DRUM Evaluation Manager helps usability practitioners manage the log files and metric data generated by the other DRUM components. This ability helps usability practitioners perform a meta-analysis that could extend beyond the efforts a single evaluation.

Of the newer tools, the Usability Activity Log is the most basic. It provides functionality for logging usability data and synching that data with a video source. The tool itself does not record any video streams, but it does timestamp comments made by usability practitioners during a session with a participant. UP analysis support is provided primarily through the ability to sort and search entries in the log file.

Morae is one of the most popular new tools in this category [TechSmith, 2005]. Morae Recorder runs on the participant's machine and collects keystrokes, mouse clicks, system events, audio, video, and screen capture during usability evaluations. Morae Remote Viewer allows usability practitioners to add their own tags with comments to mark the beginning and ending of tasks and to note critical incidents. Because Morae Recorder collects such a variety of data during sessions with participants, Morae Manager has the ability to compute a large number of metrics such as time metrics or activity metrics. In addition, Morae Manager gives the usability practitioner the ability to navigate and filter the data to isolate particular incidents or phenomena. Morae Manager allows usability practitioners to generate videos of sessions or particular segments of sessions. These videos can be tailored for presentation to different audiences, such as the marketing department or the design team. While Morae provides usability evaluation reporting capabilities in the form of highlight videos, it does not provide meta-analysis functionality such as that provided by the DRUM evaluation manager.

Visual Mark was developed as an alternative to Morae, particularly for users who need to work with platforms other than Windows. Visual Mark does capture up to four video streams and log and timestamp annotations made by the usability practitioner, but it does not capture any other data such as keystrokes or system events. At the conclusion of a session with a participant, Visual Mark automatically generates an html report that includes annotations and associated links to the video file. This report is useful in that it provides a quick and automated way to document a session. Visual Mark, however, does not provide any functionality for further UP analysis and recommends the use of third party tools for sorting log files and editing the video file.

Spectator helps usability practitioners structure usability data collection; more specifically, it provides project databases with up to five levels (ex. projects and subprojects), participant databases, task lists, behavior lists, and session scheduling tools [Biobserve, 2005]. Spectator does not capture low-level data such as keystrokes like Morae, but it can work with screen video captured by a hardware digital recording device marketed by Biobserve. Also like DRUM, Spectator allows usability practitioners to pool and analyze data from sessions with different participants.

The Ovo Logger is a fairly advanced observational logging tool [Ovo Studios, 2005]. Like DRUM and Spectator, Ovo Logger helps usability practitioners structure their usability data collection by providing functionality for creating and managing test scenarios and scheduling participants. The Ovo Logger itself is freeware, but additional add-ons must be purchased a la carte or in a package to enable the creation and administration of web surveys, video and screen capture, capture of keystrokes and mouse clicks in web applications, and remote viewing. Ovo Logger provides usability practitioners with timeline and grid views of data with searching and filtering functions that are similar to Morae. Ovo Logger also allows usability practitioners to compute the standard array of metrics provided in other tools in this category. OVO Logger is unique in that it provides a report writer that helps usability practitioners generate html reports in a CIF format [ANSI, 2001].

2.3 User Action Framework

This section provides an overview of the UAF and related work. The section begins with an introduction to the Interaction Cycle and an explanation of how it is used in the UAF, which is followed by a discussion of Norman's [1986] seven-stage theory of action model, the basis for the Interaction Cycle, and other research that has incorporated it.

2.3.1 The Interaction Cycle and the User Action Framework

The Interaction Cycle consists of the stages of Planning, Translation, Physical Actions, Outcome and System Functionality, and Assessment. These stages, which are also the major categories of the UAF, demonstrate the role of interaction design in supporting the cognitive, physical, and sensory actions of computer users. The Interaction Cycle is shown in Figure 4 as a circle to indicate the cyclical nature of a human's typical interaction with a computer or any kind of machine. The different sizes of the stages in the figure indicate an approximation of the relative magnitude of usability challenges (difficulties for the designer or user, number of UPs typically found) in each stage; the Translation stage clearly poses the most challenges, helping users determine which action to perform on which object in carrying out a task step. The Outcome and System Functionality stage is separate from the circle because it is concerned with actions performed by a computer (machine, in general) and involves no interaction with the human using the computer and, therefore, contains no interaction design issues.

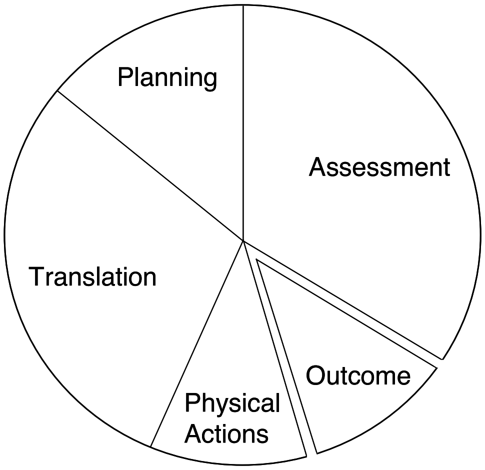


Figure 4: Interaction Cycle

Planning involves the cognitive processes of users as they decide what to do and what the system can help them 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 what action to make on what user interface object. Norman refers to such features as perceived affordances and Hartson [2003] refers to them as cognitive affordances.

The Physical Actions stage is where the user performs physical actions (e.g., clicking on an interface object). Real affordances in Norman's terms, or physical affordances in Hartson's terms, provide users with a physical feature to which the user applies the physical action to physically manipulate an object in the interface (e.g., an "active" area on a button, sensitive to clicking). The Physical Actions 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 performs cognitive actions to determine whether or not the response corresponds to the desired outcome, i.e. whether the outcome matches the goals set in Planning.

2.3.2 Norman's Model

The basis for the Interaction Cycle is a model of action proposed by Norman [1986] for a human's interaction with any type of machine. 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 the work of Lim et al. [1996] with Norman's model as a basis for determining why and in what context direct manipulation is superior to other types of interfaces. 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 et al. [1997] describe a modification of the cognitive walkthrough based on Norman's model. 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 et al. [1999] describe another application of Norman's theory. 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.

2.3.3 Evaluations of the UAF

The UAF has been evaluated in two major studies. The first study conducted by Andre et al. [2001] tested the reliability of the UAF. The second study by Andre et al. [2002] compared the UP Inspector, an inspection tool interface to the UAF, with heuristics and cognitive walkthroughs.

The goal of the reliability study was to document the level of agreement among professional usability 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 UP Taxonomy, an earlier diagnosis structure that was helpful in the creation of the UAF [Keenan, 1996, Keenan *et al.*, 1999].

In the study, 10 usability professionals with brief training on the UAF structure diagnosed 20 UPs using the UAF. 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 category, and overall. The UAF showed very strong agreement for all measures. For comparison with the UP 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 UP Taxonomy's score of .403.

The second study focused on comparing the following usability inspection methods: the UP Inspector, heuristics, and cognitive walkthroughs. The UAF serves as the theoretical base for the UP Inspector. 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 et al. [2001]. Thoroughness is the ratio of real UPs identified by the usability evaluation method over the base set of UPs, validity is the ratio of the base set of UPs over the identified UPs, and effectiveness is the product of thoroughness and validity. The authors used an address book application and developed a base set of UPs 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 UP Inspector 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 UPs that were not in the base set yielding low validity measures. The UP Inspector and the cognitive walkthrough performed equally. The authors conclude that the UP Inspector will perform better in the long term because it provides more detailed UP information.

2.4 Determining the Need for Micro-Iteration

The process of diagnosis with the UAF involves associating a UP with a path of UAF nodes that completely describes the problem type and its causes. Diagnosis with the UAF can be time consuming, and it is not practical to try to diagnose UPs during a session with a participant. However, if the necessary information needed for diagnosis is not captured during a session with a participant, complete diagnosis may not be possible later in the UP analysis stage.

I present the results of exploratory studies and analogies to medical diagnosis that helped me determine the need for micro-iteration, a process by which necessary information can be identified and recorded during usability data collection. This work is documented in [Howarth, 2006]. The first sub-section describes exploratory studies that I originally performed to determine how well new and intermediate users of the UAF could diagnose UPs. While analyzing the results, however, I discovered that many of the UP descriptions used in the study did not contain the data that analyst subjects needed to accurately diagnose UPs. I then looked to other fields to find analogies for how to perform diagnosis. The medical field provided an excellent analogy in terms of how doctors diagnose patients.

2.4.1 Exploratory Studies

I conducted exploratory studies to help me understand how problem analysts perform diagnosis with the UAF. The first study focused on the performance of analyst subjects who were new users of the UAF, and the second study utilized verbal protocol to help me better understand the diagnosis process used by analyst subjects who were intermediate users of the UAF.

The first study was intended to get an indication of how well analyst subjects who were new users of the UAF could use it to diagnose UPs and what I could do to improve the accuracy of diagnoses. The study involved 25 graduate UE students who were new users of the UAF. The students used the UAF to diagnose the UPs described in 20 UP descriptions based on a usability inspection of a kiosk ticket system that they had used in class. The students had two weeks to complete the diagnoses and did it in a time and place of their choosing.

I gave each student a unique username and password pair for the UAF Problem Reporting Tool (PRT), a web-based tool, and told them to use it to report their answers. I entered the 20 UP descriptions in the PRT as exercise originals. Each exercise original contained a UP description and an expert diagnosis path within the UAF. 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 his own instance of an exercise original, he would be presented with a form that contained the UP description and an empty text box for the diagnosis. The student would then use the UAF Viewer, a web-based tool that allows for navigation of the UAF, to find the most appropriate path through the UAF for the UP described in the UP description and paste that path into the form. The student could revise each diagnosis as much as he liked until he confirmed it. Upon confirmation, the system would present the student with both his 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.

I 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, I considered the students' diagnoses to match the experts' if they had the same top three levels. I also gave credit for a match if students described how flaws in the UP description led to a misdiagnosis.

For some of the UP descriptions, students consistently selected the wrong top-level node for the UP (top-level nodes map to stages in the Interaction Cycle – see Section 2.3.1). As I read through the students' rationale for selecting a different top-level node, it became clear that the UP descriptions did not provide the necessary information for helping them distinguish between stages in the Interaction Cycle. For example, a particular UP description read, "The color coding scheme for seats is problematic for individuals with red/blue color blindness. In addition, in the detailed seat view, purple isn't noticeable as a color." The expert diagnosis had the Translation stage as its top-level node because the expert considered the poor color coding to affect the user's ability to determine what to do next in the task of selecting a seat for a theatre performance. The expert believed that color blindness would prevent users from recognizing that the seats were in fact selectable objects. The students did not have this information because it was not explicitly recorded in the UP description, and they assumed that it was a Physical Actions UP that resulted from the inability of a color blind user to determine the availability of a seat based on the colors of red, blue, and purple. For example, one student's rational read, "I chose this [Physical Actions] because I assumed that the person knew how to select the seats".

Examples such as the previous one helped me to realize that the UP descriptions did not contain all the information necessary to correctly diagnose the UP that they described. I decided to run another study to determine if more experienced UAF users would have the same difficulties.

The second study used verbal protocol taken from six UE graduate students. These students were intermediate users of the UAF who had participated in a training session.

I worked with each student for two hours. The students used the PRT to diagnose UPs described in UP descriptions from professional UE labs in the same manner as in the first part of the study. Via verbal protocol, I asked the students to talk me through the node decision process and tell me when they felt that they were confused. As in the first study, the students had trouble deciding between stages of the Interaction Cycle for some of the UP descriptions.

The inability of the students to choose the correct top-level category for the UPs was directly related to the lack of necessary information in the UP descriptions. This lack of information is particularly problematic given the fact that the usability data collection stage and the analysis stage, which includes diagnosis, are separate in typical usability evaluation sub-processes.

2.4.2 Analogy to Medical Diagnosis

The need for problem diagnosis is not new with UE; it is central to any domain that involves finding and fixing problems, including automobile repair and the medical field. In medicine, a nurse might see the patient, gather some initial data via common measurements such as temperature and blood pressure, and take a statement of the patient's complaint. The doctor will review this initial information, possibly making more measurements and observations, and will probably ask the patient to repeat a description of the complaint. Throughout the case, the doctor draws on a structured knowledge base of medical concepts and issues that relates symptoms with diseases and serves as a guide to formulating potential diagnoses (diagnostic hypotheses).

Even while the patient is still in the examining room, the doctor begins to use the medical knowledge framework to highlight common and distinguishing characteristics among the potential diagnoses and to determine and ask questions that represent additional information necessary to rule in or rule out each of these diagnostic hypotheses.

This initial analysis then drives further data collection as the doctor makes more measurements and observations (e.g., looks in patient's throat) and asks the patient more questions (e.g., about symptoms, background), seeking to prune the hypotheses. This "micro-iteration" (using my term) of data collection with analysis taps information that was not collected initially but is still available (for example, by asking the patient or, if necessary, bringing the patient back for a return visit), just when it is needed for diagnosis. The medical procedure supports micro-iteration but, while the typical UE cycle is iterative overall, it does not support micro-iteration between data collection and analysis.

Capra [2001] makes an interesting comparison between UP diagnosis and medical diagnosis. Both forms of diagnosis rely on expertise (i.e., skill and experience) rather than just factual knowledge, and both require the ability to focus on relevant information and discard irrelevant information. Work by Griffen et al. [1998] on implicit processes in medical diagnosis offers some opportunities to draw interesting parallels with diagnosis in UE. Diseases may have many signs (observed by a doctor) or symptoms (experienced by a patient), only some of which are present for a given instance of the disease. In much the same way, UPs often manifest themselves with users in different manners, and very different design flaws may have similar manifestations. UE, however, has the advantage that there are often opportunities to observe the cause (flaw in the design) and effect (on the user) relationship in fairly close proximity if usability practitioners are sensitive to it.

My concept of micro-iteration has a counterpart in a type of reasoning used by medical doctors called hypothetico-deductive reasoning, which involves generating several diagnostic hypotheses and then working backwards to prove or disprove them [Godfrey-Smith, 2003]. While working backwards, it may be necessary to collect additional data to disambiguate or distinguish a hypothesis. Doctors interact with the patients to develop hypotheses about potential diseases, identifying key distinguishers to determine what tests are needed to get critical data to rule in or rule out these diseases. The diagnosis process is iterative and doctors continue to interact with patients until they get the necessary data.

2.5 Formative Studies of the Wizard

The exploratory studies and analogies to medical diagnosis documented in Section 2.4 helped me determine the need for micro-iteration. Immediate intention is a term used to refer to the necessary information that usability practitioners need to identify and record during micro-iteration. More specifically, immediate intention provides information about what the participant is doing when he encounters a design flaw that results in a UP.

Evaluators need some kind of support in asking the right questions to elicit immediate intention information. The UAF has proved to be useful in structuring the process of capturing missing UP data, but the UAF is intended for use in the analysis stage and is probably too bulky and time-consuming for use by most evaluators for initial diagnosis as part of the usability data collection stage. As a result, I developed the Wizard, a lighter-weight tool that is limited to the top-levels of the UAF (the Interaction Cycle) and tailored specifically for helping evaluators to quickly identify the immediate intention associated with a UP during micro-iteration.

I describe two formative studies of the Wizard in Howarth [2006]. Because these studies were formative with the primary goal of improving the Wizard design, I report here my lessons learned only as informal observations or intuitive insights, and not as formal results or statistically significant claims. My observations during these studies suggested to me that the Wizard has potential as an effective tool for helping evaluators determine the correct stage of the Interaction Cycle for a given UP.

For the first formative study, I developed a static prototype (a series of linked web pages) of the Wizard that had abstract descriptions of stages of the Interaction Cycle and concrete examples. Table 4 contains the pairs of questions presented at each decision point. The participants for the first study, 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. The participants were given 10 UP descriptions with varying levels of immediate intention specified. After the participants read a UP description, I first asked them to choose a stage in the Interaction Cycle and then had them use the Wizard. The participants had one hour to complete as many identifications as they could and were allowed to skip UP descriptions and return if they had time.

Table 4: 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)

The results of the first study are shown in Table 5. Participants 1, 3, and 5 did not complete all the identifications; the forward slash indicates the number of correct identifications as compared to the total number attempted. The non-UAF user did not feel that he was capable of selecting a stage in the Interaction Cycle first and only used the Wizard. Table 5 also contains counts of the number of correct identifications confirmed by the Wizard, the number of incorrect identifications corrected by the Wizard, and the number of times the participant was led astray by the Wizard after making a correct diagnosis (Confirmed, Corrected, and Led Astray, respectively).

Table 5: Results of the first Wizard study

	P1 beg	**P2** beg	**P3** non	**P4** ex	**P5** in	**P6** ex
W/O Wizard - Correct	3/5	6/10		10/10	7/8	10/10
With Wizard - Correct - Confirmed - Corrected - Led astray	1/5 1 0 2	6/10 4 2 2	4/7	10/10 10 0 0	7/8 7 0 0	10/10 10 0 0

non = non-UAF, beg = beginner, in = intermediate, ex = expert

Participants 4 and 6, expert UAF users, identified the correct stage of the Interaction Cycle for all UPs and then confirmed their choices with the Wizard. After using the Wizard a few times, they began to focus only on key words in the abstractions and used the Wizard much more rapidly. Participant 5, an intermediate user, spent more time describing the decision process than did the expert users (as part of verbal protocol) and did not complete all 10 UP descriptions. On the completed UP descriptions, however, the intermediate user performed almost as well as the expert users. These results suggest that the Wizard helped the advanced users of the UAF improve their identification speed and associate words and concepts with stages of the Interaction Cycle.

One beginning user (participant 1) performed particularly poorly with the Wizard and the other (participant 2) performed well overall. They both, however, were led astray twice by the wording in the Wizard, which indicated that it required improvement. Participant 3, the non-UAF user who had no experience with the UAF and no training, was able to use the Wizard to make four correct identifications. This result suggests that the Wizard could be a useful training tool.

The feedback provided by the participants led me to develop a second version of the Wizard (Table 6). In this version, the most important change was in the wording of the questions and examples. Verbal protocol in the first study revealed that certain words and phrases confused the participants and led to misdiagnoses. For example, the question for the Outcome and System

Functionality stage in the first version read: “Is your problem one that is internal to the system and invisible to the user?” The participants, particularly those with limited experience with the UAF, did not understand the phrase “internal to the system.” Several times, in fact, these participants selected this choice when the UP 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 mistakes. I corrected the problem in the second version by specifying that the Outcome category referred to backend functional issues not in the user interface software.

The first study strengthened my belief that intermediate and expert UAF users can do well identifying the correct node of the UAF to specify immediate intention with the Wizard, and I wanted to see if novices (with respect to the UAF and the Wizard) could do the same. The second Wizard study included five participants who had some familiarity with UE, but who were not familiar with the UAF. These participants had not participated in the first study. For this study, I gave each participant a five minute training course on the Interaction Cycle, so that they could select a stage without using the Wizard to avoid the situation experienced by participant 3 in the first study. Each new participant was given the same 10 UP descriptions that we used in the first study. For the first five UP descriptions, I had the participants first choose a stage of the Interaction Cycle that specified the immediate intention of the UP described in the UP description without the Wizard and then with the Wizard; these allowed me to test the Wizard’s ability to provide confirmation for correct identifications and help users after incorrect identifications. For the next three UP descriptions, I had the participants use only the Wizard, which allowed me to evaluate the Wizard’s ability to help users select the correct stage of the Interaction Cycle the first time. For the last two UP descriptions, I had the participants select a stage in the Interaction Cycle without the Wizard and then tell me words and phrases that they had learned while using the Wizard that had helped them make a decision. With these UPs I hoped to indirectly evaluate what the participants had learned.

Table 7 shows the results of the second Wizard study. The results suggest that the second version of the Wizard helped new users of the UAF identify the correct stage of the Interaction Cycle. Only participant 3 was led to a misdiagnosis once by the Wizard. In addition, all participants correctly identified the last two UP descriptions without the Wizard, which suggests that they had learned from the Wizard and were incorporating the concepts that help to distinguish between stages of the Interaction Cycle.

Table 6: Second version of the Wizard

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 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 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 7: Results of the second Wizard study

	P1	P2	P3	P4	P5
Problems 1-5					
- W/O Wizard	2	1	3	3	3
- With Wizard	3	3	2	4	3
-- Confirmed	2	1	2	3	3
-- Corrected	1	2	0	1	0
-- Led astray	0	0	1	0	0
Problems 6-8 - Correct	2	3	3	2	2
Problems 9-10 - Correct	2	2	2	2	2

3 Current State of Usability Engineering Tool Support

This section is intended to answer research questions RQ1a and RQ1b (Section 1.5). Specifically, it includes a discussion of difficulties experienced by usability practitioners and features found in existing UE tools.

3.1 Categorization of Difficulties

The related work in Section 2.1 on evaluator effect and skill, the content of UP descriptions, and the content of usability evaluation reports describes difficulties with current UE processes, specifically the usability evaluation sub-process. While knowledge of these difficulties as they are documented is useful, I needed to consider them at a more abstract level. I developed the following three general categories of difficulties that map directly to the stages of the usability evaluation sub-process shown in Figure 3:

- Identifying and recording critical usability data (the usability data collection stage)
- Understanding and relating usability data (the UP analysis stage)
- Communicating usability information (the usability evaluation reporting stage)

Difficulties with identifying and recording critical usability data occur during the usability data collection stage while difficulties understanding and relating the data occur during the UP analysis stage. A certain amount of communication occurs among the individuals involved with the usability evaluation sub-process, but this dissertation is most concerned with the difficulties with communicating usability information that occur during the construction of a usability evaluation report in the usability evaluation reporting stage.

3.1.1 Identifying and Recording Critical Usability Data

When a facilitator observes a participant experiencing a critical incident, such as during lab-based testing, there is such an enormous amount of data and details in the context that the facilitator often cannot know what is important to record. In addition, facilitators have to record potentially large numbers of critical incidents and, out of necessity, write brief descriptions or comments for each, so that they can keep up with the participants. In such a situation, less experienced facilitators (Section 2.1.1) may not know what to record and may not have the benefit of a structured format for recording data (Section 2.1.2). Data not collected are lost early on and are not available for use later in the usability evaluation sub-process or the overall UE process.

3.1.2 Understanding and Relating Usability Data

Usability data may exist in a variety of forms such as notes, video, audio, and textual critical incident descriptions and may come from a variety of sources such as self reports by remote users, usability lab testing data, and inspections performed by usability experts. As such, it may be difficult for a problem analyst to make meaning out of the data and recognize UPs, particularly if the data are unstructured (Section 2.1.2). In addition, the problem analyst is operating in a mostly open-loop fashion (i.e., without feedback to the usability data collection stage) making it difficult to answer questions and resolve ambiguities. The analyst can sometimes ask questions of the facilitator who collected the data, but often at significant effort. The problem analyst is likely to have important questions for the participant (for empirical testing), but neither has access to the participant after usability data collection is completed. As a result, too often the problem analyst can only try to interpret and reconstruct the missing usability data; the degree of the completeness of the resulting UP descriptions is highly dependent on the knowledge and experience of the problem analyst (Section 2.1.1).

3.1.3 Communicating Usability Information

In real-world projects, project team members have many responsibilities for many parts of possibly many projects and cannot necessarily maintain continuity of information flow throughout the UE process for one particular product. Systems analysis and design often are separated from usability evaluation by a delay in time (that affects human memory), are performed by different people (affected by poor communication), and occur at different physical locations, rendering all information not well communicated to be unrecoverable. Usability evaluation reports lacking contextual information and containing brief UP descriptions good enough for the problem analyst at the time of UP analysis end up being too vague for the designers who were not necessarily present for the usability testing (Section 2.1.3).

3.2 Existing Usability Engineering Tool Features

Section 3.1 focuses on identifying and categorizing difficulties experienced by usability practitioners to support my statement of the problem that typical UE processes are not as effective as they could be. In this section, I identify features in state-of-the-art UE tools (Section 2.2.3.5) and analyze how they address the categories of difficulties developed in Section 3.1. The general argument is that the effectiveness of the UE process is negatively affected by the difficulties experienced by usability practitioners. These difficulties would not be present in the literature if existing UE tools addressed them.

3.2.1 Low-Level Data Capture

Morae and Ovo Logger allow usability practitioners to capture low-level data including keystrokes, mouse clicks, and system events. These tools also allow usability practitioners to search, sort, and filter logged low-level data.

The focus on low-level data capture in many of the UE tools in the survey may be technology led [Macleod & Rengger, 1993]. Data capture may be partially or completely automated, but accurate UP analysis requires a thorough review of the data by a problem analyst. The volume of data produced by low-level events, particularly when capture is automated, can be overwhelming [Hammontree et al., 1992, Theaker *et al.*, 1989]. As such, low-level data capture is not included in the list of desirable features. Although it may be useful, it does not directly support usability practitioners in understanding and critiquing the usability of an interaction design.

3.2.2 Metrics

Morae, Spectator, and Ovo Logger can generate a number of metrics, such as time or activity metrics.

Like low-level data capture features, features for computing metrics have been excluded from the list of desirable features. Such features are outside of the scope of this dissertation. As stated in Section 1.3.4, the focus is on formative usability evaluations. Tool support for calculating metrics from quantitative data is most directly related with summative usability evaluations, in which values are used to statistically validate or invalidate certain properties of an interaction design.

3.2.3 Screen Video Capture

Morae, Visual Mark, Spectator, and Ovo Logger all provide for the capture and integration of digital screen video.

The use of video in usability testing is well documented (for examples, see [Badre *et al.*, 1994, Kennedy, 1989]). Screen video captures aids usability practitioners in the usability data collection and the usability evaluation reporting stages. Usability practitioners can use screen video capture to review sessions to identify UP instances that they may have missed during the live session with the participant. I can provide no suggestions for improving the support for screen video that is found in tools like Morae, Ovo Logger, and Spectator, and as a result, I do not address this feature in this dissertation.

3.2.4 Observational Capture

Observations are comments made by usability practitioners during usability data collection. All state-of-the-art tools in the survey support the logging of observational comments. These comments map to the raw usability data shown in the usability data collection stage of Figure 3.

Observational capture facilitates the usability data collection stage by supporting the recording of usability data. All the state-of-the-art tools provide some mechanism for time stamping and logging comments. Those tools that support event definitions also allow observational comments to be associated with events. Observational comments, however, have the potential to cause difficulties in the UP analysis stage because they are free form and may lead to the ad hoc unstructured data described by Andre et al. [2001]. Large numbers of brief or terse comments may prove difficult to integrate, particularly if the person performing UP analysis is not the same person who recorded the comments.

3.2.5 Configuration Support

Configuration support includes functionality for configuring usability evaluation sessions. Both Spectator and Ovo Logger have task (scenario in Ovo Logger) and participant databases. Spectator also has a project database that supports up to five definable levels (project, subproject, etc).

Configuration support as it exists in state-of-the-art tools is very useful in the usability data collection stage. Providing details about the session and task provides a context for collected data. This context limits the amount of data that a facilitator needs to record during usability data collection. For example, the task object already contains a description of the task, so the facilitator does not need to include any of these details in comments about UP instances experienced by participants during the task. Configuration support also provides benefits for relating data during UP analysis. Spectator's project database allows problem analysts to pool usability sessions, so that they can more easily relate data.

3.2.6 Event definitions

Tool support for event definitions allows usability practitioners to create event objects that represent events of interest, such as errors committed by the participant. Spectator provides for event types with definable behaviors while both Ovo Logger and the Usability Activity Log allow for the creation of categories for observations. All three tools allow usability practitioners to associate hot keys with events, so that they can be easily tagged during usability evaluations.

Event definitions primarily support facilitators in the usability data collection stage because they facilitate the identification and recording of usability data. They support identification of data by enabling the definition of events before a session with a participant. Because usability practitioners will have already established the events that they deem important before a session, they will be more aware of these predefined events and more easily able to identify them during the session.

Event definitions support recording of usability data through mappings to hot keys that make it easy to rapidly record events.

The same aspect of event definitions that enables the identification of important usability data also has the potential to complicate it. A usability practitioner may define a large number of events before a session. In such a case, identification becomes a multi-way decision among multiple events. If the events are sufficiently different from one another, this multi-way decision is not problematic. However, if multiple events are similar, rapid selection of an event may be difficult. For example, it may be difficult to distinguish between a simple error in which a participant inadvertently clicks on an incorrect button or link and a more serious error in which a participant misinterprets the meaning of the label on a button or link.

Event definitions also have the potential to cause difficulties with understanding and relating usability data in the UP analysis stage. Without standardization, event definitions may vary from session to session, which will make it more difficult for usability practitioners to familiarize themselves with definitions and apply them consistently. As a result, it may be difficult to compare data among sessions to develop an understanding of trends and patterns. In addition, support for relationships among events in state-of-the-art-tools would help with relating usability data. DRUM, while not a state-of-the-art tool, provides for hierarchies of event definitions that relate events by task. Support in state-of-the-art tools is limited to functionality like that found in Spectator that allows usability practitioners to specify that one behavior is exclusive and terminates other behaviors.

4 Desirable Tool Features

This section is intended to answer research questions RQ2a and RQ2b (Section 1.5). Specifically, I develop a set of desirable features for UE tools and describe a tool that implements specific instances of these features.

4.1 Abstract Descriptions

In this section, I use the results of the analysis in Section 3.2 to create a set of desirable features for a UE tool. The features that I include in the set of desirable features are of two types. The first type is features from state-of-the-art tools that I have modified or extended to address some of the difficulties identified in the analysis. The second type is features that I suggest to meet a need identified through the analysis. I describe these features at an abstract level.

4.1.1 Usability Problem Instance Records

The observation capture features (Section 3.2.4) allow for the creation of logs of time-stamped comments. Evaluators, however, must manually review these comments and combine them to form UP instances. A desirable tool feature would allow for the creation of UP instance records while the evaluator is observing the participant. These records would serve as the most basic unit of usability data; one UP instance record would contain enough data to completely specify one UP instance. UP instance records would allow evaluators to work at a higher level of abstraction thereby addressing difficulties with identifying and recording critical usability data (Section 3.1.1) and understanding and relating usability data (Section 3.1.2). Figure 5 shows the difference between the evaluator comments that represent raw usability data and UP instances for a sample photo album application; comments C1-C5 are combined into UP instance UPI1.

The idea for UP instance records takes into account Vygotsky's [1978] concept of the zone of proximal development, which is the distance between what an individual can do on his own and what he could be helped to achieve with competent assistance; scaffolding is a term used to describe this assistance. The idea of scaffolding is not new to HCI; several research efforts have included some form of scaffolding (see [Jackson *et al.*, 1996, Quintana *et al.*, 2002, Rosson *et al.*, 1990, Soloway *et al.*, 1994] for examples). I believe that making the leap from raw usability data in the form of comments to UPs is difficult for novice practitioners. UP instances serve as a scaffolding to help novice usability practitioners construct UPs from comments.

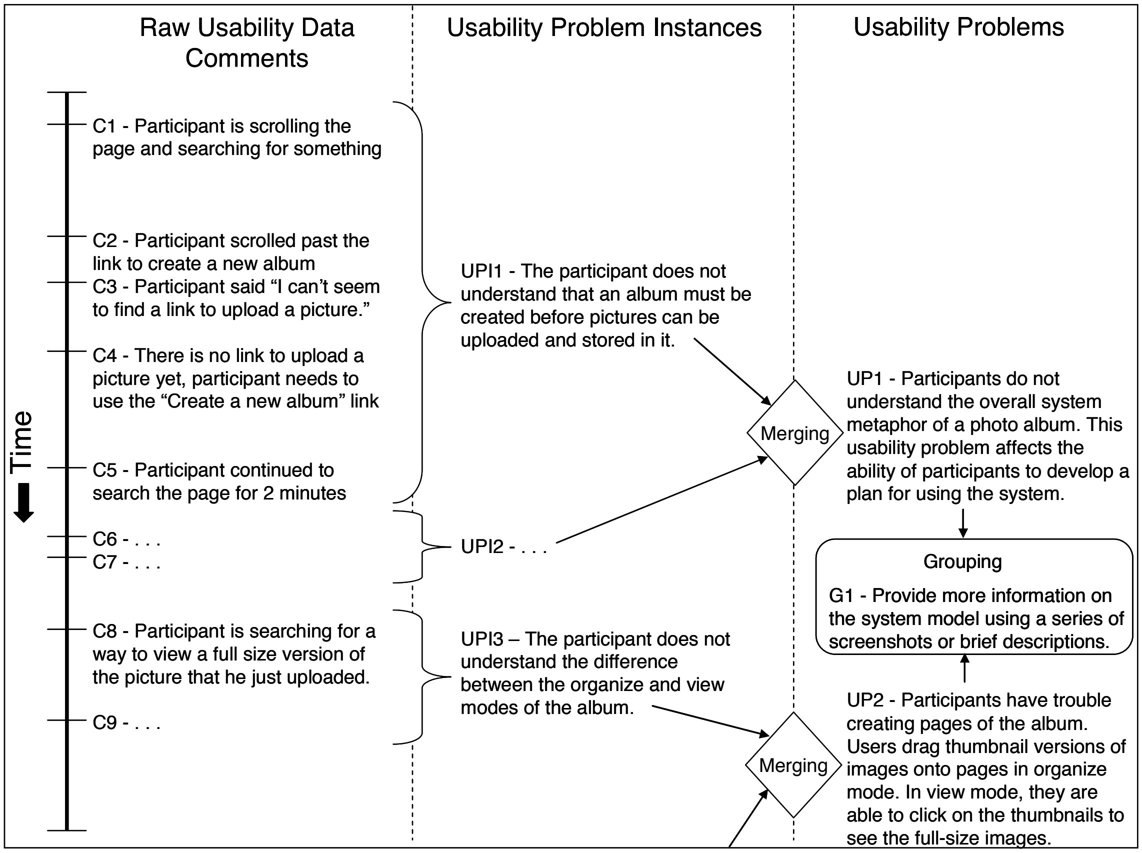

Raw Usability Data Comments
Usability Problem Instances
Usability Problems
Time
C1 - Participant is scrolling the page and searching for something
C2 - Participant scrolled past the link to create a new album
C3 - Participant said "I can't seem to find a link to upload a picture."
C4 - There is no link to upload a picture yet, participant needs to use the "Create a new album" link
C5 - Participant continued to search the page for 2 minutes
C6 - . . .
C7 - . . .
C8 - Participant is searching for a way to view a full size version of the picture that he just uploaded.
C9 - . . .
UPI1 - The participant does not understand that an album must be created before pictures can be uploaded and stored in it.
UPI2 - . . .
UPI3 – The participant does not understand the difference between the organize and view modes of the album.
Merging
Merging
UP1 - Participants do not understand the overall system metaphor of a photo album. This usability problem affects the ability of participants to develop a plan for using the system.
Grouping
G1 - Provide more information on the system model using a series of screenshots or brief descriptions.
UP2 - Participants have trouble creating pages of the album. Users drag thumbnail versions of images onto pages in organize mode. In view mode, they are able to click on the thumbnails to see the full-size images.

Figure 5: Levels of usability problem data for an example photo album application

4.1.2 Diagnosis

A feature to support diagnosis is intended to address the difficulties associated with understanding and relating usability data (Section 3.1.2). The idea for the feature developed from the analysis of the event definition feature (Section 3.2.6) that exists in state-of-the-art tools. The potential variability of event definitions among sessions could complicate efforts to understand and relate usability data. I propose using a conceptual framework of usability concepts to give usability practitioners a common way to understand and relate usability data and a common vocabulary for discussing it.

Gray and Salzman [1998] noted:

> "To the naıve observer it might seem obvious that the field of HCI would have a set of common categories with which to discuss one of its most basic concepts: usability. We do not. Instead we have a hodgepodge collection of do-it-yourself categories and various collections of rules-of-thumb . . . Developing a common categorization scheme, preferably one

> grounded in theory, would allow us to compare types of usability problems across different types of software and interfaces" (p. 241).

I agree; I believe a conceptual framework and standard usability vocabulary are essential to organize and guide UP analysis.

4.1.3 Merging and Grouping

A feature for merging UP instances and grouping UPs is intended to help address difficulties with understanding and relating usability data (Section 3.1.2) and communicating usability information (Section 3.1.3). Ovo Logger allows for the association of observations with entries in a usability evaluation report; this functionality is a combination of merging and grouping. My proposed feature separates merging and grouping in a structured manner and works with UP instances.

Much research has been devoted to developing usability evaluation methods that are used in evaluations of interaction designs. Example usability evaluation methods include cognitive walkthroughs [Polson et al., 1992], heuristic evaluations [Nielsen, 1992, Nielsen, 1994, Nielsen & Molich, 1990], remote usability evaluation methods [Castillo *et al.*, 1998], and empirical testing [Hix & Hartson, 1993a]. The focus of these methods is the generation of lists of UPs.

More recently, however, research has shifted away from methods and comparisons of methods to issues of how to use the data generated by methods. Researchers have begun to look beyond the detection of UPs to other aspects of importance. Wixon [2003], for example, discusses issues that are important in actually fixing UPs, such as resource limitations and contextual factors. Hornbæk and Frøkjær [2005] also take a practical perspective and discuss the effectiveness of redesign proposals to accompany UPs.

It is no longer enough to simply identify UPs, they must be combined in a meaningful way. The proposed feature combines merging from the UP analysis stage, which is necessary to combine similar UP instances, and grouping from the usability evaluation reporting stage, which is necessary to relate UPs, to report usability information in a manner that is useful to other individuals involved in the UE process. Figure 5 illustrates merging UP instances to form UPs; UP instances UPI1 and UPI2 are merged to form UP1. Figure 5 also depicts grouping UPs; UP1 and UP2 are related through group G1.

4.2 Specific Instances

In this section, I propose specific instances of features that are in accordance with the abstract descriptions in Section 4.1.

4.2.1 Usability Problem Instance Records

I propose a feature that is in accordance with the abstract description presented in Section 4.1.1 for UP instance records. This is just one instance of a feature; other features could also be proposed. The same is true for the specific instances of features introduced in Sections 4.1.2 and 4.1.3.

The proposed feature for UP instance records is based on a concept that I refer to as hierarchical context. UP instance records are nested within a multi-level structure of contexts. I begin this section by describing the levels of context and then introduce a specific UP instance record format.

4.2.1.1 Context

Associating UP instance records with a particular context would reduce the amount of data that a facilitator must record to specify a UP instance in the usability data collection stage and help problem analysts understand and relate UP instances. Existing configuration support features (Section 3.2.5) involve databases that store details about projects, sessions, tasks, participants, and facilitators. These provide some context, but I believe that it is important to record more context. Context implies an understanding of the circumstances in which something occurs. It is referred to repeatedly as being of high importance, but specifically what context entails is difficult to define. Much previous work uses context in a general sense; one notable exception is work by Lavery et al. [1997] that defines context for a structured UP instance record as the user context, the interaction context, and the work context. The authors, however, do not describe any of these types of contexts in detail.

I have defined a number of levels of context to create a hierarchical context inside of which UP instance records are nested. The top level of the hierarchy involves the broadest context. The second level of the hierarchy is nested inside the first and has a narrower context. Each progressive level is nested inside the previous one and has a narrower context. As a result, the more deeply nested the level, the more specific the context.

Figure 6 shows the six levels of hierarchical context that I have developed in an attempt to help practitioners better specify and capture context: organization, project, version, session, task run, and problem. The organization, project, and version levels provide general application context, such as the need or purpose for the application and its target environment. The session and task run contexts are directly related to the configuration support features in existing tools (Section 3.2.5). Finally the problem context, the most deeply nested level, contains details about UP instances experienced by participants.

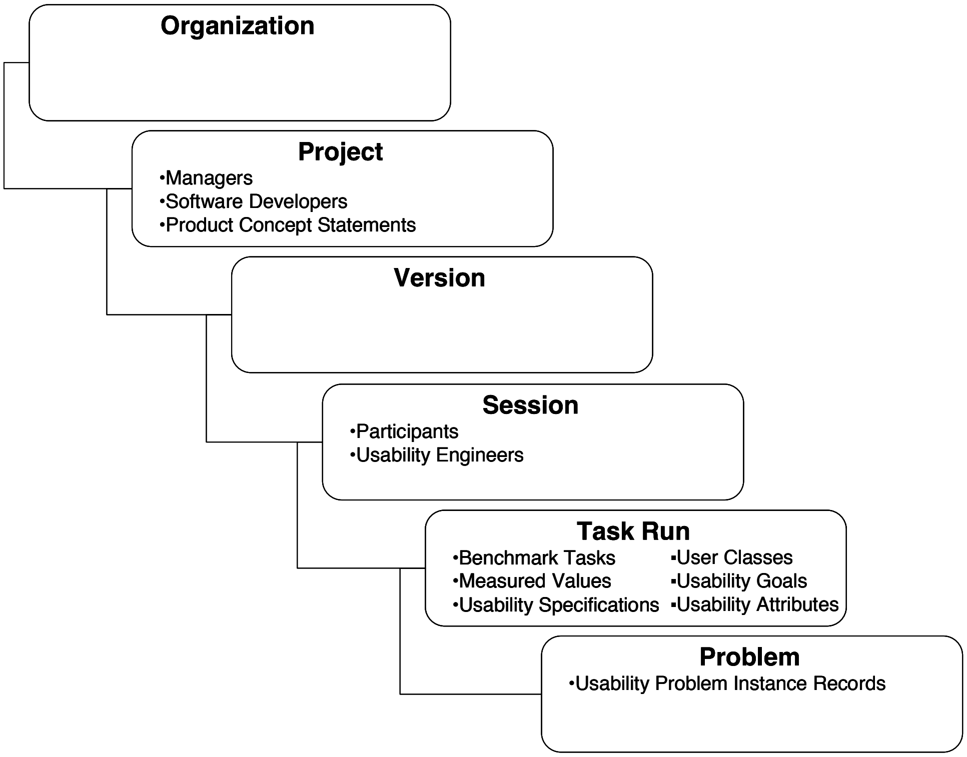


Figure 6: Levels of hierarchical context and associated resources

The organization level contains details about an organization. The project level contains details about software applications that an organization wants to evaluate, and the version level contains details about each of the versions of a project. The session level represents a session between one or more facilitators and one or more participants. The task run level represents one task as performed by a participant or participants as part of a session. Finally, the problem level represents a UP instance experienced by a participant during a task run.

All levels of the hierarchy except the organization and version levels have resources associated with them. The term resource is used for people or objects that perform a function at a given context level. The following is a list of resources by context level:

- Project
    - Managers – Manage projects by assigning individuals to them and allocating resources for them.
    - Software developers – Develop prototypes for use in the usability evaluation sub-process.

    - Product concept statements – A brief descriptive summary of the product being developed. As a kind of mission statement for the project, the product concept statement is typically 50-75 words in length and sets the focus and scope for the design team in the overall development effort.

- Session

    - Participants – Participate in usability evaluation sessions.

    - Usability practitioners – Collect, analyze, and report data in the usability evaluation sub-process.

- Task Run – The resources included at the task run level are adapted from the approach to UE developed by Hix and Hartson [1993a]. The resources are as follows:

    - User classes – Descriptions about the various roles users play while interacting with the system. These descriptions provide a set of attributes such as users' knowledge of computers or users' training and application-related experiences and guide the overall design effort. For example, for a user class with little to no computer knowledge or training, the system design will probably include a significant amount of “handholding” with detailed instructions for each stage of the interaction. This might contrast with the design for another user class with extensive computer knowledge and domain expertise where the focus will probably be on providing “power” features with shortcut keys.

    - Usability goals – High-level objectives stated in terms of usability and design of user interaction. They reflect real use of a product in the real world and determine what is important to an organization and its users. Usability goals may be market driven. Examples include customer satisfaction and walk-up-and-use usability.

    - Usability attributes – The general usability characteristic that is to be measured for an interface. Some common usability attributes include: initial performance, long-term performance, learnability, retainability, advanced feature usage, first impression, and long-term user satisfaction.

    - Benchmark tasks – Standardized unambiguous descriptions of representative, frequently performed, and critical tasks, to be used in usability evaluation tests.

    - Measured values – Quantitative data that are collected from a user during or after a user interacts with a software system. These values can be either objective or subjective. Objective measured values are quantitative measures of observable user performance while performing tasks with a user interface. Subjective measured values are quantitative measures based on user opinion about the user interface.

  - Usability specifications - Quantitative usability goals against which user interaction design is measured. They include target levels for usability attributes and are often used as a guide and process management tool to know whether the development process is converging toward a successful design.

- Problem
  - UP instance record – A record of a user experiencing a usability problem. The fields included in a UP instance record are described in Section 4.2.1.2.

4.2.1.2 Usability Problem Instance Record Format

In addition to context, a consistent UP instance record format for UP instance records would standardize the way in which UP instance data are recorded. Such a format would make facilitators aware of needed usability data in the usability data collection stage and provide problem analysts with more consistent data in the UP analysis stage. UP instance records exist within the problem context (Figure 6).

Based on my synthesis of the related work presented in Section 2.1.2 and my own experience, I suggest the use of a record format that includes the following three types of data and associated fields:

- Descriptive – These data describe the UP instance itself including outcomes experienced by the participant.
  - Name of the UP instance
  - Description of the UP instance
  - User interface object(s) involved
  - Relevant designer knowledge
  - Timestamp and associated video recording
- Diagnostic – These data describe the cause of the UP instance.
  - A UP instance diagnosis
- Prescriptive – These data contain suggestions for fixing the UP instance.
  - Suggestions for fixing the UP instance

4.2.2 Diagnosis

I propose a feature that is in accordance with the abstract description of a desirable feature for diagnosis introduced in Section 4.1.2. I adapt the User Action Framework (UAF) as a conceptual framework (Section 2.3). The UAF provides a structured framework of usability concepts and issues for understanding a UP in terms of its problem type, how it interfered with a user's sensory, cognitive, or physical actions in task performance, and its causes within the interaction design [Hartson *et al.*, 1999]. I support two levels of diagnosis: full and partial.

4.2.2.1 Full Diagnosis

Diagnosis involves associating a UP (as described in a UP instance record or a UP description) with the correct usability concept that describes its cause within the interaction design. Work by Springett [1998], 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. The UAF is intended to help usability practitioners consistently determine the correct link and translate it into a diagnosis.

The process of diagnosis with the UAF involves associating a UP with a path of UAF nodes that completely describes the UP and its causes. Figure 7 shows the Interaction Cycle of Figure 4 extended into the full UAF, a tree structure of usability concepts representing the multidimensional space of design features and UP data. The three dots to the right of the tree are an ellipsis that indicates that the tree continues many levels deeper; only three levels are shown in the illustration. A tree structure allows a problem analyst to navigate the dimensions of the space, arriving at a specific location within the space. Each level of the tree structure maps to a dimension, and each node (diagnosis choice) 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 attributes that don't apply to a given usability situation, thereby filtering or pruning off irrelevant sub-trees. Making these choices while traversing the full depth of the tree is equivalent to selecting a path within a decision tree, thereby building a set of dimensions and attributes (one pair for each node in the path) that best represents the UP and its causes.

Once a UP has been associated with a node, the path to that node contains all the information needed to identify the UP specifically. Precision is ensured by the standardized usability vocabulary used. Reliability is ensured because other UPs that have the same attributes will be placed in the same node, and completeness is also ensured because the process leads the problem analyst to include all the relevant usability attributes.

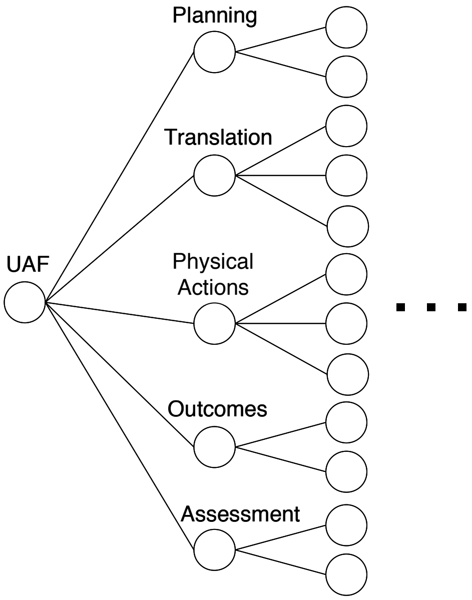


Figure 7: User Action Framework as a tree structure

Other existing techniques for understanding data include affinity diagrams, priority ranking, and Pareto diagrams; such techniques require grouping data for the purpose of organization [Nayak et al., 1995]. Trees provide a natural way for grouping related UPs, but do so with a structure that can be reused in future development efforts.

In addition, the UAF tree structure facilitates redesign by organizing UPs in a way that facilitates the identification of design changes. Nayak et al. argue that techniques that are easy to translate into solutions increase team acceptance. The UAF allows developers to understand the specific causes of the UP, and the changes necessary to correct it are often almost self-suggesting. Through time, developers can associate generic sample UP descriptions and solutions with nodes and increase the speed of the correction process by reusing UP analysis effort.

4.2.2.2 Partial Diagnosis

Full diagnosis with the UAF can be time consuming, and it is not practical to try to diagnose a UP during a session with a participant. Trying to perform full diagnosis by reviewing screen capture video after the session when the participant is gone, however, may also not be possible, especially if the necessary information for making a decision among multiple diagnoses is known only to the participant. It is therefore necessary to capture the right information about what a participant is doing or trying to do, which I refer to as immediate intention, during the usability data collection stage to enable complete and consistent diagnosis in the UP analysis stage. I propose modifying the usability evaluation sub-process to support a non-sequential, micro-iterative usability data collection and analysis process that I refer to as micro-iteration, which helps facilitators identify and capture the usability data needed by problem analysts to accurately and consistently diagnose problems.

4.2.2.2.1 Immediate Intention

Unlike medical doctors who have a structured diagnostic framework to help them determine what questions to ask and tests to run (Section 2.4.2), problem analysts often cannot know which diagnostic questions need answering until beginning the analysis stage, after the participant is typically gone. My exploratory studies in Section 2.4.1 suggested that these key early diagnostic questions involve very specific details about what the participant was doing or attempting and why at the time of experiencing a UP. I refer to these key details as the user's or participant's immediate intention, expressing them in terms of the type of user action involved (e.g., sensory, cognitive, physical) in the context of the location within the Interaction Cycle of the UAF (e.g., Planning, Translation, Assessment).

The UAF provides the necessary structure for determining which diagnostic questions apply and whether the appropriate data has been collected to completely specify immediate intention. Selecting a top-level node of the UAF completely specifies the kind of action that the participant was doing or attempting when he encountered an interaction design flaw. Understanding a participant's immediate intention therefore involves getting the data to distinguish among stages of the Interaction Cycle. For example, determining immediate intention for the UP with the seat selection interface in the first exploratory study in Section 2.4.1 would involve gathering data to distinguish between the Translation and Physical Actions stages.

Immediate intention allows designers to select an appropriate solution from a number of possible solutions. In some situations, one solution will fix UPs with different immediate intentions. In the seat selection example, changing the seat colors to ones that could be differentiated by color blind users would have fixed both Translation and Physical Actions UPs. In other situations, however, UPs with different immediate intentions have very different fixes. The following example illustrates this point.

A digital library website has a variety of tabs at the top of every page that serve as a navigation bar. A participant had trouble using the site to locate a specific journal because tabs associated with information-seeking tasks are mixed with those associated with other tasks. A possible solution to this UP is to reorder the tabs, so that tasks of a similar nature are adjacent to one another. This solution is sufficient if the participant had already planned for the task and was simply trying to determine which tab to select. In such a case, the participant has an immediate intention that maps to the Translation stage because he had already formulated a goal and developed a high-level task sequence to accomplish that goal. If the participant was not in the Translation stage, reordering the tabs may not be a sufficient solution. For example, if the participant was not familiar with digital library sites or with the functionality of the particular site being tested, he may not have formed a high-level task sequence before he experienced the UP. The participant's intention may have been to understand the site and determine possible uses. In such a case, the participant was in the Planning stage of the Interaction Cycle when the UP occurred, and the tab ordering problem is a planning problem. An appropriate solution for a planning problem might require additional organization of the tabs, possibly into groups labeled by high-level task and workflow categories, accompanied by a link to an overview page with descriptions of functions.

As the example illustrates, the difference in immediate intention results in two different diagnoses with potentially two different solutions. Key details needed to distinguish between the Planning and Interaction stages of the Interaction Cycle are necessary to help the developers know which UP is the real one that occurred for the participant and, therefore, which solution is most appropriate.

4.2.2.2.2 Micro-Iteration

The exploratory studies (Section 2.4.1) helped me realize that it is necessary to capture key data in the usability data collection stage to enable correct diagnosis in the analysis stage. If important diagnosis questions cannot be answered with data captured while the participant is present during usability data collection, it is difficult or even impossible to answer them later in the usability evaluation sub-process. I concluded that is necessary to move some of the UP analysis forward to the usability data collection stage to keep the facilitator in touch with the participant long enough to fill in the missing information, thereby reducing the information losses that occur in the current process. The part to be moved forward would have to be the minimum amount of analysis to determine and document participants' immediate intentions. Having captured the necessary immediate intention information, the evaluator can complete UP analysis and usability evaluation reporting after the testing subject is gone. This is a simple, but I believe, crucial conclusion, and it has reshaped my thinking about how UP analysis should be performed within the usability evaluation sub-process.

The changes to the usability evaluation sub-process can also be adapted to help clarify and better document the UP descriptions produced by inspection methods, such as heuristics or cognitive walkthroughs. The heuristic method, with its broad, general categories, can definitely benefit from more precise and specific UP descriptions. The cognitive walkthrough method, with its focus on task and sequences of actions, would be easy to adapt to explicitly include immediate intention information.

Including initial diagnosis may result in increased costs for the usability data collection stage because it requires keeping the participant for a longer period of time to establish and confirm immediate intention. The added cost, however, is the key to capturing the immediate intention information needed for UP analysis and usability evaluation reporting. Without the correct information, later stages of the UE process cycle could potentially be less effective and more costly.

4.2.2.2.3 Wizard

I have developed two important concepts: immediate intention and micro-iteration. In summary, immediate intention provides information about what the participant is doing when he experiences a UP. Micro-iteration is a modification to the usability evaluation sub-process that gives facilitators the chance to ask questions of the participant during empirical evaluations or of themselves in analytical evaluations to get the data that are necessary to determine immediate intention. In this section, I introduce the Wizard, a tool that is to be used during micro-iteration to help facilitators determine what to ask to specify immediate intention.

Evaluators need some kind of support in asking the right questions to elicit immediate intention information. The UAF has proved to be useful in structuring the process of capturing missing UP data, but the UAF is intended for use in the UP analysis stage and is probably too bulky and time-consuming for use by most evaluators for initial diagnosis as part of the usability data collection stage. As a result, I developed the Wizard, a lighter-weight tool that is limited to the top-levels of the UAF (the Interaction Cycle) and tailored specifically for helping evaluators to quickly identify the immediate intention associated with a UP during micro-iteration.

The exploratory studies (Section 2.4.1) helped me understand top-level diagnosis by allowing me to follow participants' thought processes while they tried to map UPs to stages in the Interaction Cycle. The participants generally understood what was represented by the stages of the Interaction Cycle, but they had no process for comparing them. I noticed that when I coached participants at making this top-level diagnostic decision in the second exploratory study, it helped to break the multi-way decision down into a sequence of dependent two-way decisions, allowing the evaluators to focus on a single issue or question at a time. Encouraged by initial success with this approach, I codified it into a sequence of two-answer questions, each comparing one stage of the Interaction Cycle with the other stages, based on the distinguishing attributes of that stage. Each answer prunes the number of stages remaining. Through a process of elimination, the Wizard helps evaluators home in on the correct stage. If at any point the evaluator is unable to answer a question, he should interact with the participant to get the answer. The sequence is designed to first rule out the least likely stages of the Interaction Cycle and then continue to the most likely stages. Stages are ruled out in the following order: Outcome and System Functionality, Overall, Physical Actions, Assessment, Planning, and Translation.

Figure 8 depicts the ruling-out strategy. Each black node represents a decision point where the UP analyst chooses between a given stage in the Interaction

Cycle and all the remaining stages. UP analysts start the Wizard by choosing between the Outcome and System Functionality stage and the rest of the Interaction Cycle.

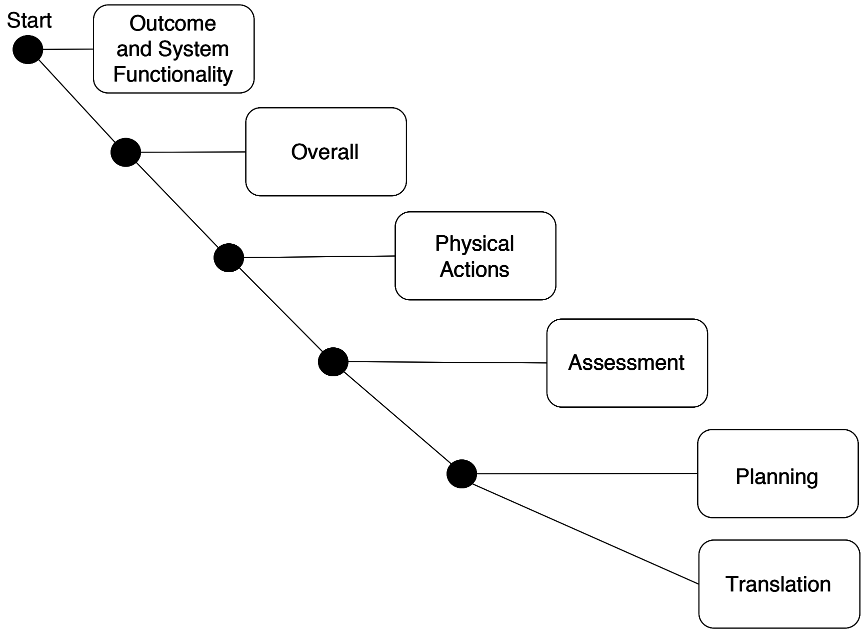


Figure 8: Wizard decision structure

A distinguisher is a set of words that tersely captures the essential difference between the semantics of one UAF node and the semantics of the other nodes. For example, the text for the Physical Actions node in the Wizard is as follows: “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?” In this way the Wizard brings the right distinguisher to bear at the right time and the right place for the evaluator. While the distinguishers needed are usually among the words in the UAF, most UAF nodes contain a description of the semantics of that node and not direct comparisons with other possible choices in sibling nodes. In contrast, the Wizard helps evaluators focus directly on the distinguishers by converting more verbose n-way UAF decision points into a series of more crisply stated binary questions based specifically on the differences between a given node and its siblings. At any one time, the facilitator can think about just one direct A vs. B face-off choice distinguished by participant immediate intention.

Section 2.5 documents formative evaluations that I performed with the Wizard. The results of the study suggest that the Wizard is useful in helping usability practitioners identify immediate intention.

4.2.3 Merging and Grouping

None of the existing tools provide support for merging UP instances and grouping UPs. I propose a feature that is in accordance with the abstract description of a desirable feature in Section 4.1.3.

4.2.3.1 Merging Usability Problem Instances

As discussed in Section 1.3.3, one or more usability problem instances may map to the same UP. The UP analysis stage of the evaluation sub-process involves merging UP instances into UPs to abstract out important information on interaction design flaws from the data produced during the usability data collection stage. The feature that I propose builds on the UP instance records described in 4.2.1. When UP instance records are merged into a UP record, they are treated as one object. UP records can be separated back into UP instance records if new understandings or relationships are uncovered. UP records have the same hierarchical context and report structure as UP instance records.

4.2.3.2 Grouping Usability Problems

UPs can be related in a number of ways. It is the job of the reporter to relate the data in the way that is best for the given situation to indicate to designers and developers which UPs should be considered together for redesign. Sometimes it is appropriate to relate UPs by task flow. In other cases it may facilitate redesign to organize UPs by interface objects, such as screens or dialogs. Additionally, it might be more appropriate to relate UPs by their cause. For example, if the team has access to a technical writer, it might be beneficial to relate all UPs dealing with terminology or the semantics of text used in the application.

Reporters weigh a variety of factors when preparing usability evaluation reports, such as budget and time constraints, software architecture issues, personnel limitations, and needs and abilities of target users. The grouping feature helps the reporters develop their own relationship schemes that are appropriate for their own situations or circumstances. Grouping UP records relates them, but unlike the merging feature, UP records in a group remain individual objects. One UP record may be included in zero or more groups.

4.3 Data Collection, Analysis, and Reporting Tool

I developed the Data Collection, Analysis, and Reporting Tool (DCART) to study the desirable features discussed in Section 4.2. In this section, I first describe the technical specifications of DCART. I then discuss how DCART supports each of the desirable features. In my discussion of each feature, I show screen shots from DCART version 1.1 of an evaluation of DCART version 0.1.

4.3.1 Technical Specifications

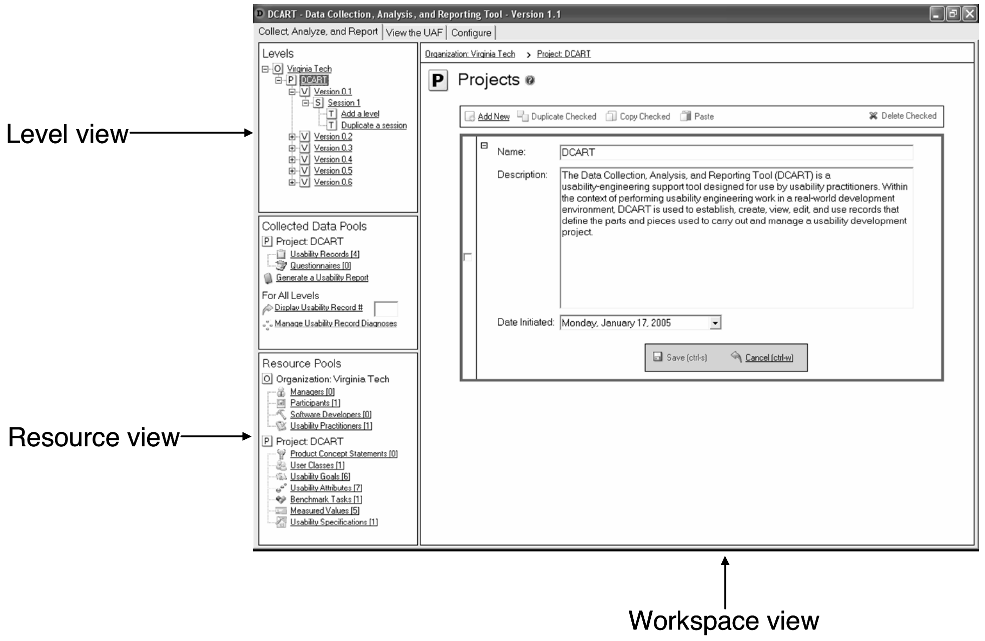
DCART is written in C# and uses the Microsoft .NET Framework version 1.1. It runs only under the Windows operating system. DCART can be configured to store data in local database files and in networked databases. DCART works with databases that support the ADO.NET OleDb provider. I use Microsoft Access for local database files and Microsoft SQL Server 2000 for networked databases.

4.3.2 Support for Usability Problem Instance Records

DCART provides support for the UP instance record feature described in Section 4.2.2. I first describe a structure that helps to capture context and then introduce a UP instance record format that is embedded in this structure.

4.3.2.1 Support for Context

Figure 9 shows support for levels of context and associated resources. The levels view in the top left hand corner shows the levels of context in a tree form. Nested contexts are represented as children of the parent context. The tree is expanded to show all the context levels; the letters to the left of the name of each node in the tree indicate the context level: organization (O), project (P), version (V), session (S), and task run (T). Clicking on a context level updates the resource view in the lower left hand corner and displays the level in the workspace view on the right side of the screen.



Name: DCART

Description: The Data Collection, Analysis, and Reporting Tool (DCART) is a usability-engineering support tool designed for use by usability practitioners. Within the context of performing usability engineering work in a real-world development environment, DCART is used to establish, create, view, edit, and use records that define the parts and pieces used to carry out and manage a usability development project.

Date Initiated: Monday, January 17, 2005

Figure 9: Support for levels of context and associated resources

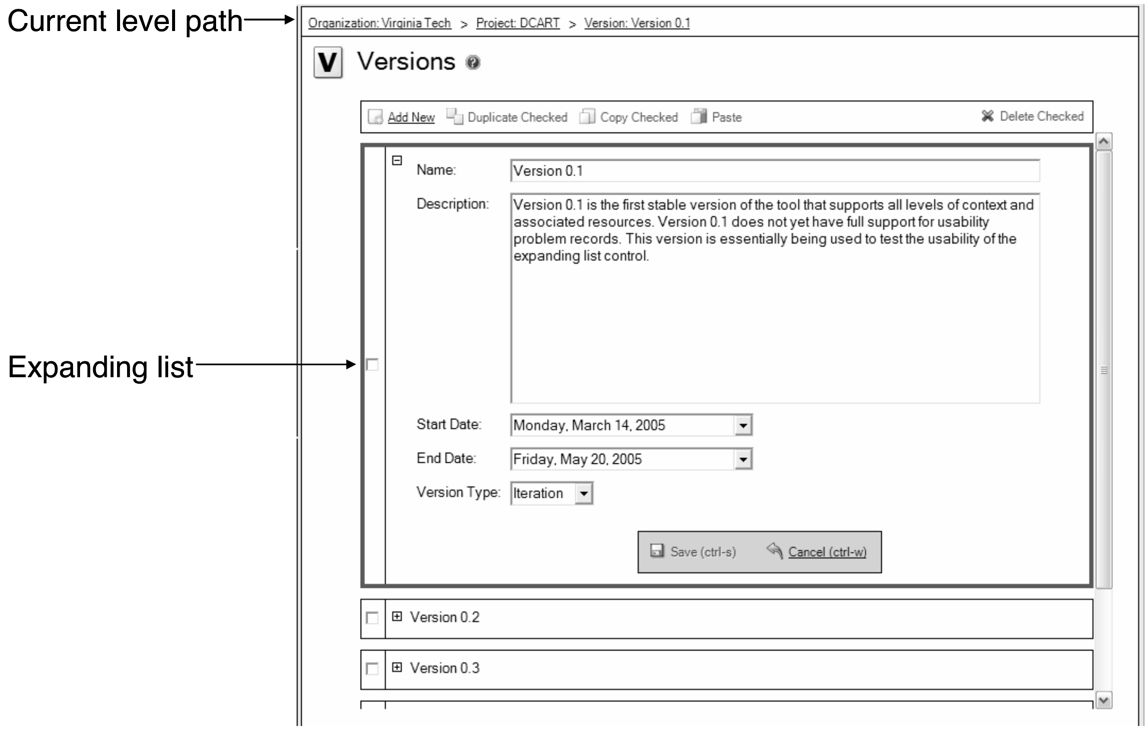
In Section 4.2.1.1, resources are described within the contexts in which they are used (Figure 6). In DCART, resources are used in these same contexts, but they are pooled at higher levels of context to facilitate reuse. For example, although benchmark tasks are used at the task run level, they are pooled at the project level, so that they can be reused for multiple task runs of multiple sessions of multiple versions of the project.

The resource view has two lists of resources. The top list contains resources that are pooled in the organization level: managers, participants, software developers, and usability practitioners. The bottom list contains resources that are pooled in the project level: product concept statements, user classes, usability goals, usability attributes, benchmark tasks, measured values, and usability specifications. To the left of each resource is an icon that is used to represent the resource in other parts of the application, such as in UP instance records. Like the level view, selecting a resource will display it in the workspace view.

The resources that are pooled in the selected level of context are made available in the resource view. In Figure 9, the DCART project level is selected, so the resources pooled in the Virginia Tech organization and DCART project levels are available. The organization resource pool is available because projects are nested inside of organizations, and selecting a project implies selecting its parent organization. If the Virginia Tech organization were selected in the level view, then only the organization resource pool would be active, and the project resource pool would be grayed out.

Figure 10 shows the workspace view when a version level is selected. The workspace view consists of two parts. The first part at the top of the view shows the path of levels to the current level. The second part is a control that I refer to as an expanding list. Records, individual rows in the expanding list, can be contracted and expanded. A contracted record, such as the record for Version 0.2, shows only the name of the level or resource that it contains. Clicking on the plus symbol or selecting the name text expands the record to show additional fields, as in Version 0.1. When a level is selected in the level view, it and all of its sibling levels are displayed in the expanding list. The selected level is initially expanded and the non-selected sibling levels are initially contracted. Resources exist in pools; selecting a pool of resources displays those resources in the expanding list. All the resources are initially contracted.

All the levels and resources can be edited in place inside of an expanded record. Figure 11 shows an edited version of Version 0.1. After a version has been edited, the background color changes to a salmon color, and the save link becomes active. If a record is saved, the background color turns back to white and the save link becomes grayed out. If the changes to a record are cancelled, the record is contracted. If changes have been made and another level or resource is selected, DCART prompts the user to save changes before loading the new expanding list.



Name: Version 0.1

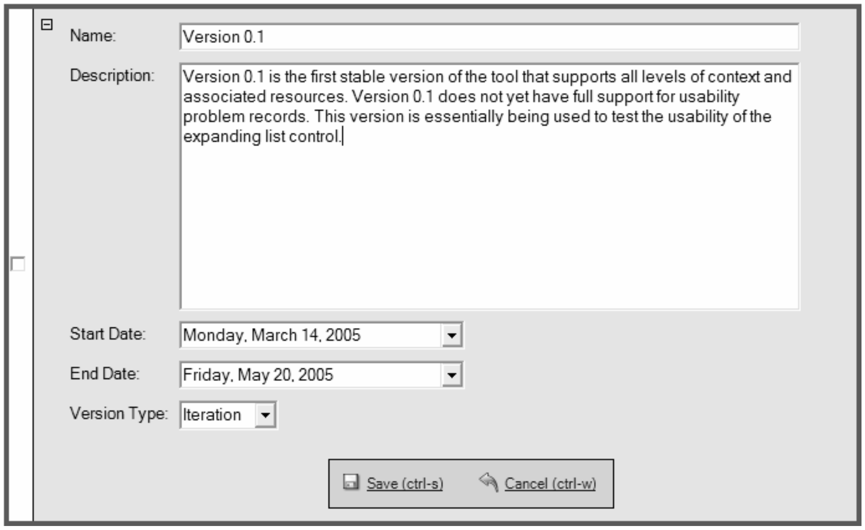
Description: Version 0.1 is the first stable version of the tool that supports all levels of context and associated resources. Version 0.1 does not yet have full support for usability problem records. This version is essentially being used to test the usability of the expanding list control.

Start Date: Monday, March 14, 2005

End Date: Friday, May 20, 2005

Version Type: Iteration

Figure 10: Workspace view



Name: Version 0.1

Description: Version 0.1 is the first stable version of the tool that supports all levels of context and associated resources. Version 0.1 does not yet have full support for usability problem records. This version is essentially being used to test the usability of the expanding list control.

Start Date: Monday, March 14, 2005

End Date: Friday, May 20, 2005

Version Type: Iteration

Figure 11: Edited expanding list record

In addition to editing the fields of individual records, the expanding list control can be used to modify records. Figure 12 shows the record modifications option bar at the top of the expanding list control. The “Add New” option is always active; selecting this option will create a new record in the list and expand it. When individual records are selected using the selection checkbox, the appropriate options on the option bar become active. For example, the record for Version 0.1 has been selected, so the “Duplicate Checked”, “Copy Checked”, and “Delete Checked” options are active. Selecting any of these options will perform the requested action using the selected record as the target. The “Duplicate Checked” and “Delete Checked” options work on multiple selected records.

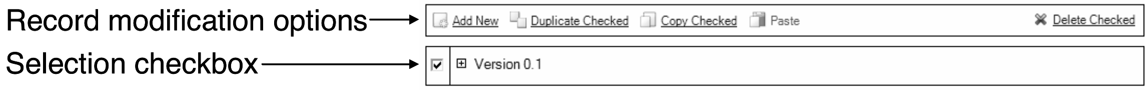


Figure 12: Modification options bar and selection checkbox

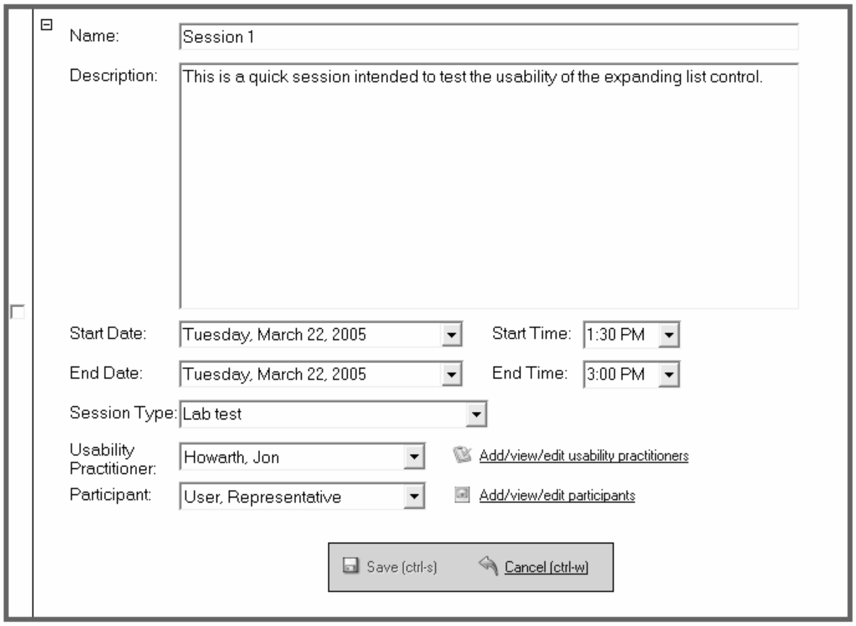
4.3.2.2 Support for Problem Report Format

UP instances are identified and recorded during usability sessions. A session can consist of a usability practitioner observing a participant or performing an inspection or an expert walkthrough. Session levels exist inside of a given organization, project, and version. Figure 13 shows a session record for a session that was run with a participant to evaluate DCART version 0.1. The usability practitioner and participant shown in the record are selected from the associated resource pool at the organization level.

During each session, participants perform a given number of tasks. Figure 14 shows a task run record for one of the tasks for the session in Figure 13. The user class, benchmark task, and usability specification shown in the record are selected from the associated resource pool at the project level.

The "Load a video" option allows evaluators to associate a video with the task run. It is particularly useful for post-hoc or detailed analyses of task runs.

Each task run consists of two steps: collecting UP instances in the form of UP instance records and reviewing the collected UP instance records to fill in necessary details. The evaluator does the first step of collecting UP instances while the participant is performing the task. When the participant is finished with the task, the evaluator reviews the UP instance records that he created and adds additional notes or observations that he did not have time to record during the running of the task.



Name: Session 1

Description: This is a quick session intended to test the usability of the expanding list control.

Start Date: Tuesday, March 22, 2005 Start Time: 1:30 PM

End Date: Tuesday, March 22, 2005 End Time: 3:00 PM

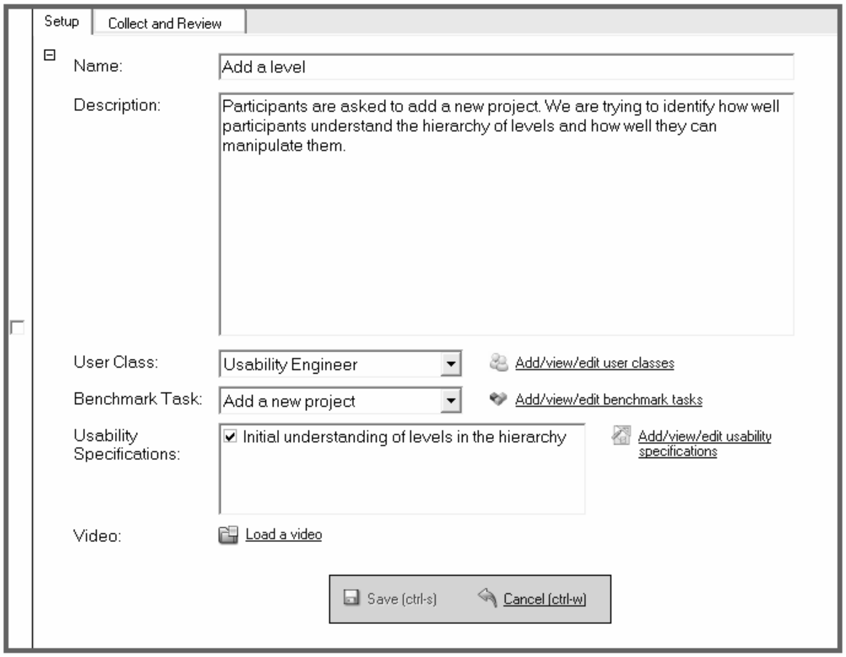
Session Type: Lab test

Usability Practitioner: Howarth, Jon

Participant: User, Representative

Add/view/edit usability practitioners
Add/view/edit participants
Save (ctrl-s)
Cancel (ctrl-w)

Figure 13: Session record



Name: Add a level

Description: Participants are asked to add a new project. We are trying to identify how well participants understand the hierarchy of levels and how well they can manipulate them.

User Class: Usability Engineer

Benchmark Task: Add a new project

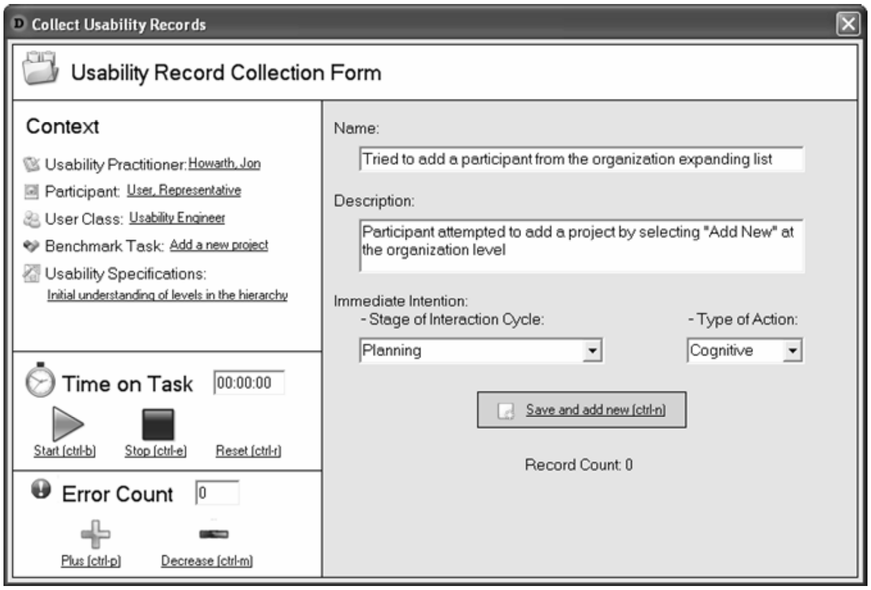
Usability Specifications: ☑ Initial understanding of levels in the hierarchy

Video:

Add/view/edit user classes
Add/view/edit benchmark tasks
Add/view/edit usability specifications
Load a video
Save (ctrl-s)
Cancel (ctrl-w)

Figure 14: Task run record

Figure 15 shows the first step; each task run has an option under the “Collect and Review” tab that starts a form, which is used to create UP instance records during the task run. If a video has been associated with the task run, a separate window displays the video. The form has four separate areas. The top left corner shows the context including the usability practitioner, participant, user class, benchmark task, and usability specifications. The evaluator can select any of these resources during the task run to see the resource’s full record in a separate window. Below the context area is an error counter that evaluators can use to tally errors committed by a participant; not all errors indicate usability problems. The time on task area below the error count area is a manual timer that evaluators can use to record the amount of time that a participant is actively involved in the task. The timer can be paused to account for interactions that are not part of the task, such as further explaining task instructions.


Collect Usability Records
Usability Record Collection Form
Context
Start (ctrl-b)
Stop (ctrl-e)
Reset (ctrl-r)
Plus (ctrl-p)
Decrease (ctrl-m)
Save and add new (ctrl-n)

Usability Practitioner: Howarth, Jon
Participant: User, Representative
User Class: Usability Engineer
Benchmark Task: Add a new project
Usability Specifications: Initial understanding of levels in the hierarchy

Time on Task 00:00:00

Error Count 0

Name: Tried to add a participant from the organization expanding list

Description: Participant attempted to add a project by selecting "Add New" at the organization level

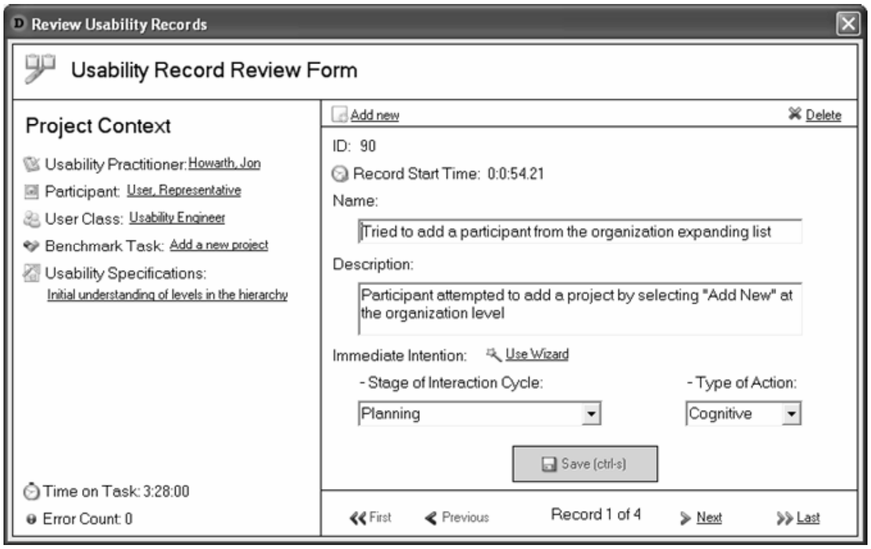
Immediate Intention:
- Stage of Interaction Cycle: Planning
- Type of Action: Cognitive

Record Count: 0

Figure 15: Usability problem instance collection form

The final area on the right is the UP instance collection form. It is designed to allow evaluators to quickly create UP instance records as participants experience UPs during the task run and contains the basic fields needed to capture the essence of a UP. During a task run, the evaluator uses the ctrl-n hotkey combination or the “Save and add new (ctrl-n)” link to create a new UP instance record for each instance of a UP encountered by the participant. The evaluator gives each UP instance record a name and a brief description. The evaluator can also assign an immediate intention (Section 4.2.2.2.1).

After the participant has performed the task, DCART displays a brief summary of the task under the “Collect and Review” tab of the trial record. The evaluator then selects an option that opens a form used for the second step of reviewing the collected UP instance records (Figure 16). The form is similar to the form used to collect UP instances except that it provides a way for evaluators to iterate through the collection of UP instance records. The evaluator uses this form to review UP instance descriptions and fill in details. If a video has been associated with the task run, the video is synched to the timestamp of the currently displayed UP instance, so that evaluators can easily review the instance.


Review Usability Records

Usability Record Review Form

Project Context

Usability Practitioner: Howarth, Jon

Participant: User, Representative

User Class: Usability Engineer

Benchmark Task: Add a new project

Usability Specifications: Initial understanding of levels in the hierarchy

Time on Task: 3:28:00

Error Count: 0

ID: 90

Record Start Time: 0:0:54.21

Name: Tried to add a participant from the organization expanding list

Description: Participant attempted to add a project by selecting "Add New" at the organization level

Immediate Intention:

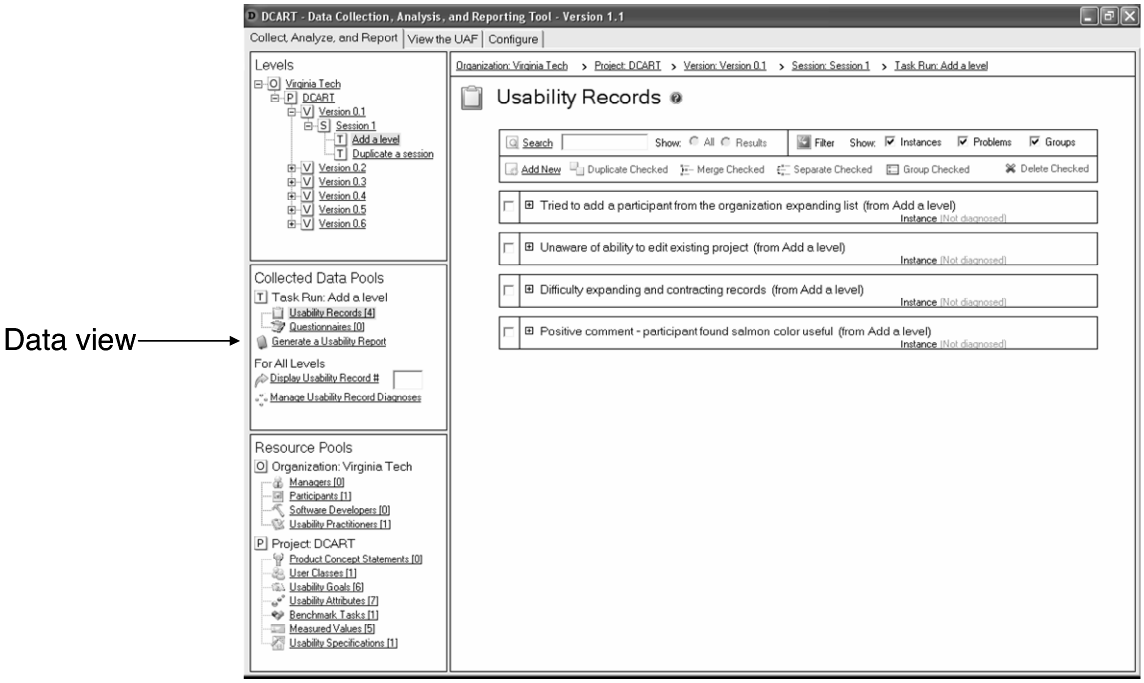
- Stage of Interaction Cycle: Planning

- Type of Action: Cognitive

Record 1 of 4

Figure 16: Usability problem instance review form

The UP instance records created during the task run are made accessible through the data view shown in Figure 17. When a task run is selected in the level view and the “Usability Records” option is selected in the data view, all the usability records associated with the task run are displayed in the workspace view. The usability records are shown in an expanding list. They can be edited or modified in the same manner as level or resource records. The usability records shown for the “Add a level” task run all have the text “Instance” in the right hand side of their records to indicate that they are UP instances.



□ ⊞ Tried to add a participant from the organization expanding list (from Add a level)
Instance (Not diagnosed)

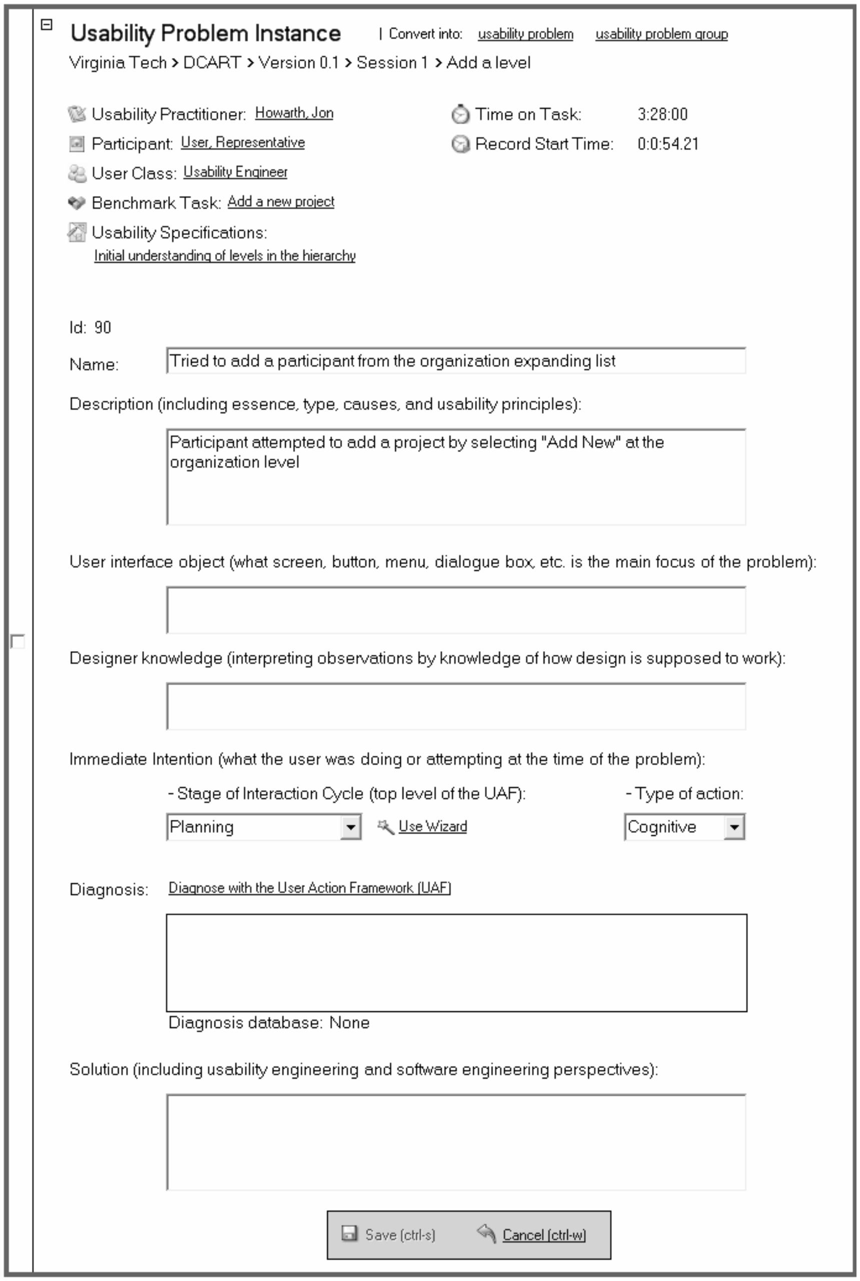
□ ⊞ Unaware of ability to edit existing project (from Add a level)
Instance (Not diagnosed)

□ ⊞ Difficulty expanding and contracting records (from Add a level)
Instance (Not diagnosed)

□ ⊞ Positive comment - participant found salmon color useful (from Add a level)
Instance (Not diagnosed)

Figure 17: Data view

Figure 18 shows an expanded UP instance record. The section at the top contains all the context information that appeared in the UP instance collection and record review forms as well as information about the time at which the UP was encountered in the task run. The remainder of the record contains a number of fields that can be used to describe and specify the UP instance. Each UP instance record is assigned a unique id. The information entered in the UP instance collection and record review forms is included in the name, description, and immediate intention fields of the UP instance record. The other fields in the UP instance record are filled out after the participant has left and are used to document the user interface object or objects associated with the UP instance, designer knowledge about how the design should work, immediate intention, UAF diagnosis, and solution suggestions.



Usability Problem Instance | Convert into: usability problem usability problem group

Virginia Tech > DCART > Version 0.1 > Session 1 > Add a level

Usability Practitioner: Howarth, Jon

Participant: User, Representative

User Class: Usability Engineer

Benchmark Task: Add a new project

Usability Specifications: Initial understanding of levels in the hierarchy

Time on Task: 3:28:00

Record Start Time: 0:0:54.21

Id: 90

Name: Tried to add a participant from the organization expanding list

Description (including essence, type, causes, and usability principles):

Participant attempted to add a project by selecting "Add New" at the organization level

User interface object (what screen, button, menu, dialogue box, etc. is the main focus of the problem):

Designer knowledge (interpreting observations by knowledge of how design is supposed to work):

Immediate Intention (what the user was doing or attempting at the time of the problem):

- Stage of Interaction Cycle (top level of the UAF): Planning

- Type of action: Cognitive

Diagnosis: Diagnose with the User Action Framework (UAF)

Diagnosis database: None

Solution (including usability engineering and software engineering perspectives):

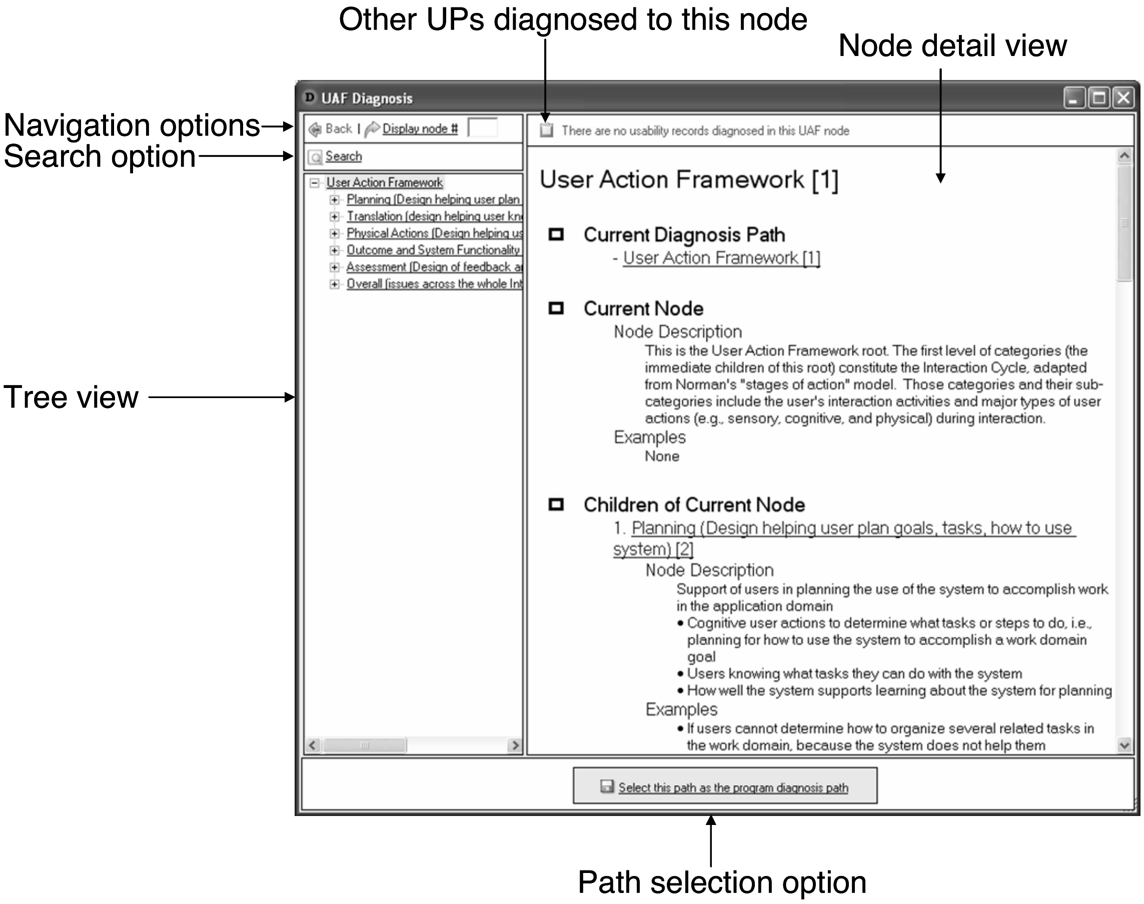
Figure 18: UP record

4.3.3 Support for Diagnosis

My proposed feature for diagnosis in Section 4.2.2 involves the use of a conceptual framework of usability concepts, the UAF, to diagnose UPs. I first discuss DCART’s support for full diagnosis with the UAF. Thereafter, I describe how DCART supports micro-iteration to help evaluators capture immediate intention for partial diagnosis.

4.3.3.1 Support for Full Diagnosis

As described in Section 2.3, diagnosis involves associating a UP (as described in a UP instance record or UP description) with the correct usability concept that describes the cause within the interaction design. The UAF is built into DCART; selecting the “Diagnose with the User Action Framework (UAF)” option inside of UP records (Figure 18), opens the UAF diagnosis form in a new window (Figure 19).


Other UPs diagnosed to this node
Node detail view
Navigation options
Search option
Tree view
Path selection option
UAF Diagnosis
Back | Display node #
Search
User Action Framework
Planning (Design helping user plan
Translation (design helping user kn
Physical Actions (Design helping us
Outcome and System Functionality
Assessment (Design of feedback a
Overall (issues across the whole Int
Select this path as the program diagnosis path

□ There are no usability records diagnosed in this UAF node

User Action Framework [1]

□ Current Diagnosis Path
- User Action Framework [1]

□ Current Node
Node Description
This is the User Action Framework root. The first level of categories (the immediate children of this root) constitute the Interaction Cycle, adapted from Norman's "stages of action" model. Those categories and their sub-categories include the user's interaction activities and major types of user actions (e.g., sensory, cognitive, and physical) during interaction.
Examples
None

□ Children of Current Node
1. Planning (Design helping user plan goals, tasks, how to use system) [2]
Node Description
Support of users in planning the 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 the system supports learning about the system for planning

Examples
- If users cannot determine how to organize several related tasks in the work domain, because the system does not help them

Figure 19: UAF Diagnosis form

Figure 19 shows the four major areas of the UAF diagnosis form. The top left corner contains navigation options to allow an evaluator to go back or jump directly to a node with a given node number. Below the navigation options is a search mechanism that allows an evaluator to find all tree nodes that contain a given search string. The results of the search are displayed in a new window. The tree view on the left-hand side allows practitioners who are familiar with the

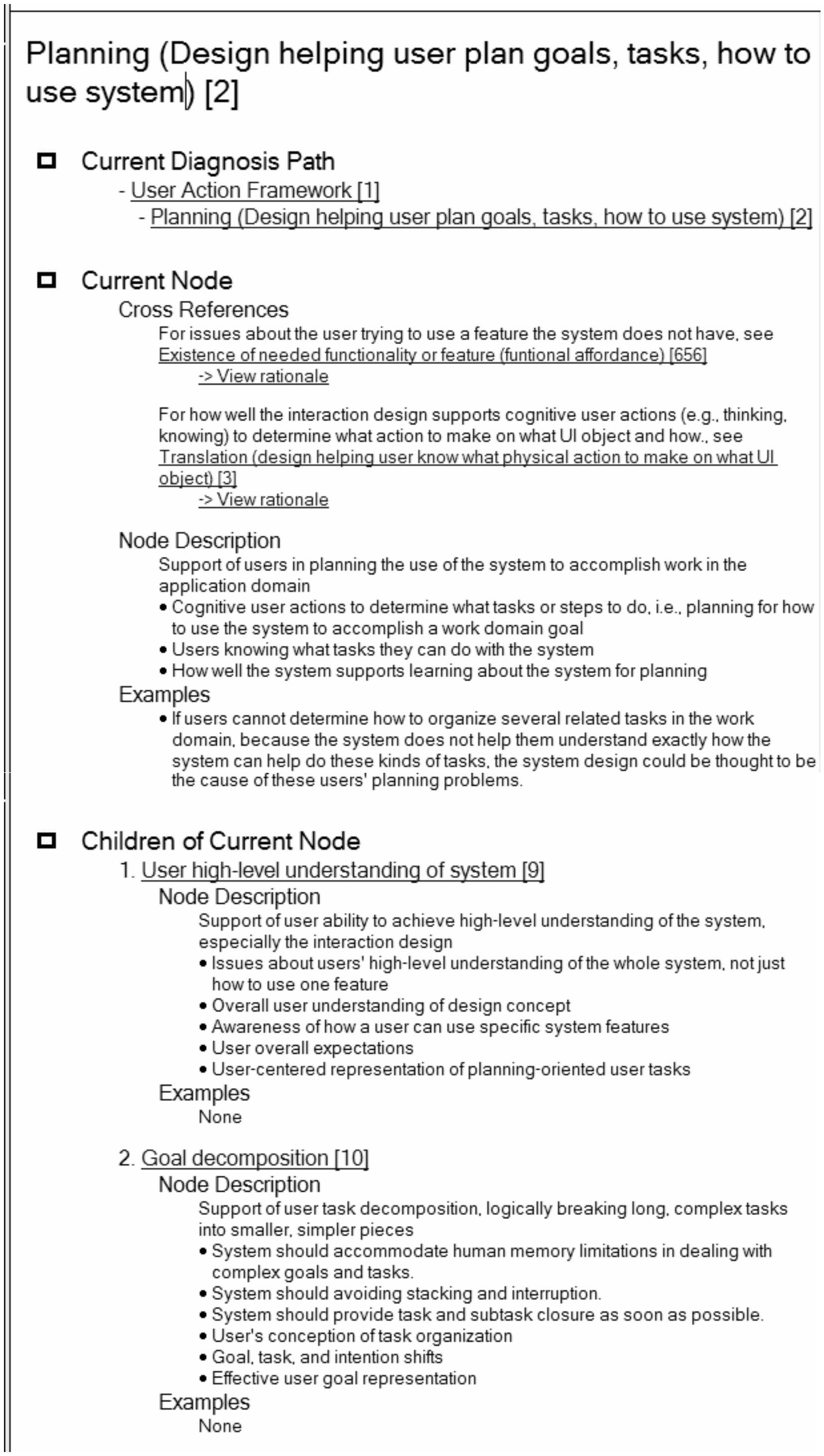
UAF to quickly traverse it. Practitioners that are not familiar with the UAF can traverse the tree using the node detail view. The tree is modeled after the Windows Explorer tree view and uses minus signs for expanded nodes and plus signs for expanding nodes with children. Selecting the link for a node 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 top of the node detail view displays the number of other usability records that have been diagnosed to the displayed node. Clicking on this number will open a new window that contains a listing of these records. The path selection option below the tree view allows evaluators to select the current path as the diagnosis path for their UP. When a path has been selected, the window closes and the path is inserted into the usability record.

Figure 20 shows the node detail view. The first item is the name of the node; the node's unique id is displayed in brackets at the end of the node's name. Below the name is a representation of the current diagnosis path in a horizontal tree similar to the tree view. The next item is the current node, which contains cross references, a node description, and examples. The final item is a listing of children of the current node. The Planning node shown in Figure 20 actually has eight children, but the screenshot is limited to two children.

In the current node item, cross references appear first to immediately redirect evaluators who have incorrectly arrived at the 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 current node's relationship to the cross referenced node and tells the practitioner why the cross referenced node may better describe the UP. The rationale is hidden and must be displayed with the "View rationale" option to limit the amount of information that a practitioner must initially process.

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 when the current node is displayed in the listing of children for its parent node. The look-ahead description bullet helps to guide practitioners down a particular path to a node. The examples are descriptions of UPs that would be classified in the node. Like description bullets, an example may be classified as a look-ahead example.

The children of the current node appear after the current node item. Each child is displayed with the high-level description and description bullets, including any look-ahead description bullets from its children. Examples are not displayed with the children to minimize display space.



Planning (Design helping user plan goals, tasks, how to use system) [2]

□ Current Diagnosis Path

- User Action Framework [1]
  - Planning (Design helping user plan goals, tasks, how to use system) [2]

□ Current Node

Cross References

For issues about the user trying to use a feature the system does not have, see Existence of needed functionality or feature (funtional affordance) [656]
-> View rationale

For how well the interaction design supports cognitive user actions (e.g., thinking, knowing) to determine what action to make on what UI object and how., see Translation (design helping user know what physical action to make on what UI object) [3]
-> View rationale

Node Description

Support of users in planning the 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 the system supports learning about the system for planning

Examples

- If users cannot determine how to organize several related tasks in the work domain, because the system does not help them understand exactly how the system can help do these kinds of tasks, the system design could be thought to be the cause of these users' planning problems.

□ Children of Current Node

1. User high-level understanding of system [9]

Node Description

Support of user ability to achieve high-level understanding of the system, especially the interaction design

- Issues about users' high-level understanding of the whole system, not just how to use one feature
- Overall user understanding of design concept
- Awareness of how a user can use specific system features
- User overall expectations
- User-centered representation of planning-oriented user tasks

Examples

None

2. Goal decomposition [10]

Node Description

Support of user task decomposition, logically breaking long, complex tasks into smaller, simpler pieces

- System should accommodate human memory limitations in dealing with complex goals and tasks.
- System should avoiding stacking and interruption.
- System should provide task and subtask closure as soon as possible.
- User's conception of task organization
- Goal, task, and intention shifts
- Effective user goal representation

Examples

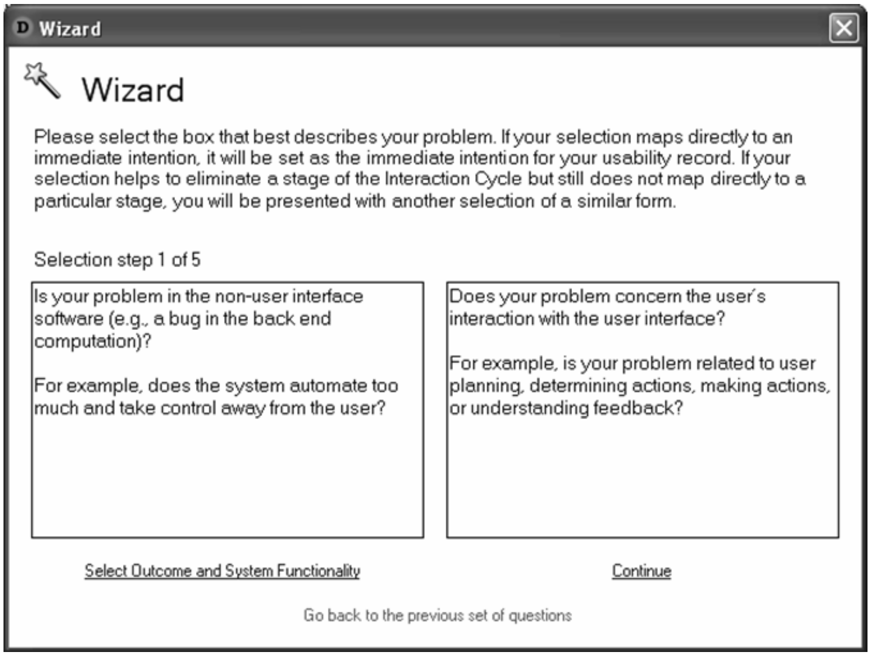
None

Figure 20: Node detail view

4.3.3.2 Support for Partial Diagnosis

As discussed in Section 4.2.2.2, full diagnosis with the UAF may be time consuming. In this section, I discuss how DCART supports micro-iteration to capture immediate intention and describe an implementation of the Wizard.

DCART provides support for micro-iteration and immediate intention through a two step process for identifying and recording UP instances that is described in Section 4.3.2.2. During the first step, the evaluator observes the participant and creates UP instance records using the usability record collection form. The form has fields for immediate intention information (Figure 15). If the evaluator is unsure of the immediate intention, he leaves the field blank. During the second step, the evaluator reviews the UP instance records while the participant is still available and asks questions of the participant to elicit necessary information to determine immediate intention. The usability record review form (Figure 16) has a “Use Wizard” option that opens the Wizard in a new window (Figure 21).



Please select the box that best describes your problem. If your selection maps directly to an immediate intention, it will be set as the immediate intention for your usability record. If your selection helps to eliminate a stage of the Interaction Cycle but still does not map directly to a particular stage, you will be presented with another selection of a similar form.

Selection step 1 of 5

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?

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?

Select Outcome and System Functionality
Continue
Go back to the previous set of questions

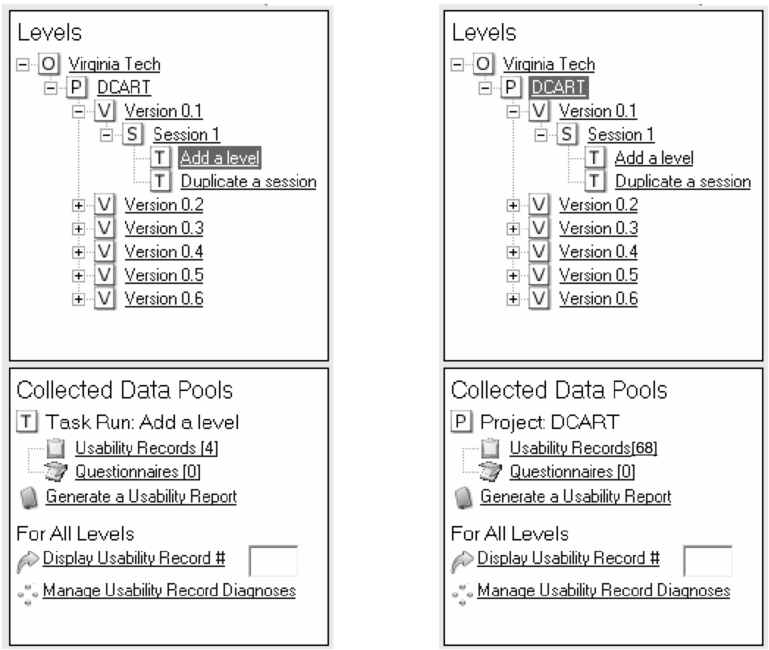
Figure 21: Wizard

4.3.4 Support for Merging and Grouping

As described in Section 4.2.3, merging involves combining UP instances and grouping involves associating UPs. Support for merging and grouping in DCART uses a form of scoping based on hierarchical context. In this section, I first describe the scoping functionality and then describe merging and grouping support in DCART.

4.3.4.1 Scoping Usability Problem Instance Selection

Although UP instance records are created and edited at the task run level, they can be viewed at higher levels. Figure 22 shows the level and data views when a task run is selected (left) and a project is selected (right). In the data view, the number in square brackets after the “Usability Records” option indicates the number of usability records associated with the selected level. The individual task run has four usability records associated with it while the project level has 68 usability records associated with it. The project level includes all UP instance records from all task runs of all sessions of all versions of the project. Selecting a level essentially scopes the selection of UP instance records.



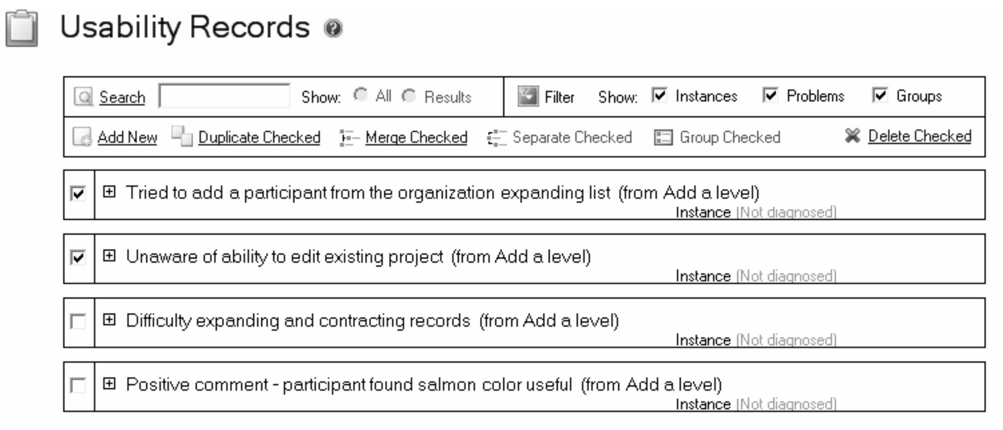
```
Levels
⊟ O Virginia Tech
    ⊟ P DCART
        ⊟ V Version 0.1
            ⊟ S Session 1
                T Add a level
                T Duplicate a session
        ⊞ V Version 0.2
        ⊞ V Version 0.3
        ⊞ V Version 0.4
        ⊞ V Version 0.5
        ⊞ V Version 0.6

Collected Data Pools
T Task Run: Add a level
    Usability Records [4]
    Questionnaires [0]
Generate a Usability Report

For All Levels
Display Usability Record #
Manage Usability Record Diagnoses
```

```
Levels
⊟ O Virginia Tech
    ⊟ P DCART
        ⊟ V Version 0.1
            ⊟ S Session 1
                T Add a level
                T Duplicate a session
        ⊞ V Version 0.2
        ⊞ V Version 0.3
        ⊞ V Version 0.4
        ⊞ V Version 0.5
        ⊞ V Version 0.6

Collected Data Pools
P Project: DCART
    Usability Records[68]
    Questionnaires [0]
Generate a Usability Report

For All Levels
Display Usability Record #
Manage Usability Record Diagnoses
```

Figure 22: Usability problem records at the task run level (left) and the project level (right)

4.3.4.2 Support for Merging Usability Problem Instances

Checking two or more UP instance records and selecting the “Merge Checked” option creates a new UP record as shown in Figure 23 and Figure 24. The newly created UP record has the text “Problem[2]” to indicate that it is a UP record composed of two UP instance records. A UP record can be separated back into its constituent UP instance records by checking it and selecting the “Separate Checked” option. In addition, the options at the top right of the record modifications bar allow for filtering the view to just show certain types of usability records.



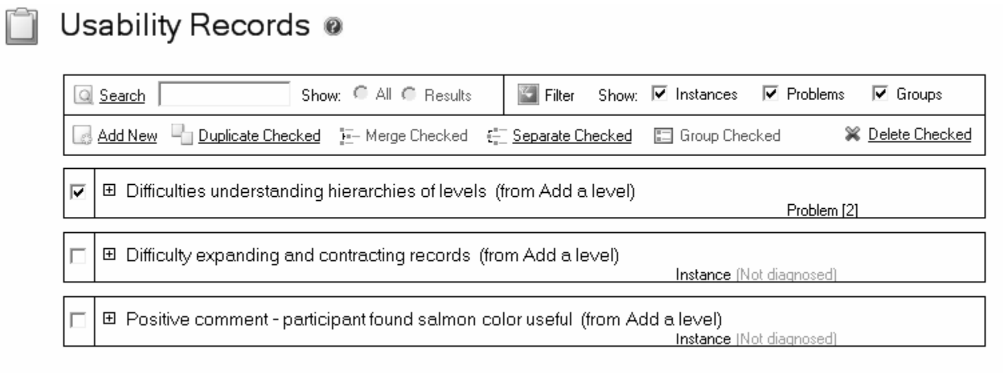
☑ ⊞ Tried to add a participant from the organization expanding list (from Add a level) Instance (Not diagnosed)

☑ ⊞ Unaware of ability to edit existing project (from Add a level) Instance (Not diagnosed)

□ ⊞ Difficulty expanding and contracting records (from Add a level) Instance (Not diagnosed)

□ ⊞ Positive comment - participant found salmon color useful (from Add a level) Instance (Not diagnosed)

Figure 23: “Merge Checked” and “Separate Checked” options



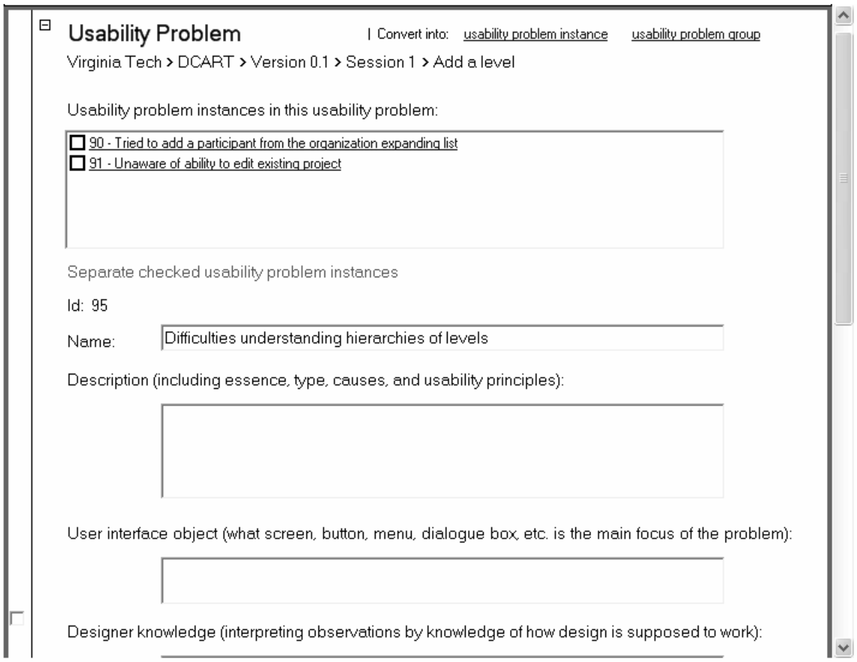
☑ ⊞ Difficulties understanding hierarchies of levels (from Add a level) Problem [2]

□ ⊞ Difficulty expanding and contracting records (from Add a level) Instance (Not diagnosed)

□ ⊞ Positive comment - participant found salmon color useful (from Add a level) Instance (Not diagnosed)

Figure 24: An expanding list that contains two usability problem instance records and one usability problem record

The expanded record for a UP will differ from the expanded record for a UP instance in one way. The context information at the top of the UP instance record is not appropriate for UP records because a UP record may be composed of UP instance records from different contexts. For example a UP record may be created at the version level from UP records from different sessions. The top of a UP record will instead contain the list of UP instance records included in it (Figure 25); individual UP instance records can be removed from the UP record by checking them and selecting the “Separate checked usability problem instances” option. The additional fields will remain the same because UP records have the same properties as UP instance records, such as associated designer knowledge.



Usability Problem | Convert into: usability problem instance usability problem group

Virginia Tech > DCART > Version 0.1 > Session 1 > Add a level

Usability problem instances in this usability problem:

□ 90 - Tried to add a participant from the organization expanding list
□ 91 - Unaware of ability to edit existing project

Separate checked usability problem instances

Id: 95

Name: Difficulties understanding hierarchies of levels

Description (including essence, type, causes, and usability principles):

User interface object (what screen, button, menu, dialogue box, etc. is the main focus of the problem):

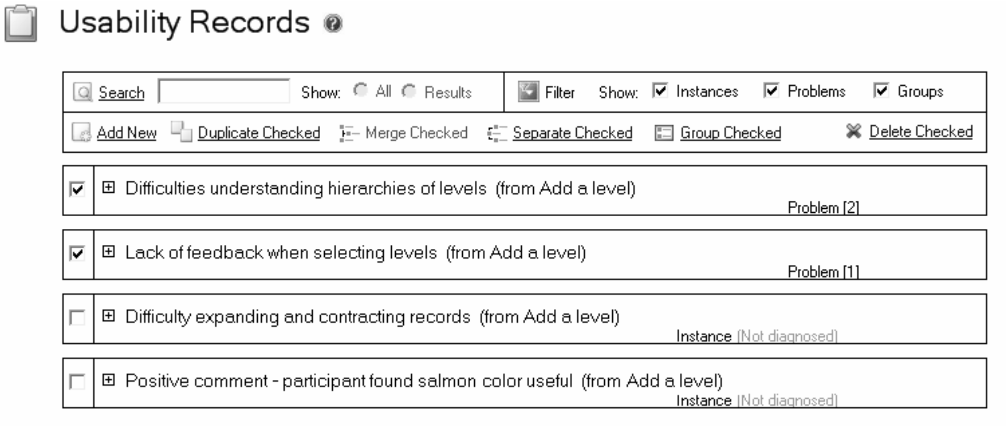
Designer knowledge (interpreting observations by knowledge of how design is supposed to work):

Figure 25: A usability problem record

The scoping functionality also applies to UP records. For efforts aimed at increasing the usability of a given version, evaluators can compare UP records among trials and sessions. To study trends over the life of a project, evaluators can compare UP records from different versions. Finally to assess the effectiveness of a given usability process, evaluators can work with UP records across projects.

4.3.4.3 Support for Grouping Usability Problems

Checking two or more UP records and selecting the “Group Checked” option will create a new group of the UP records (Figure 26). Whereas the “Merge Checked” option will combine two or more UP instance records into one UP record and remove them from the list, the “Group Checked” option will create a new group record and leave the UP records that are involved in the list as shown in Figure 27. The group can be deleted like any other record using the “Delete Checked” option.



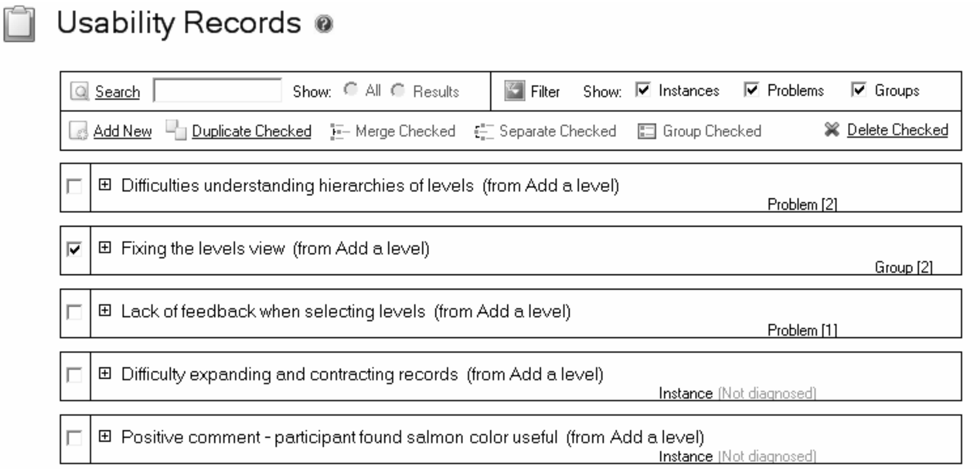
☑ ⊞ Difficulties understanding hierarchies of levels (from Add a level) Problem [2]

☑ ⊞ Lack of feedback when selecting levels (from Add a level) Problem [1]

□ ⊞ Difficulty expanding and contracting records (from Add a level) Instance (Not diagnosed)

□ ⊞ Positive comment - participant found salmon color useful (from Add a level) Instance (Not diagnosed)

Figure 26: Grouping usability problems



□ ⊞ Difficulties understanding hierarchies of levels (from Add a level) Problem [2]

☑ ⊞ Fixing the levels view (from Add a level) Group [2]

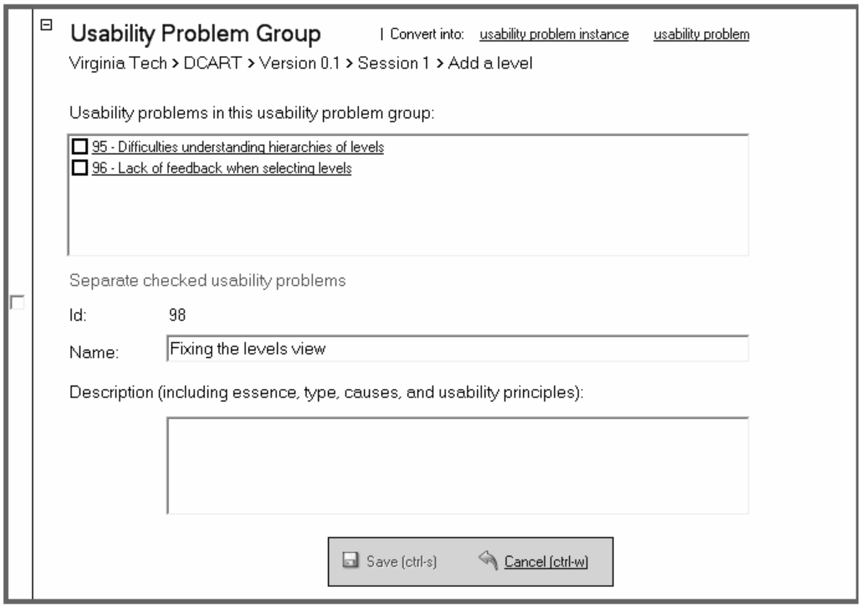
□ ⊞ Lack of feedback when selecting levels (from Add a level) Problem [1]

□ ⊞ Difficulty expanding and contracting records (from Add a level) Instance (Not diagnosed)

□ ⊞ Positive comment - participant found salmon color useful (from Add a level) Instance (Not diagnosed)

Figure 27: An expanding list that contains a group record

The expanded record for a group will be similar to the expanded record for a UP record in that it will have a list of the associated UP records at the top (see Figure 28). The expanded record for a group will be different in that it will only contain name and description fields.



Usability Problem Group

| Convert into: usability problem instance usability problem

Virginia Tech > DCART > Version 0.1 > Session 1 > Add a level

Usability problems in this usability problem group:

☐ 95 - Difficulties understanding hierarchies of levels
☐ 96 - Lack of feedback when selecting levels

Separate checked usability problems

Id: 98

Name: Fixing the levels view

Description (including essence, type, causes, and usability principles):

Figure 28: A group record

5 Evaluation of Desirable Tool Features

This section is intended to answer research questions RQ3a, RQ3b, and RQ3c (Section 1.5). I describe studies to evaluate each of the desirable features with respect to the effectiveness of evaluators, with a specific focus on novice evaluators. The IRB approval document for the studies is in Appendix A.

5.1 Study 1: Support for Usability Problem Instance Records

Using paper and existing UE tools, evaluators write notes and comments during a lab-based usability evaluation and manually review and relate them to identify instances of UPs. Using the approach described in Section 4.2.1, evaluators create UP instance records during the evaluation. This approach allows evaluators to work with UP data at a relatively abstract level and removes the need for a second pass through the data to consolidate raw usability data in the form of comments into UP instances. Figure 5 shows the difference between the two levels of UP data in terms of a timeline of an example session with a participant.

This study compared the lists of UP instances produced by evaluators with and without explicit support for UP instance records. Evaluator effectiveness was of primary interest for this study. Because I assumed a fixed-resources environment, I wanted to remove efficiency as a point of consideration. To confirm this operating assumption, I recorded the amount of time that it took evaluators to perform the evaluations. Of interest with regard to effectiveness were measures of UP instance discovery and quality as rated by usability practitioners.

Figure 29 is an overview of study 1; it shows roles and the tools and objects that people in the roles interacted with and produced. This figure is referenced in future sections that describe the various roles in more detail.

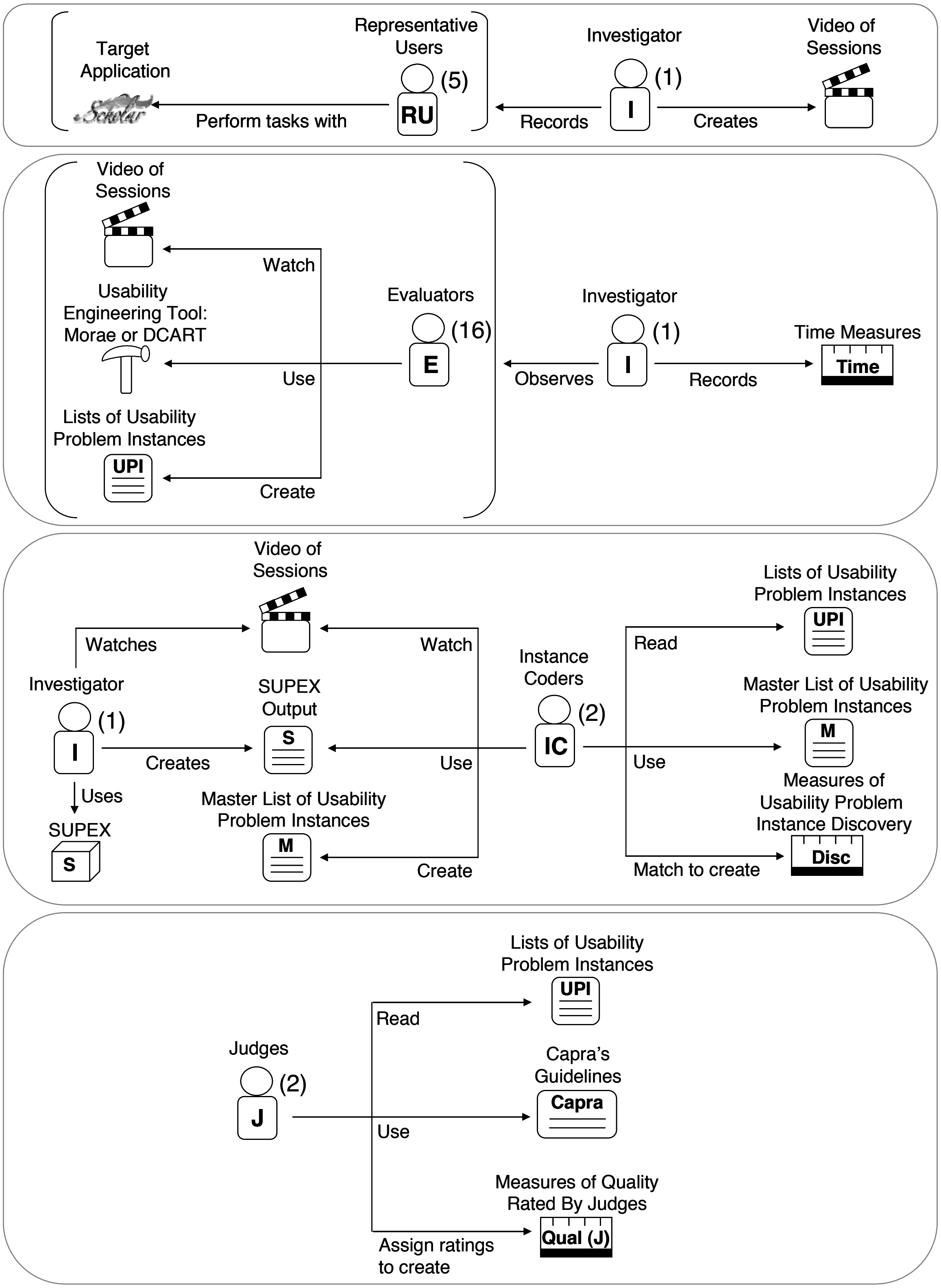


Figure 29: An overview of study 1. The numbers in parentheses indicate the number of individuals in each role.

5.1.1 Research Question and Hypotheses

The research question addressed by this study is directly related to RG3 in Section 1.5.

- RQ3a – How does tool support for UP instance records affect the effectiveness of novice evaluators?

The following experimental hypotheses apply to RQ3a:

- Hypothesis 3a.1 (H3a.1) – Tool support for UP instance records will not affect the time that it takes novice evaluators to perform evaluations.
- Hypothesis 3a.2 (H3a.2) – Tool support for UP instance records will increase the UP instance discovery of novice evaluators.
- Hypothesis 3a.3 (H3a.3) – Tool support for UP instance records will increase the quality of novice evaluators' descriptions of UP instances as rated by usability practitioners (judges in this study).

5.1.2 Method

5.1.2.1 Overview

The participants in this study watched videos of representative users performing tasks with Scholar, a course management system. These participants, whom I refer to as evaluators, produced lists of UP instances using one of two usability engineering tools: Morae or DCART. Morae did not have support for UP instance records; DCART did. I recorded time data while the evaluators created their lists of UP instances. Individuals, whom I refer to as instance coders, compared the UP instances to a master list of UP instances to create measures of UP instance discovery. Individuals with usability experience, whom I refer to as judges, rated the lists of UP instances from the perspective of a usability practitioner to create measures of quality.

5.1.2.2 Participants

As mentioned in the overview for this study, the participants are referred to as evaluators. Sixteen evaluators participated in this study. All the evaluators were Virginia Tech graduate students with one or more of the following qualifications:

- Had taken or were taking a usability engineering course
- Had taken or were taking a human-computer interaction (HCI) course
- Had research experience related to usability engineering

Additionally, all the evaluators selected for the study had less than one year of job experience related to usability engineering, thereby qualifying them as novices.

Thirteen of the evaluators were students in the Department of Computer Science and 3 were students in the Department of Industrial and Systems Engineering. Twelve had experience with course management systems, but none had ever used Scholar, the course management system used in the study.

Evaluators were recruited from three mailing lists at Virginia Tech, one for computer science graduate students, one for HCI students, and one for human factors students. The recruitment email message is in Appendix B.1. I paid each evaluator a fixed fee of $25.

5.1.2.3 Materials and Equipment

5.1.2.3.1 Target Application

Evaluators in this study watched a video of sessions of representative users performing tasks in Scholar (Section 5.1.2.3.2), a course management system. In the context of this study, I refer to Scholar as the target application. Scholar is an integrated learning, collaboration, and research support system. It is Virginia Tech's adaptation of a larger open-source project called Sakai (http://www.sakaiproject.org/).

I selected Scholar as the target application because it met two important criteria:

1. The individuals involved in the study are familiar with the domain addressed by Scholar (university course management).
2. The developers of the application were willing to review the usability evaluation reports produced by evaluators and participate in interviews regarding the content and usefulness of these reports.

5.1.2.3.2 Video of Representative User Sessions

I worked with the developers of Scholar to develop a list of 17 common tasks for the application. I recorded the screen video and audio of representative users as they performed these tasks. Figure 30 is an excerpt from Figure 29 that shows only the representative user role.

I recruited representative users by sending emails (Appendix C.1) to professional contacts. I asked the individuals who responded to complete a background survey (Appendix C.2). I selected five representative users based on the information provided in the background survey. The five representative users were graduate students in the following departments: Computer Science, Crop and Soil Environmental Sciences, Physics, Public and International Affairs, and Veterinary Medicine. The representative users signed a consent form to allow for the creation of audio and screen recordings of their sessions (Appendix C.3). Each representative user performed as many of the 17 tasks (Appendix C.4) as possible in 2 hours. 4 of the representative users completed all 17 tasks; one representative user only completed 13 tasks. I conducted the sessions in the McBryde 102 usability lab (Section 5.1.2.3.4). I paid each representative user a fixed fee of $20.

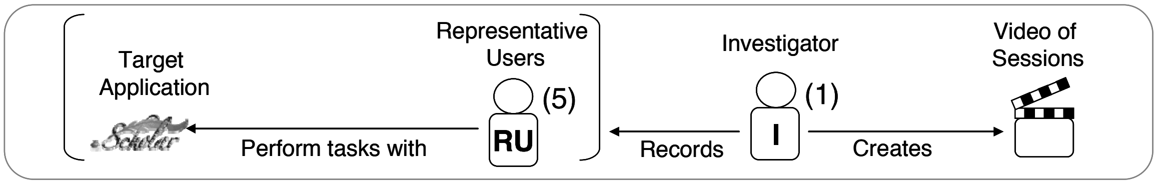


Figure 30: An excerpt from Figure 29 of the representative user role. The numbers in parentheses indicate the number of individuals in each role.

5.1.2.3.3 Usability Engineering Tools

Evaluators used two different UE tools in this study. Morae (Section 2.2.3.5) is a widely used UE tool; it was used as the representative tool for UE tools without explicit support for UP instance records. DCART (Section 4.3) served as the tool with explicit support for UP instance records.

5.1.2.3.4 Usability Lab

I conducted the study in the McBryde 102 usability lab. Figure 31 shows images of the lab.

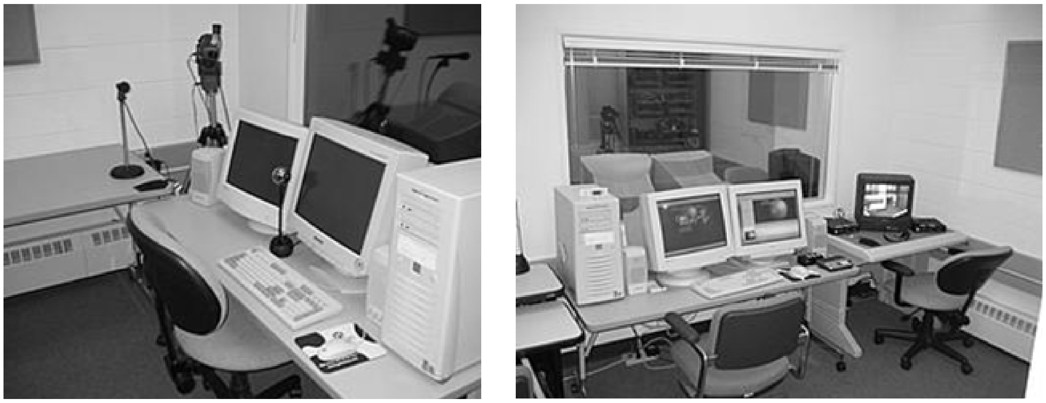


Figure 31: Images of the McBryde 102 usability lab. The left image is the setup used to record a session with a participant in 102 A. The right image is the setup used by the facilitator to monitor a session with a participant and work with the data captured during a session in 102 B.

Figure 32 is a diagram of the devices and connections in the McBryde 102 usability lab.

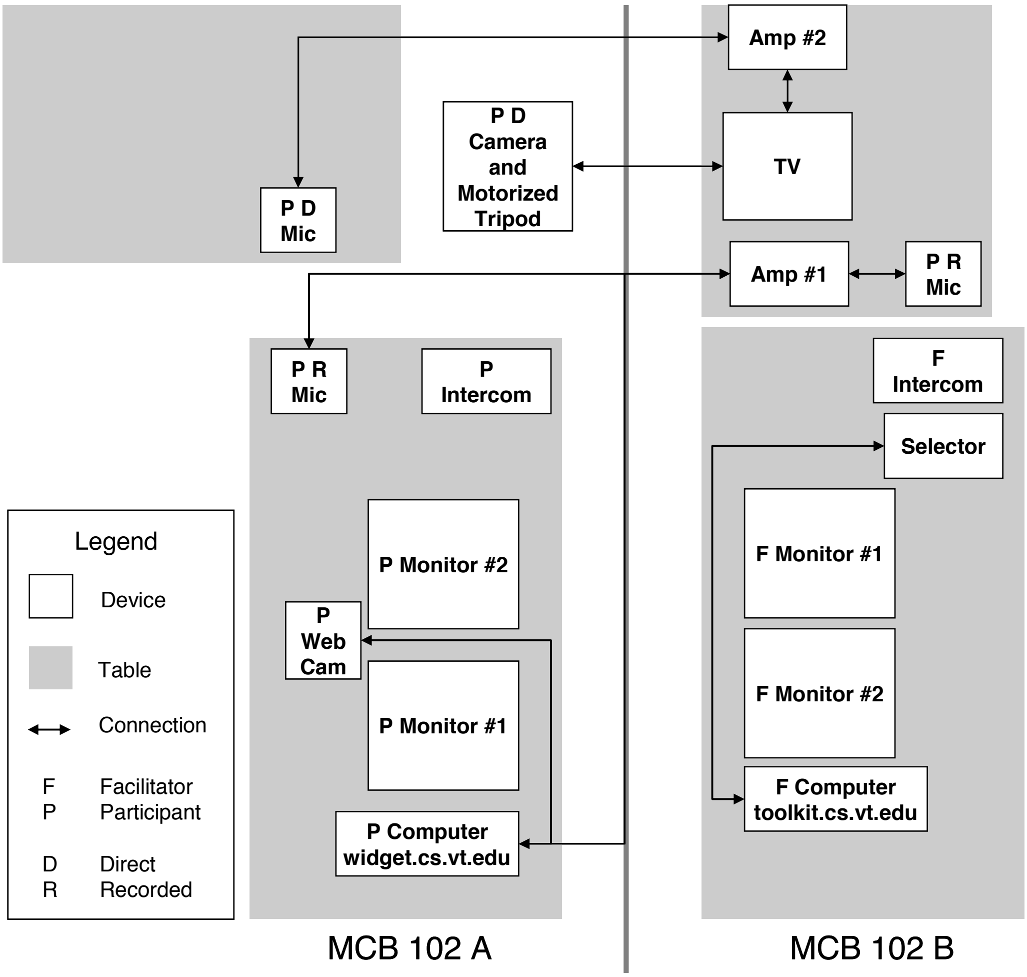


Figure 32: Diagram of the devices and connections in the McBryde 102 usability lab

There are two audio-video feeds coming from the participant: a direct feed that is not recorded and a recorded feed. The components of the direct feed are marked with the letter D in the figure. The direct feed is not recorded; it is set up to provide the facilitator with a way to monitor the participant in real time. The audio for the direct feed comes from an omnidirectional microphone that has been mounted on a stand in front of the participant. The audio is routed through Amp #2 to the TV. The facilitator can listen either by playing the audio through the TV's speakers or by plugging a headset into the TV. The video for the direct feed is provided by a camera mounted on a motorized tripod. There is a control located near Amp #2 that the facilitator can use to pan the camera. The camera's output is sent to the TV.

The components of the recorded feed are marked with the letter R in the figure. The recorded feed is the audio that will be recorded by the usability engineering software tool. The recorded microphones are omnidirectional table microphones. Both the participant's audio and the facilitator's audio are routed through Amp #1. They are combined using stereo audio, so that the participant's input is the right channel and the facilitator's input is the left channel. The combined audio is then routed back to the participant's computer, so that it can be recorded with Morae.

During playback, the facilitator can choose among the participant's audio (right channel), the facilitator's audio (left channel), or both at the same time by using the selector.

The intercoms are provided to allow the facilitator to communicate with the participant. The participant's intercom is intentionally located at the back of the workspace in 102 A (and therefore out of reach of the participant) because the participant will not need to use it to communicate with the facilitator. The facilitator can listen to the participant through the direct feed. The facilitator must push the button on the intercom to speak with the participant. This allows the facilitator to take verbal notes or talk with other facilitators about the participant's performance without being heard by the participant.

5.1.2.4 Procedure

I filtered evaluators and placed them into one of two treatment conditions via a background survey (Section 5.1.2.5, Appendix B.2). In one treatment, evaluators used Morae to conduct a usability evaluation; in the other treatment, evaluators used DCART to conduct a usability evaluation. I notified evaluators who had been selected to participate in the study via email and had them choose a date and time that was convenient for them from a list of available dates and times. Each evaluator participated in one study session that lasted no more than two and a half hours. Evaluators participated individually; each study session consisted of only one evaluator. Figure 33 is an excerpt from Figure 29 that shows only the evaluator role.

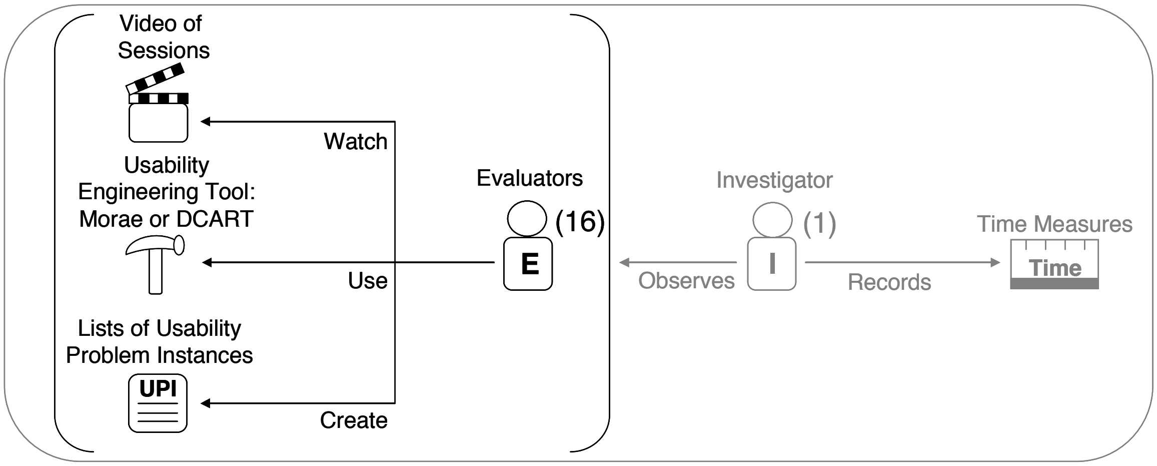


Figure 33: An excerpt from Figure 29 of the evaluator role. The numbers in parentheses indicate how many individuals participated in each role. Activities and objects related to the investigator role are grayed out.

When they arrived for the study, the evaluators read an informed consent form (Appendix B.3) and were given the chance to ask questions about the study. Evaluators who agreed to participate in the study signed the informed consent form.

After they had signed the consent form, the evaluators received a printed instruction booklet that was specific to the tool that they would be using (Morae – Appendix B.7; DCART – Appendix B.8). Regardless of the tool that they would be using, the evaluators followed the same basic process. During the first hour, the evaluators performed activities to familiarize themselves with their tool and the steps involved with performing a usability evaluation. During the next one and a half hours, the evaluators performed a usability evaluation of Scholar.

The following are the activities that the evaluators performed during the first hour of the study session:

1. The evaluators watched a tutorial video on their tool.
2. I explained the concept of UP instances to evaluators and gave them a printed diagram to show how raw usability data relates to UP instances (Appendix B.4).
3. The evaluators performed a practice usability evaluation of the Internet Movie Database (IMDB) website.
    a. The evaluators watched a video of a correct way to perform a task in the IMDB.
    b. The evaluators watched a video of a user trying to perform the task and used their tool to record UP instances experienced by the user.

The evaluators watched the video one time through without pausing or stopping to simulate conducting a usability evaluation in real time. Thereafter, they were allowed to rewind, play, fast forward, pause, and stop the video as much as they needed.

c. The evaluators recorded their lists of UP instances in a Word document and compared their list to a sample list specific to their tool (Morae – Appendix B.9; DCART – Appendix B.10). I spoke with them and gave them feedback on the UP instances that they had recorded.

The following are the activities that the evaluators performed during the next one and a half hours of the study session:

- The evaluators performed a usability evaluation of Scholar.

  a. The evaluators watched a video that introduced Scholar, a video of a correct way to add a student to a course, and a video of a correct way to remove a student from a course.

  b. The evaluators watched a video of a user trying to add a student, a video of a second user trying to add a student, and a video of the first user trying to remove a student. The evaluators used their tool to record UP instances experienced by the users. The evaluators watched the three videos one time through without pausing or stopping to simulate conducting a usability evaluation in real time. Thereafter, they were allowed to rewind, play, fast forward, pause, and stop the videos as much as they needed.

  c. The evaluators recorded their lists of UP instances in a Word document.

The Morae group evaluators made time-stamped comments using the observational capture features of Morae Remote Viewer while they watched the videos of representative users. They reviewed their comments, added new comments, and reviewed the video using Morae Manager. The evaluators edited and combined comments into UP instances in Morae and then exported them to a Word document, exported comments to a Word document and then edited and combined them into UP instances, or directly recorded UP instances in Word.

The DCART group evaluators used the session, task run, UP instance collection, UP instance review, and UP record forms (Section 4.3.2.2). I created the necessary session and task run objects for the DCART group evaluators. For this study, the UP collection and review forms did not have a field for capturing immediate intention, and the UP record did not have fields for immediate intention or UAF diagnosis. Study 2 (Section 5.1.4) explores diagnosis and includes these fields. DCART users used a function built into DCART to generate a Word document of UP instances from the UP instance records that they had created.

5.1.2.5 Experimental Design

This study was a between-subjects design with support for UP instance records (no support = raw comments, used Morae or support = UP instances, used DCART) as the independent variable. The dependent variables were time measures and measures of UP instance discovery and UP instance quality as rated by usability practitioners (judges in this study). More detailed information on these measures is available in Section 5.1.2.6.

I chose a between-subject design for two main reasons. First, less time was required of each evaluator, so I had less risk of participant dropout. Second, I would have had to account for learning that would have occurred in a within-subjects design. For a within-subjects design, I would have needed to have a number of different videos of representative user sessions and to perform counter-balancing.

The between-subject design also had certain limitations. First, it required more participants than a within-subjects design. However, funding was available to pay an hourly rate for a reasonable number of participants, so I was able to recruit enough participants. Second, between-subject designs often require matching or the establishment of groups based on characteristics that are highly correlated with the dependent variables. For this particular study, I anticipated that performance would be most closely related to basic knowledge (UE or HCI), experience with course management software, and English language skills. I filtered participants using the online questionnaire mentioned in the procedure (Section 5.1.2.4) and assigned participants, so that they were as evenly distributed between treatments as possible (Table 8).

Table 8: Matching of evaluators for Study 1

Treatment	UE Experience with or without HCI Experience	HCI Experience without UE Experience	CM Software Experience	Fluent in English
Raw Comments	5	3	6	6
Usab Prob Inst	5	3	6	6

HCI = human-computer interaction, CM = course management. The cell values indicate the number of participants that met each criterion. There were 8 participants per treatment. The values in the rows sum to more than 8 because the columns are not mutually exclusive. For example, an individual with UE experience might also have CM software experience and be fluent in English.

5.1.2.6 Data Collection and Analysis

5.1.2.6.1 Time Measures

As described in the introduction to this study, evaluator effectiveness was of primary interest for this study. Because I assumed a fixed-resources environment, however, I wanted to remove efficiency as a point of consideration. As a result, I recorded the following:

- The amount of time that the evaluators spent performing the evaluation
- Whether evaluators finished

Figure 34 is an excerpt from Figure 29 that shows only the investigator role in the generation of the time measures.

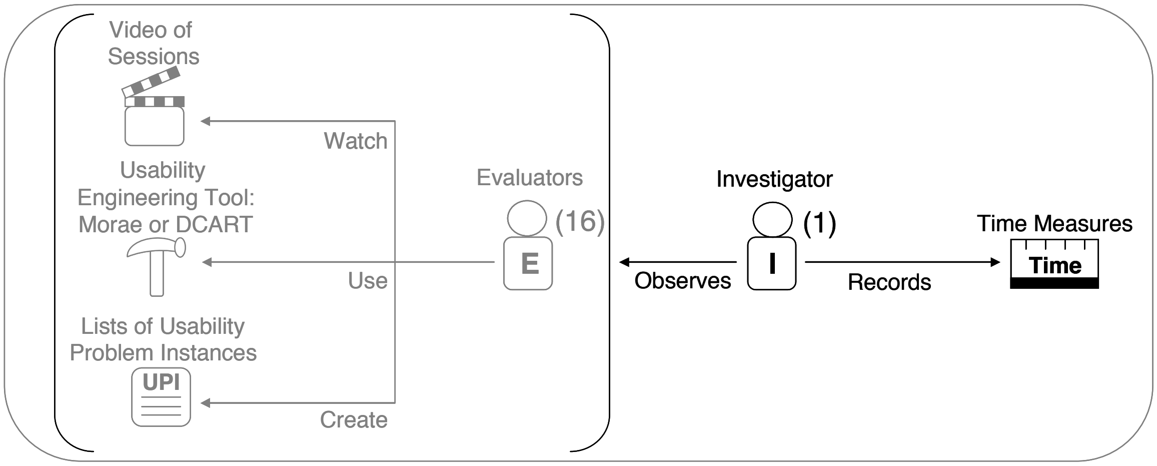


Figure 34: An excerpt from Figure 29 of the investigator role recording time measures. The numbers in parentheses indicate how many individuals participated in each role. Activities and objects related to the evaluator role are grayed out.

The time measurement is the total amount of time that evaluators spent watching the introduction video and the correct videos, recording comments or UP instances, and creating a list of UP instances.

I calculated a separate measure of whether evaluators finished their evaluations to accompany the time measure. Evaluators who were still working up to 5 minutes before the end of the time limit were given a 5 minute notice and asked to finish. After they turned in their evaluations, they were asked if they had finished their evaluation or if they had just turned it in because they were out of time. Those who answered that they were not finished were marked as not finished.

5.1.2.6.2 Measures of Usability Problem Instance Discovery

A number of steps were involved in calculating measures of UP instance discovery. First, I developed and applied a modified version of the SUPEX framework presented in Cockton & Lavery [1999] to structure the extraction of UP instances from the Scholar videos watched by evaluators during their study sessions (Section 5.1.2.4). Next, two individuals, whom I refer to as instance coders, used the SUPEX output to create a master list of the UP instances experienced by the representative users. The instance coders then matched lists of UP instances generated by evaluators to the master list. Finally, the master list and the counts of actual UP instances generated by the matching process were used as inputs to calculate the measures.

Modified SUPEX Framework

The SUPEX framework structures the process of extracting UPs from raw usability data. SUPEX consists of a number of stages, through which usability practitioners iterate until they achieve a desired level of UP extraction. For this study, however, I was concerned with extracting all UP instances, so I modified SUPEX to work with UP instances. As described in 1.3.3, multiple UP instances may represent the same UP. The modified SUPEX framework is shown in Table 9.

The isolation stage is concerned with identifying episodes or basic units of the representative users' interactions with Scholar and then associating UP instances with those episodes. The episode with which a UP instance is associated provides contextual information to the task run level (Section 4.2.1.1). The original SUPEX analysis stage includes three steps for describing, collecting, and generalizing UP instances into UPs. This study only included the description step because the goal was to identify all UP instances.

I performed the segmentation, abstraction, and threading steps of the isolation stage of the modified SUPEX framework on each of the three Scholar videos watched by evaluators during their study sessions (Section 5.1.2.4). The output is available in Appendix D. Figure 35 is an excerpt from Figure 29 that shows only the investigator role in the creation of the SUPEX output.

Table 9: Modified version of the SUPEX framework

Isolation stage

- Performed by the investigator
  - Segmentation – Divide each representative user's session into episodes. The boundaries of an episode exist where the representative user expresses or indicates a conscious goal. The use of conscious goals ensures that related actions stay in the same episode.
  - Abstraction – Divide episodes into basic, step, and sub task levels of granularity. One or more basic episodes comprise a step, and one or more steps comprise a sub task.
  - Threading – Identify sequences of related episodes that are not necessarily contiguous. Threading helps avoid over-reporting by identifying UP instances that users quickly recover from and helps avoid under-reporting by indicating seemingly minor UP instances that result in major problems.
- Performed by the instance coders
  - Coding – Identify UP instances and associate them with episodes of an appropriate level of abstraction.

Analysis stage

- Performed by the instance coders
  - Description – Describe each UP instance succinctly and completely.

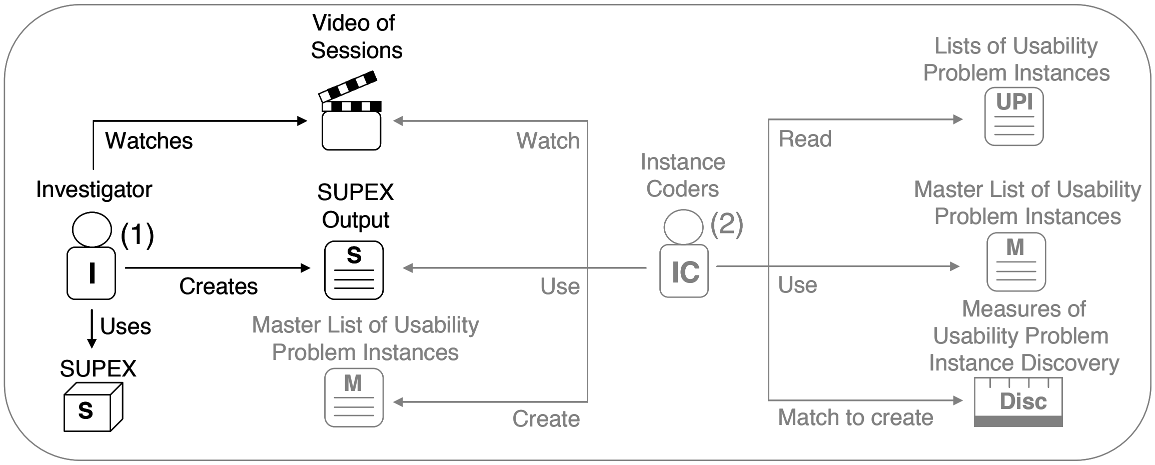

Video of Sessions
Watches
Watch
Lists of Usability Problem Instances
UPI
Read
Investigator
(1)
I
Creates
SUPEX Output
S
Use
Instance Coders
(2)
IC
Use
Master List of Usability Problem Instances
M
Uses
SUPEX
S
Master List of Usability Problem Instances
M
Create
Measures of Usability Problem Instance Discovery
Disc
Match to create

Figure 35: An excerpt from Figure 29 of the investigator role in the creation of the SUPEX output. The numbers in parentheses indicate how many individuals participated in each role. Activities and objects related to the instance coder role are grayed out.

Instance Coders

The instance coders created a master list of UP instances and matched UP instances in evaluators' lists with those in the master list. Figure 36 is an excerpt from Figure 29 that shows only the instance coder role.

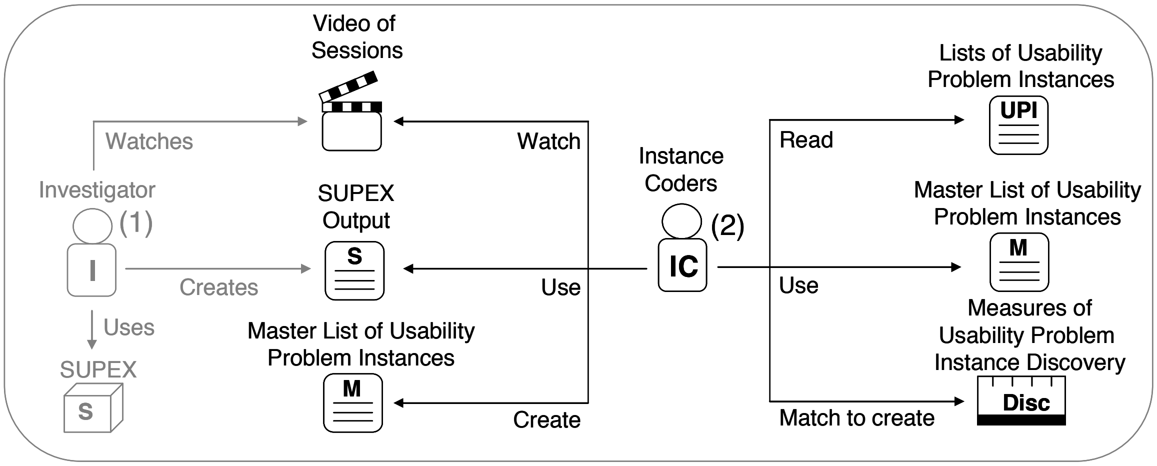

Video of Sessions
Watches
Watch
Investigator
(1)
I
Creates
Uses
SUPEX
S
SUPEX Output
S
Use
Master List of Usability Problem Instances
M
Create
Instance Coders
(2)
IC
Read
Use
Match to create
Lists of Usability Problem Instances
UPI
Master List of Usability Problem Instances
M
Measures of Usability Problem Instance Discovery
Disc

Figure 36: An excerpt from Figure 29 of the instance coder role. The numbers in parentheses indicate the number of individuals who participated in each role. Activities and objects related to the investigator role are grayed out.

I asked two professional contacts to serve as instance coders. I did not pay these individuals for their involvement, but I did provide food for them during scheduled meetings.

During our first meeting, I asked the instance coders to read and sign a consent form (Appendix E.1). After they had signed the consent form, I gave them an instruction booklet (Appendix E.2) that detailed each of the five tasks that they would perform for the study.

For task 1, the instance coders met with me as a group to learn about the process that they would be using to create a master list of UP instances and to practice identifying and recording UP instances. I explained the concept of UP instances to the instance coders; the instruction booklet contained a printed diagram to show how raw usability data relates to UP instances. The instance coders then watched the same videos of the IMDB as the evaluators watched in their familiarization sessions (Section 5.1.2.4) and recorded UP instances. The instance coders compared their lists of UP instances to a reference list (Appendix E.3) and discussed similarities and differences.

For task 2, the instance coders watched the same videos of Scholar as the evaluators watched during their study sessions (Section 5.1.2.4) and created lists of UP instances. The instance coders performed the coding step of the isolation stage and the description step of the analysis stage of the modified SUPEX process described in Table 9. The instance coders were asked to map each UP instance that they identified to a particular step in the SUPEX output. They were also instructed to include the following in each UP instance record: a name, a timestamp, and a description. The instance coders emailed me their lists of UP instances.

For task 3, the instance coders met with me as a group to compare the lists of UP instances that they had created during task 2. One instance coder presented UP instances while the other matched them; they then switched roles. They discussed UP instances that did not match and decided either to include them in the master problem list or to discard them if they did not represent actual instances of UPs. I recorded their decisions during the meeting and created the master list from their UP instances. I emailed the master list to them for approval and made any additional changes that they requested.

For task 4, the instance coders compared the lists of usability problem instances produced by the evaluators to the master list. They used a spreadsheet to record their results (Appendix E.4). The instance coders worked independently and viewed the evaluators' lists of UP instances in different orders; one instance coder's ordering was the reverse of that of the other. For each UP instance in each evaluator's list, they assigned values to indicate one of the following:

- The UP instance matched to a UP instance in the master list
- The UP instance did not exist in the master list and should be added
- The UP instance was not an actual UP instance experienced by a representative user or was too vague to be matched to a UP instance in the master list

For task 5, I used the following process to reconcile the results of the comparisons performed by the instance coders:

- If both instance coders agreed, the reconciled value was the agreed upon value.
- If both instance coders assigned values that represented UP instances in the master problem list, but these values did not agree, I reconciled them and decided upon a final value.
- If one or both instance coders indicated that a UP instance did not represent an actual UP instance, the reconciled value was that it was not an actual UP instance experienced by a representative user.

- If both instance coders indicated that a UP instance needed to be added to the master list, I added it. If only one instance coder indicated that a UP instance should be added, the reconciled value was the other instance coder's value.

I emailed the list of reconciliations and additions to the master usability problem instance list to the instance coders for approval.

Measures

Work by Hartson et al. [2001] provides a basis for comparing usability evaluation methods based on UP discovery. Although this study is not directly a study of usability evaluation methods, the measures developed by Hartson et al. are still appropriate. I modified the authors' measures in that I applied them to UP instances instead of UPs. The measures all require a baseline or master UP instance list, which the instance coders created using the process described earlier in this section. I calculated the following measures for each evaluator: discovery thoroughness, discovery validity, and discovery effectiveness. I also calculated a measure of discovery reliability by group. I prefaced all the measures with "discovery", so that they are not confused with other concepts, such as UE process effectiveness (Section 1.3.1).

Table 10: Formulas for calculating measures of usability problem instance discovery

Discovery thoroughness	$\dfrac{\text{\# actual UP instances identified by the evaluator}}{\text{\# UP instances in the master list}}$	
Discovery validity	$\dfrac{\text{\# actual UP instances identified by the evaluator}}{\text{\# total UP instances identified by the evaluator}}$	
Discovery effectiveness	identification thoroughness * identification validity	
Discovery reliability	$\dfrac{\sum_{i=2}^{n}\sum_{j=1}^{i-1}\frac{\lvert P_i \cap P_j \rvert}{\lvert P_i \cup P_j \rvert}}{n(n-1)/2}$	where n is the number of evaluators and P_i is the set of problems found by evaluator i

Discovery thoroughness is a relative measure of the actual number of UP instances identified by the evaluator as a fraction of the number of actual UP instances that exist in the master list for the videos of the representative user sessions. Discovery validity is a measure of the actual UP instances identified by the evaluator expressed as a fraction of the total number of UP instances identified by the evaluator. Finally, discovery effectiveness is the product of thoroughness and validity.

Discovery reliability is a measure of the consistency or agreement with which evaluators are able to use a tool to identify UP instances. A number of methods have been used to calculate reliability. For example, Nielsen [1994] used Pearson's coefficient of correlation; Hartson et al. [2001], however, argue that a measure of agreement is preferred over a measure of correlation. Andre et al. [2001] used an extension to Cohen's Kappa [1960] developed by Fleiss [1971] in evaluating the reliability with which usability practitioners could use the UAF to diagnose problems. Capra [2006], however, argues that Cohen's Kappa merges judgment association and bias, thereby making it difficult to compare kappa across samples, such as between treatments. Instead, Capra recommends any-two agreement [2006]. Any-two agreement compares each evaluator's set of UP instances to each other evaluator's set of UP instances; reliability is high among evaluators who identify the same UP instances. Any-two agreement is ratio of the intersection to the union of UP instances identified by a pair of evaluators [Hertzum & Jacobsen, 2003].

5.1.2.6.3 Measures of Usability Problem Instance Quality as Rated by Judges

A number of steps were involved in calculating measures of UP instance quality as rated by judges. First, I modified guidelines developed by Capra [2006] for UP descriptions. Next, two individuals with usability experience, whom I refer to as judges, rated the lists of UP instances produced by evaluators based on the guidelines from the perspective of a usability practitioner. Finally, the ratings were used as inputs to calculate the measures.

Capra's Guidelines

Capra [2006] developed 10 guidelines (Table 11) for UP descriptions based on surveys of usability practitioners. These guidelines were tested in a study in which practitioners and graduate students watched the same 10 minute recording of sessions with representative users of a web site and created usability evaluation reports. Three judges rated each of the usability evaluation reports on the guidelines. The practitioners received higher ratings overall and for the following three guidelines: *support with data*, *describe the impact*, and *describe a solution*. Capra's work suggests that the guidelines can be applied as measures of quality of usability evaluation reports. Higher ratings on the guidelines, however, did not map to higher values of thoroughness or validity. As such, it is appropriate to include both the discovery measures in Section 5.1.2.6.2 and these quality measures in this study.

Table 11: Capra's guidelines for usability problem descriptions

1. **Be clear and precise while avoiding wordiness and jargon.** Define terms that you use. Be concrete, not vague. Be practical, not theoretical. Use descriptions that non-HCI people will appreciate. Avoid so much detail that no one will want to read the description.
2. **Describe the impact and severity of the problem,** including business effects (support costs, time loss, etc.), impact on the user's task and importance of the task. Describe how often the problem will occur, and system components that are affected or involved.
3. **Support your findings with data** such as: how many users experienced the problem and how often; task attempts, time and success/failure; critical incident descriptions; and other objective data, both quantitative and qualitative. Provide traceability of the problem to observed data.
4. **Describe the cause of the problem**, including context such as the interaction architecture and the user's task. Describe the main usability issue involved in the problem. Avoid guessing about the problem cause or user's thoughts.
5. **Describe observed user actions,** including specific examples from the study, such as the user's navigation flow through the system, user's subjective reactions, screen shots and task success/failure. Mention whether the problem was user-reported or experimenter observed.
6. **Consider politics and diplomacy when writing your description.** Avoid judging the system, criticizing decisions made by other team members, pointing fingers or assigning blame. Point out good design elements and successful user interactions. Be practical, avoiding theory and jargon.
7. **Be professional and scientific in your description.** Use only facts from the study, rather than opinions or guesses. Back your findings with sources beyond the current study, such as external classification scheme, proven usability design principles, and previous research.
8. **Describe a solution to the problem,** providing alternatives and tradeoffs. Be specific enough to be helpful without dictating a solution, guessing, or jumping to conclusions. Supplement with pictures, screen capture, usability design principles and/or previous research.
9. **Describe your methodology and background.** Describe how you found this problem (field study, lab study, expert evaluation, etc.). Describe the limitations of your domain knowledge. Describe the user groups that were affected and the breadth of system components involved.
10. **Help the reader sympathize with the user's problem** by using descriptions that are evocative and anecdotal. Make sure the description is readable and understandable. Use user-centric language rather than system-centric. Be complete while avoiding excessive detail. Capra suggests that further work is needed to develop the guidelines for grading usability problem descriptions.

The guidelines are listed in order from most to least required per Capra

In studies in her dissertation, Capra used only guidelines 1, 2, 3, 4, 5, and 8. I use the same subset of guidelines. The first five guidelines were rated in a survey of practitioners by Capra as being most required; guideline 8 is important is this study because I am interested in documenting solutions at the UP instance level. I modify the selected subset of guidelines both in terms of content and presentation for my study. The modified subset of guidelines is shown in Table 12.

Table 12: Modified subset of Capra's guidelines

1. **Be clear and precise while avoiding wordiness and jargon.**
   - Define terms that you use.
   - Be concrete, not vague.
   - Be practical, not theoretical.
   - Use descriptions that non-HCI people will appreciate.
   - Avoid so much detail that no one will want to read the description.
2. **Describe the impact and severity of the problem.**
   - Describe how it impacts the user's task.
   - Describe how often the problem will occur, and system components that are affected or involved.
3. **Support your findings with data.**
   - Include information on how many users experienced the problem and how often.
   - Include objective data, both quantitative and qualitative, such as the number of times a task was attempted or the time spent on the task.
   - Provide traceability of the problem to observed data.
4. **Describe the cause of the problem.**
   - Describe the main usability issue involved in the problem.
   - Avoid guessing about the problem cause or user's thoughts.
5. **Describe observed user actions.**
   - Include contextual information about the user and the task.
   - Include specific examples, such as the user's navigation flow through the system, user's subjective reactions, screen shots and task success/failure.
   - Mention whether the problem was user-reported or experimenter observed.
6. **Describe a solution to the problem.**
   - Provide alternatives and tradeoffs.
   - Be specific enough to be helpful without dictating a solution.
   - Supplement with usability design principles.

The guidelines are listed in order from most to least required per Capra

Judges

Two individuals, whom I refer to as judges, applied the modified subset of guidelines to the lists of UP instances created by the evaluators. The judges assigned quality ratings from the perspective of a usability practitioner. Figure 37 is an excerpt from Figure 29 that shows only the judge role.

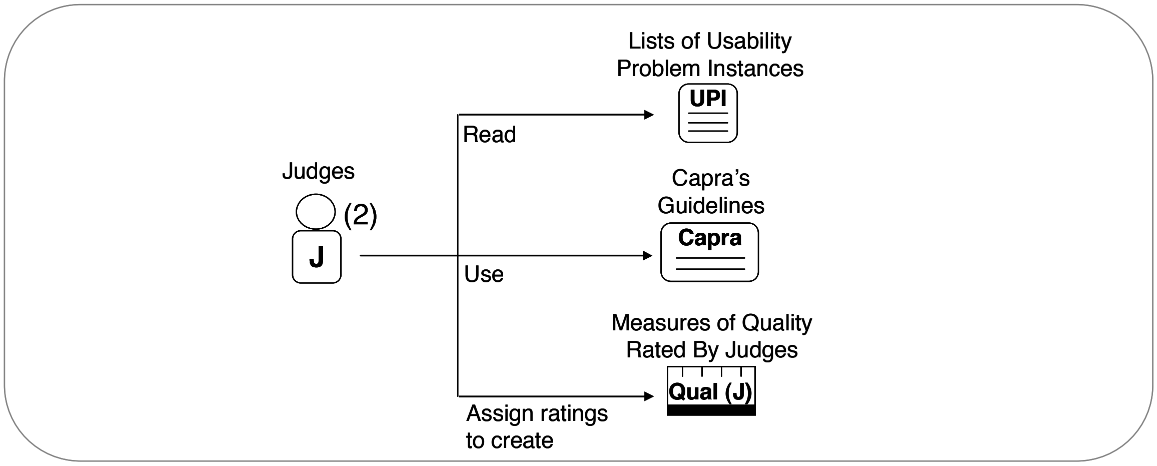


Figure 37: An excerpt from Figure 29 of the judge role. The number in parentheses indicates the number of individuals in the role.

I asked two professional contacts to serve as judges. One judge is a practicing usability professional, and the other judge is a doctoral computer science student with academic UE and HCI experience. I did not pay these individuals for their involvement, but I did provide food for them during scheduled meetings.

During our first meeting, I asked the judges to read and sign a consent form (Appendix F.1). After they had signed the consent form, I gave them an instruction booklet (Appendix F.2) that detailed each of the two tasks that they would perform for the study.

For task 1, the judges met with me as a group to learn about the process that they would be using to judge UP instances. I explained the concept of UP instances; the instruction booklet contained a printed diagram to show how raw usability data relates to UP instances. I also explained Capra's guidelines; the instruction booklet contained a table of the guidelines. The judges read a document (Appendix F.3) that provided background for three sample lists of UP instances (Appendix F.4) used as practice exercises. The judges used a spreadsheet (Appendix F.5) to rate each list of UP instances on each guideline using a six-point Likert-type scale with the following values: strongly disagree,

disagree, somewhat disagree, somewhat agree, agree, strongly agree. For example, for the first guideline, each judges assigned a rating based on whether he or she felt that the list of UP instances was clear and precise and avoided wordiness and jargon. After each sample list, the judges discussed their ratings. Through these discussions, the judges developed a general process for rating. For each guideline, they would first determine whether they agreed or disagreed. If they agreed, they would assign an initial rating of somewhat agree and then find positive examples of the guideline to increase the rating. If they disagreed, they would assign an initial rating of strongly disagree and then find positive examples of the guideline to increase the rating.

For task 2, the judges watched the same videos of Scholar as the evaluators watched during their study sessions (Section 5.1.2.4). The judges used a spreadsheet identical to the one that they used in the practice session to record their ratings. The judges worked independently and viewed the evaluators' lists of UP instances in different orders; one judge's ordering was the reverse of that of the other to balance any potential familiarization or learning effects.

Measures

The judges' ratings are the basis for the following measures of quality:

- Mean rating across all guidelines
- Mean rating per guideline

The mean ratings are intended to represent quality per treatment. A higher mean rating would map to more agreement with the guideline(s) thereby indicating higher quality.

5.1.3 Results

5.1.3.1 Hypothesis 3a.1

I hypothesized that tool support for UP instance records would not affect the time that it takes novice evaluators to perform evaluations. I calculated two time measures: the amount of time that the evaluators spent performing the evaluation and whether evaluators finished (Section 5.1.2.6.1).

5.1.3.1.1 Time

Figure 38 illustrates mean time values by treatment.

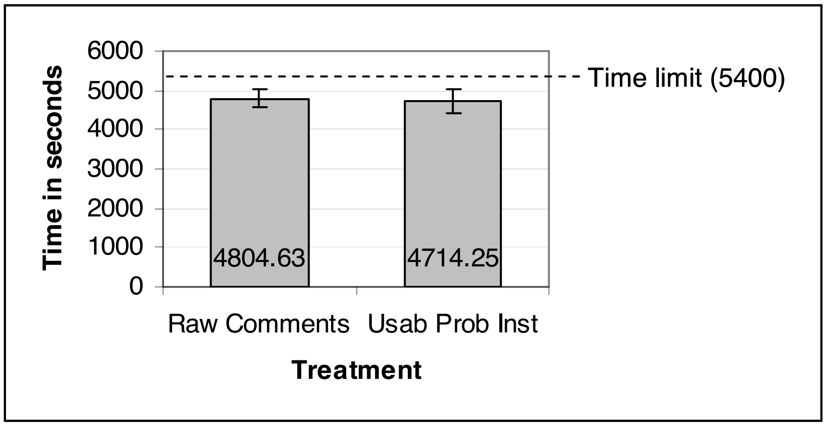


Figure 38: Study 1 - Mean time value by treatment, bars represent standard error

A histogram of the time values suggested that there was a ceiling effect due to the one and a half hour time limit on the evaluators. A Shapiro-Wilk test was performed on the time values; the null hypothesis that they came from a normal distribution was rejected, *W*=0.83, *p*=0.01. As a result, a Wilcoxon rank-sum test, a non-parametric test, was performed instead of a *t*-test, a parametric test. Using a normal approximation procedure, the test indicated that there was not a significant difference in the medians between treatments, *W*=67, *p*=0.96. Table 13 contains descriptive statistics for the time values.

Table 13: Study 1 - Time value by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
Raw Comments	4,804.63	653.28	3,862	5,400
Usab Prob Inst	4,714.25	842.97	3,386	5,400

Cell values represent time in seconds, n=8 per treatment

5.1.3.1.2 Finished

There was no significant difference between treatments in the number of evaluators who finished. In both treatments, 4 evaluators finished and 4 did not finish (Figure 39).

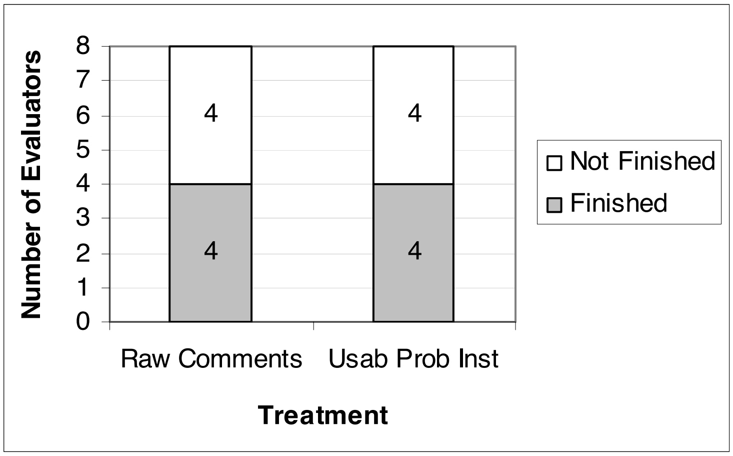


Figure 39: Study 1 - Finished count by treatment

5.1.3.2 Hypothesis 3a.2

I hypothesized that tool support for UP instance records would increase the UP instance discovery of novice evaluators. I calculated four measures of UP instance discovery: discovery thoroughness, discovery validity, discovery effectiveness, and discovery reliability (Section 5.1.2.6.2). Figure 40 illustrates the mean values for these measures.

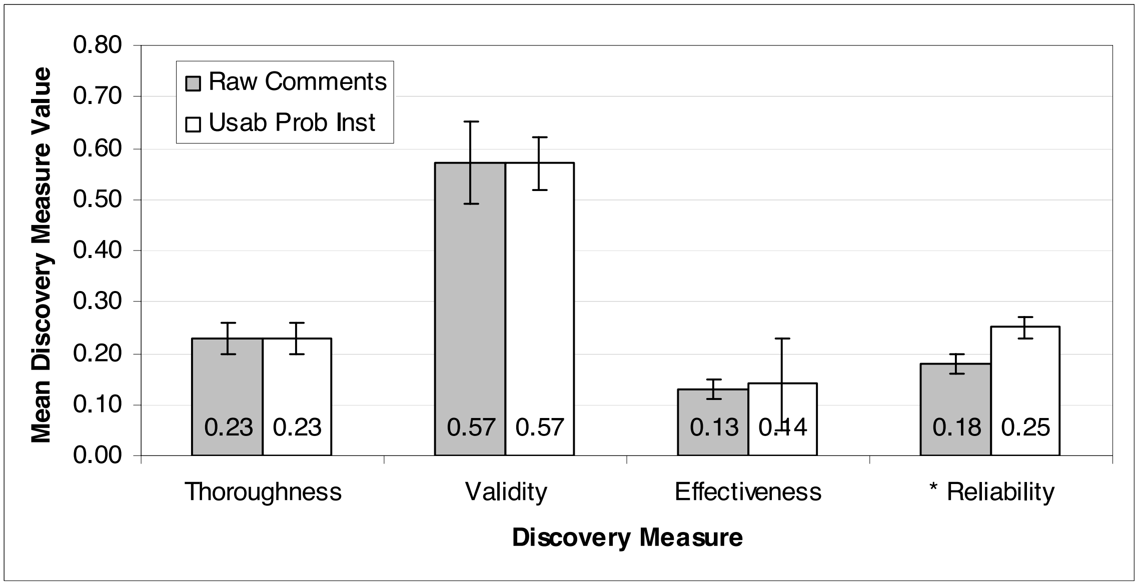


Figure 40: Study 1 - Mean discovery thoroughness, discovery validity, discovery effectiveness, and discovery reliability measures by treatment, bars represent standard error, * *p*<0.05

5.1.3.2.1 Discovery Thoroughness

A Shapiro-Wilk test was performed on the discovery thoroughness values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.93, *p*=0.26. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(1)<0.01, *p*=0.94. A t-test assuming equal variances indicated that there was not a significant difference between the treatment means, *t*(14)=0.11, *p*=0.91. Table 14 contains descriptive statistics for the discovery thoroughness values.

Table 14: Study 1 - Discovery thoroughness by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
Raw Comments	0.23	0.09	0.11	0.37
Usab Prob Inst	0.23	0.09	0.11	0.37

n=8 per treatment

5.1.3.2.2 Discovery Validity

A Shapiro-Wilk test was performed on the discovery validity values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.96, *p*=0.74. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(1)=1.25, *p*=0.26. A t-test assuming equal variances indicated that there was not a significant difference between the treatment means, *t*(14)<0.01, *p*=1.00. Table 15 contains descriptive statistics for the discovery validity values.

Table 15: Study 1 - Discovery validity by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
Raw Comments	0.57	0.22	0.24	0.85
Usab Prob Inst	0.57	0.14	0.38	0.74

n=8 per treatment

5.1.3.2.3 Discovery Effectiveness

A Shapiro-Wilk test was performed on the discovery effectiveness values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.97, *p*=0.76. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(1)=0.48, *p*=0.49. A t-test assuming equal variances indicated that there was not a significant difference between the treatment means, *t*(14)=0.39, *p*=0.70. Table 16 contains descriptive statistics for the discovery effectiveness values.

Table 16: Study 1 - Discovery effectiveness by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
Comments	0.13	0.07	0.02	0.22
Usab Prob Inst	0.14	0.09	0.04	0.27

n=8 per treatment

5.1.3.2.4 Discovery Reliability

A Shapiro-Wilk test was performed on the discovery reliability values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.97, *p*=0.29. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(1)=0.07, *p*=0.79. A t-test assuming equal variances indicated that there was a significant difference between the treatment means, *t*(14)=2.32, *p*=0.02. Table 17 contains descriptive statistics for the discovery reliability values.

Table 17: Study 1 - Discovery reliability by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
Raw Comments	0.18	0.12	0.00	0.50
Usab Prob Inst	0.25	0.11	0.06	0.50

n=8 per treatment

5.1.3.3 Hypothesis 3a.3

I hypothesized that tool support for UP instance records would increase the quality of novice evaluators' descriptions of UP instances as rated by usability practitioners (judges in this study). I calculated two measures of quality: mean rating across all guidelines and mean rating per guideline (Section 5.1.2.6.3).

5.1.3.3.1 How the Judges Rated

Although differences in mean rating were of primary interest, an understanding of how the judges rated was useful in interpreting differences. I calculated measures of association, bias, and distribution for the judges.

Association

Association, the tendency of each judge to give higher/lower ratings to the same evaluator, was tested using Pearson's product-moment correlation both by treatment and guideline (Table 18 and Table 19). Using an alpha level of 0.05, there was a significant correlation between the judges for both treatments. Using an alpha level of 0.05, there was a significant correlation between the judges for all guidelines, *describe the cause*, and *describe a solution*. The correlations were not significant for *be clear and precise*, *describe the impact*, *support with data*, and *describe observed actions*, which suggests that the judges used different underlying traits to form their judgments for these guidelines.

Table 18: Study 1 - Judge association by treatment, tested using Pearson's product-moment correlation

Treatment	*r*	*p*
Raw Comments	0.57	<0.01*
Usab Prob Inst	0.32	0.03*

n=48 per treatment, * *p*<0.05

Table 19: Study 1 - Judge association by guideline, tested using Pearson's product-moment correlation

Guideline	*R*	*p*
All guidelines	0.49	<0.01*
Be clear and precise	0.25	0.36
Describe the impact	0.34	0.20
Support with data	0.04	0.86
Describe the cause	0.59	0.02*
Describe observed actions	0.43	0.10
Describe a solution	0.87	<0.01*

n = 16 per guideline, * *p*<0.05

Bias

Bias is the tendency of each judge to give higher or lower ratings overall. Bias was tested using a 2x6x2 mixed-factor ANOVA, with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure. The results of the same ANOVA were used to compare mean rating scores; see Section 5.1.3.3.2 for the full ANOVA results. Using an alpha level of 0.05, the judge main effect was not significant, *F*(1, 168)=0.62; the judge x guideline interaction was not significant, *F*(5, 168)=0.83; and the judge x treatment x guideline interaction was not significant, *F*(5, 168)=0.57. The judge x treatment interaction, however, was significant, *F*(1, 168)=9.19. The judge x treatment interaction was explored using slices to test for simple effects due to judge for each treatment. There was an effect due to judge in the UP instances treatment. The mean rating for judge j1 (*M*=0.67, *SD*=1.34) was significantly greater than the mean rating for judge j2 (*M*=-0.04, *SD*=0.97). Figure 41 shows the judges' mean ratings by treatment.

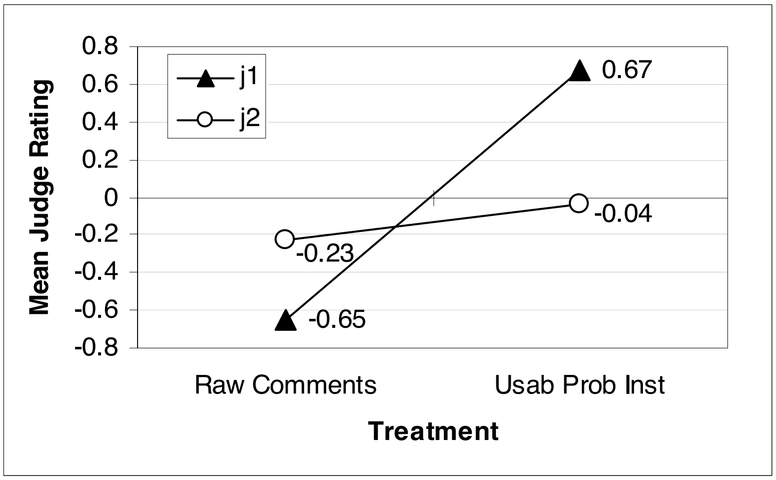


Figure 41: Study 1 - Judge bias, mean rating by judge by treatment

Distribution

Distribution is the tendency of each judge to use each point in the scale. Distribution was assessed using visual inspection of the judges' ratings by treatment (Figure 42) and by guideline (Figure 43). Each judge gave 48 ratings per treatment. Judge j1 used the endpoints (strongly disagree and strongly agree) more frequently (n=34) than judge j2 (n=11). Judge j2 used the innermost points (somewhat disagree and somewhat agree) more frequently (n=60) than judge j1 (n=28). Judge j1 gave more positive ratings (any of the agree ratings) (n=51) than judge j2 (n=38).

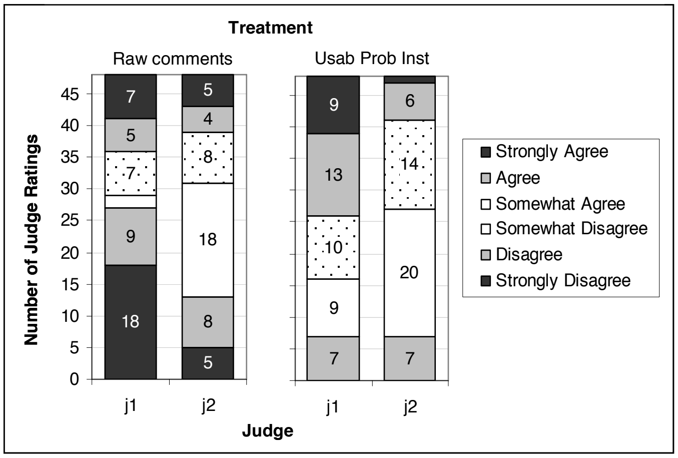


Figure 42: Study 1 - Judge distribution by treatment

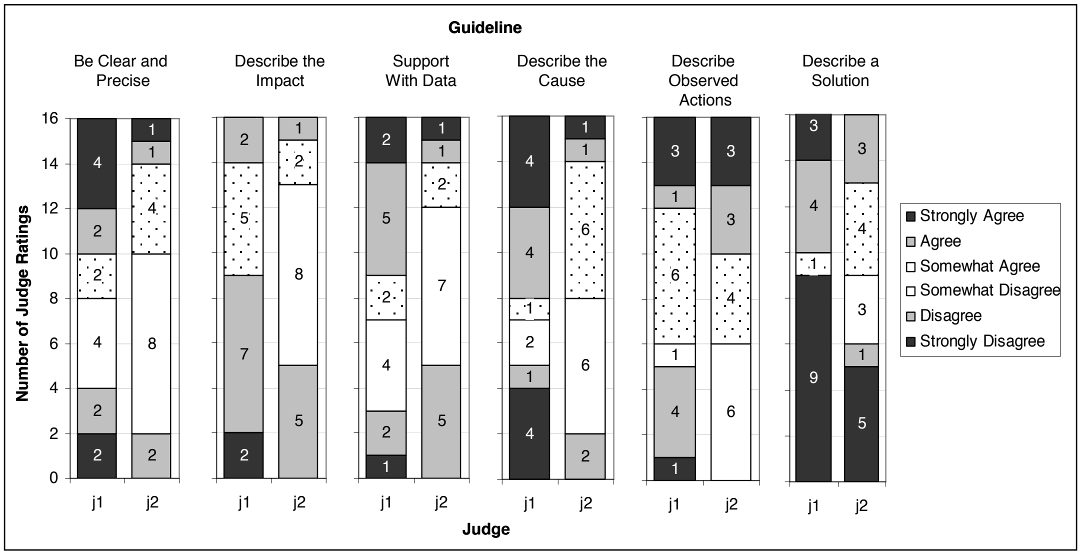


Figure 43: Study 1 - Judge distribution by guideline

5.1.3.3.2 Mean Ratings

The rating data are interval and can have only six values, so it is not possible to test for normality using the Shapiro-Wilk test. Instead, both a histogram and a normal quantile plot suggest that the rating data is jointly normally distributed.

Means are based on individual ratings given by each judge, rather than the sum of the two ratings. Judges rated on a 6-point scale, which has been adjusted to a rating from –2.5 to 2.5. Differences in mean rating across all guidelines by treatment were tested as part of a 2x6x2 mixed-factor ANOVA, with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure (Table 20). The effects specific to judge are discussed in the analysis of judge bias in Section 5.1.3.3.1.

Table 20: Study 1 - Quality as rated by judges, results of a 2x6x2 mixed factor ANOVA with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure

Source	*F*	DF Num	DF Den	*p*
Judge	0.62	1	168	0.43
Treatment	16.35	1	168	<0.01*
Guideline	3.00	5	168	0.01*
Judge x Treatment	9.19	1	168	<0.01*
Judge x Guideline	0.83	5	168	0.53
Treatment x Guideline	9.26	5	168	<0.01*
Judge x Treatment x Guideline	0.78	5	168	0.57

N=192, * *p*<0.05

Treatment x Guideline

The treatment x guideline interaction was explored using slices to test for simple effects due to treatment for each guideline (Table 21). The judges gave significantly higher ratings for *describe the cause* and *describe a solution* for the UP instances treatment. Figure 44 shows the mean ratings by treatment by guideline.

Table 21: Study 1 - Quality as rated by judges, simple effects due to guideline for each treatment explored using slices

Guideline	*F*	*p*
Be clear and precise	0.02	0.89
Describe the impact	0.17	0.68
Support with data	0.68	0.41
Describe the cause	4.26	0.04*
Describe observed actions	0.30	0.58
Describe a solution	57.23	<0.01*

DF Num=1, DF Den=168, * *p*<0.05

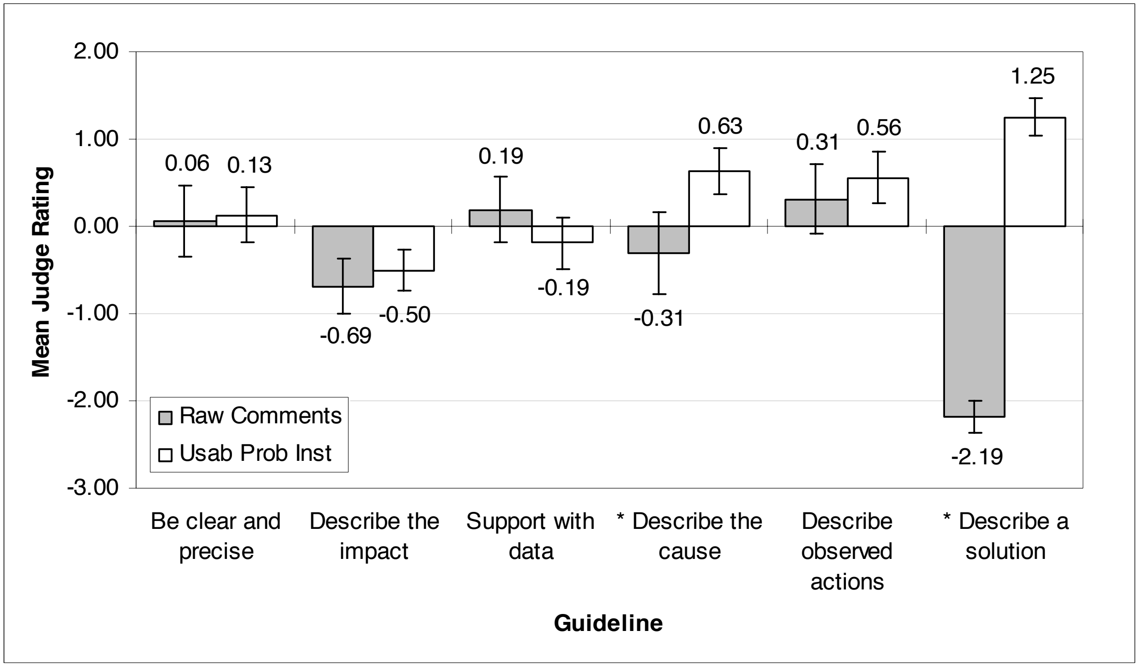


Figure 44: Study 1 - Quality as rated by judges, mean rating by treatment by guideline, bars represent standard error, * *p*<0.05

5.1.4 Discussion

Table 22 contains a summary of hypothesis testing results for study 1.

Table 22: Study 1 - Summary of Hypothesis Testing Results

Hypothesis	Result
H3a.1 – Tool support for UP instance records will not affect the time that it takes novice evaluators to perform evaluations.	**Supported** – There was not a significant difference in the time that it took evaluators to perform the evaluation or the number of evaluators who finished between treatments.
H3a.2 – Tool support for UP instance records will increase the UP instance discovery of novice evaluators.	UP instance discovery measures were calculated by matching lists of UP instances produced by evaluators to a master list of UP instances. **Supported** – The evaluators in the UP instance treatment were significantly more reliable in terms of UP instance discovery. **Not supported** – There were not significant differences between treatments for the thoroughness, validity, and effectiveness measures.
H3a.3 – Tool support for UP instance records will increase the quality of novice evaluators' descriptions of UP instances as rated by usability practitioners (judges in this study).	Measures of quality are based on Capra's guidelines [2006]. Higher mean ratings map to more agreement with the guideline(s) thereby indicating higher quality. **Supported** – The lists of UP instances produced by evaluators in the UP instances treatment were rated by the judges to be of higher quality overall. Further exploration of the difference in quality between the treatments by guideline revealed that evaluators in the UP instances treatment received significantly higher ratings for the following guidelines: *describe the cause* and *describe a solution*. **Not Supported** – There were not significant differences in quality as rated by judges between treatments for the following guidelines: *be clear and precise*, *describe the impact*, *support with data*, and *describe observed actions*.

5.1.4.1 Time

There was not a significant difference in the time that it took evaluators to perform the evaluation or the number of evaluators who finished between treatments. Regardless of the tool that they used, evaluators had to identify UP instances and then describe them. One possible explanation for the lack of a difference in time is that the evaluators in the raw comments treatment spent more time reviewing and combining comments into UP instances, while the evaluators in the UP instances treatment spent more time describing each instance. The evaluators in the raw comments treatment did not have explicit support for UP instances and were required to make a second pass through their data to recognize UP instances. Evaluators in the UP instances treatment created UP instance records during usability data collection and did not have to make a second pass through the data. The UP instance records, however, were form-based, and the evaluators had to describe each UP instance in terms of the fields in the form. The time data that I collected are not specific enough to support or refute this explanation. Regardless, the results indicate that there is no time penalty associated with working with usability data at the UP instance level of abstraction.

5.1.4.2 Usability Problem Instance Discovery

UP instance discovery measures were calculated by matching lists of UP instances produced by evaluators to a master list of UP instances. The evaluators in the UP instance treatment were significantly more reliable in terms of UP instance discovery. There were not significant differences between treatments for the thoroughness, validity, and effectiveness measures.

5.1.4.2.1 Explanation of Difference in Reliability Using a Model of the Usability Data Collection Stage

The evaluators in the UP instance treatment were significantly more reliable in terms of UP instance discovery. To explain this result, I developed a model of what occurs within the usability data collection stage (Figure 3). In Figure 45, the horizontal arrow represents time, the boxes represent activities that occur over a period of time, and the black dots represent specific points in time.

During the usability data collection stage of a lab-based usability evaluation, the facilitator observes a user performing tasks. Critical incidents provide clues or hints that the user has experienced a UP while performing a task. A facilitator may need to observe the user for a period of time after the initial onset of the critical incident to recognize or realize that a critical incident has occurred. After the facilitator has established that a critical incident has occurred, the facilitator formulates or determines how to describe the critical incident.

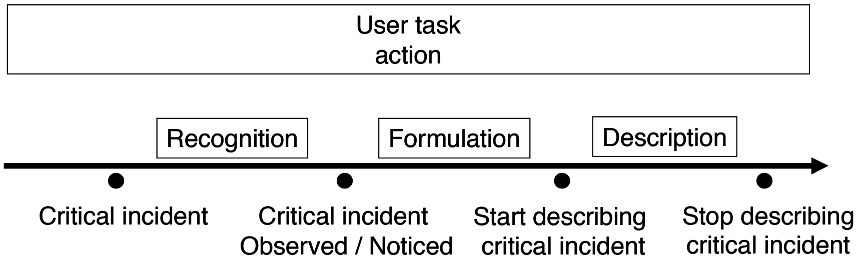


Figure 45: A model of what occurs within the usability data collection stage. The horizontal arrow represents time, the boxes represent activities that occur over a period of time, and the black dots represent specific points in time

The relationship between critical incidents and UPs can take many forms. Sometimes, a critical incident indicates that the user has experienced a particular UP. For example, a user's difficulty in selecting the dropdown arrow on the side of a font selection box may indicate that the dropdown arrow is too small and that the UP deals with physical actions.

Other times, a critical incident indicates that the user has experienced multiple UPs. For example, a user may say "I can't read this button". Closer inspection of the button may reveal that the font size of the label on the button is too small, the contrast between the colors used for the lettering and the background is poor, and the actual words used for the label are not readily understandable by users with certain backgrounds.

Still at other times, multiple seemingly unrelated critical incidents indicate that the user has experienced one particular UP. For example, consider an online photo album application in which it is necessary to create an album and create pages in that album before uploading images to the pages. A user who has created an album but not created pages may click on grayed out links for uploading images and may also search the help system for page backgrounds. These loosely related critical incidents may indicate that the user does not understand the conceptual metaphor of the photo album application.

One explanation for the higher rates of reliability in the UP instances treatment is that the evaluators more consistently interpreted the relationship between critical incidents and UPs when working with usability data at the UP instance level. The evaluators recognized critical incidents in terms of UP instances and then formulated them in context as a whole package of usability data (the recognition and formulation activities in Figure 45). The evaluators working at the raw usability data level, on the other hand, treated critical incidents as single data points during recognition and formulation and then reconstructed UP instances from them after they were finished watching the videos. Because they had packages of usability data as opposed to single data points, the evaluators in the UP instance treatment could more consistently determine when a critical incident indicated multiple UPs and when multiple critical incidents indicated only one UP.

This explanation also supports the lack of a significant difference between treatments for the thoroughness, validity, and effectiveness measures. These three measures are directly related to the ability of the evaluator to notice critical incidents. Tool support for UP instances helps evaluators more consistently work with critical incidents, but it cannot help them notice critical incidents.

5.1.4.2.2 Benefit of Reliability

If evaluators are more reliable in the UP instances that they identify and fail to identify, the usability evaluation process becomes more independent of the evaluators. Research on the evaluator effect (Section 2.1.1) indicates that evaluators find different numbers and types of UPs. As a result, involving more evaluators in a usability evaluation tends to result in the identification of more UPs. If evaluators were more reliable, however, involving more evaluators would not result in a substantial increase in the number of identified UPs because the evaluators would identify roughly the same UPs. More reliable identification of UPs would shift the focus away from the evaluators to tuning the usability engineering process.

If certain UPs are reliably identified during the usability evaluation sub-process, the systems analysis, design, and implementation sub-processes can be fine-tuned to eliminate them. Consider a situation in which there are problems with labeling in a particular suite of applications. For example, the terms used for the labels of buttons, menu items, and other interface objects may be in conflict with what users in the target domain expect. Potential solutions include integrating a more thorough review of artifacts in the systems analysis sub-process or including a technical writer in the design team.

5.1.4.3 Quality as Rated by Judges

Measures of quality were based on Capra's guidelines [2006]. Higher mean ratings map to more agreement with the guideline(s) thereby indicating higher quality. The lists of UP instances produced by evaluators in the UP instances treatment were rated by the judges to be of higher quality overall. Further exploration of the difference in quality between the treatments revealed that evaluators in the UP instances treatment received significantly higher ratings for the following guidelines: *describe the cause* and *describe a solution*.

5.1.4.3.1 Scaffolding

As discussed in Section 4.1.1, the idea for UP instance records takes into account Vygotsky's [1978] concept of the zone of proximal development, which is the distance between what an individual can do on his own and what he could be helped to achieve with competent assistance; scaffolding is a term used to describe this assistance. I believe that making the leap from raw usability data in the form of comments to UPs is difficult for novice practitioners. UP instances serve as a scaffolding to help novice usability practitioners construct UPs from comments.

The significant difference between treatments for *describe the cause* supports the idea of UP instances as scaffolding. The guidelines that weren't significantly different between treatments (*be clear and precise*, *describe the impact*, *support with data*, and *describe observed actions)* do not necessarily require the synthesis of usability data. For example, it is possible to describe a user's observed actions without really understanding the problem that is motivating those actions. To describe the cause of a UP, however, an evaluator must understand the UP and be able to clearly distinguish it from other UPs. Thinking about usability data in terms of instances of UPs allows evaluators to make this distinction.

5.1.4.3.2 Form-based Approach

Implicit in the support for UP instances built into DCART is a form-based approach to collecting usability data. One of the fields of the form is used to record potential solutions or suggestions for fixing a UP. As a result, I fully expected the lists of UP instances produced by the evaluators in the UP instances treatment to receive higher scores for *describe a solution* than those produced by evaluators in the raw comments treatment. The result provides support for a form-based approach to collecting and organizing usability data. Novice evaluators may not know or be able to quickly recall what data are important; a form-based approach helps to remind them. In the case of this study, the inclusion of a specific solution field in the form reminded them to provide a solution, which increased their ratings for *describe a solution*.

5.1.4.4 Limitation of the Study

The use of only three relatively short video clips (three to six minutes each) of representative users performing tasks with Scholar was the major limitation of this study. In a real lab-based usability evaluation, an evaluator would watch a user perform a number of tasks over a longer period of time (typically one to two hours) and would have more of an opportunity to observe and understand the difficulties experienced by the user. I limited the number and length of video clips because I wanted to simulate a fixed-resources environment, which is novel for this area of research, but which might reflect real-world development constraints. I did, however, provide the evaluators in the study with background information on the context for the tasks and explain that the tasks represented a subset of tasks from an evaluation with five representative users. I also provided the evaluators with videos on Scholar and the correct way to perform the tasks attempted by the representative users.

5.2 Study 2: Support for Diagnosis

The UAF, a conceptual framework of usability concepts (Section 4.2.2), gives usability practitioners a common way to understand and relate usability data and a common vocabulary for discussing it. Diagnosis with the UAF served as the basis for my approach to diagnosis. This study explored the role of diagnosis in analysis through a comparison of the effectiveness of evaluators based on three levels of diagnosis: no diagnosis, partial diagnosis, and full diagnosis. For this study, partial diagnosis consisted of identifying immediate intention. Full diagnosis was limited to the top three levels of the UAF; complete diagnosis for all problems would have required too much time of evaluators.

As in study 1, evaluator effectiveness was of primary interest for this study. Because I assumed a fixed-resources environment, I wanted to remove efficiency as a point of consideration. To confirm this operating assumption, I recorded the amount of time that it took evaluators to perform the evaluations. Of interest with regard to effectiveness were measures of UP instance discovery and quality as rated by usability practitioners.

Figure 46 is an overview of study 2; it shows roles and the tools and objects that people in the roles interact with and produce. This figure is referenced in future sections that describe the various roles in more detail.

5.2.1 Research Question and Hypotheses

The research question addressed by this study is directly related to RG3 in Section 1.5.

- RQ3b – How does tool support for diagnosis affect the effectiveness of novice evaluators?

This study compares the effectiveness of evaluators based on the level of diagnosis performed. Full diagnosis is the most thorough form of diagnosis, while no diagnosis is the least thorough. The following hypotheses apply to research question RQ3b:

- Hypothesis 3b.1 (H3b.1) – Tool support for diagnosis will not affect the time that it takes novice evaluators to perform evaluations.
- Hypothesis 3a.2 (H3b.2) – Tool support for diagnosis will not affect the UP instance discovery of novice evaluators.
- Hypothesis 3a.3 (H3b.3) – Tool support for diagnosis will increase the quality of novice evaluators’ descriptions of UP instances as rated by usability practitioners (judges in this study).

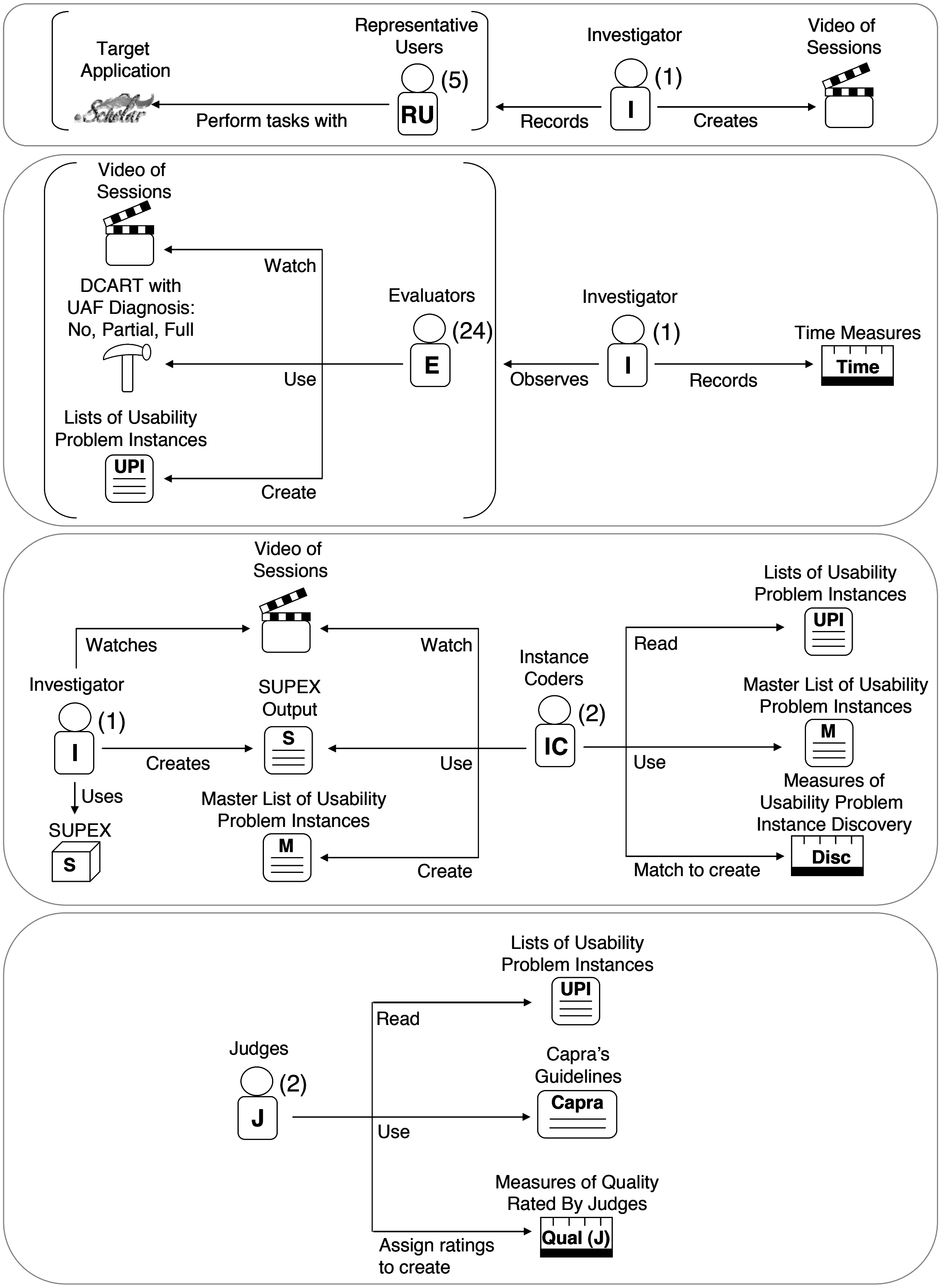


Figure 46: An overview of study 2. The numbers in parentheses indicate the number of individuals in each role.

5.2.2 Method

5.2.2.1 Overview

The evaluators in this study watched videos of representative users performing tasks with Scholar and produced lists of UP instances using DCART and one of the following levels of UAF diagnosis: no diagnosis, partial diagnosis, and full diagnosis. I recorded time data while the evaluators created their lists of UP instances. Instance coders compared the UP instances to a master list of UP instances to create measures of UP discovery. Judges rated the lists of UP instances from the perspective of a usability practitioner to create measures of quality.

5.2.2.2 Participants

I solicited participants for this study in the same manner as I did for study 1 (including the same requirements regarding UE experience); see Section 5.1.2.2 for details. The 8 evaluators in the DCART treatment for study 1 represented the no UAF diagnosis treatment for this study. As a result, 24 evaluators participated in study 2, but I only recruited 16 specifically for the partial-diagnosis and full-diagnosis treatments. The individuals recruited for the partial-diagnosis and full-diagnosis treatments were novice UAF users.

Of the 24 evaluators who participated in the study, 15 of the were students in the Department of Computer Science, 8 were students in the Department of Industrial and Systems Engineering, and 1 was a student in the Department of Biomedical Engineering. 19 had experience with course management systems, but none had ever used Scholar, the course management system used in the study.

5.2.2.3 Materials and Equipment

The materials and equipment are identical to those of study 1 (Section 5.1.2.3).

5.2.2.4 Procedure

I filtered evaluators and placed them into one of two treatment conditions via a background survey (Appendix B.2). In one treatment, evaluators used DCART with support for partial UAF diagnosis to conduct a usability evaluation; in the other treatment, evaluators used DCART with support for full UAF diagnosis to conduct a usability evaluation. As explained in Section 5.2.2.2, the evaluators in the no-diagnosis treatment were the evaluators in the DCART treatment for study 1. I notified evaluators who had been selected to participate in the study via email and had them choose a date and time that was convenient for them from a list of available dates and times. Each evaluator participated in one study session that lasted no more than two and a half hours. Evaluators participated individually; each study session consisted of only one evaluator. Figure 47 is an excerpt from Figure 46 that shows only the evaluator role.

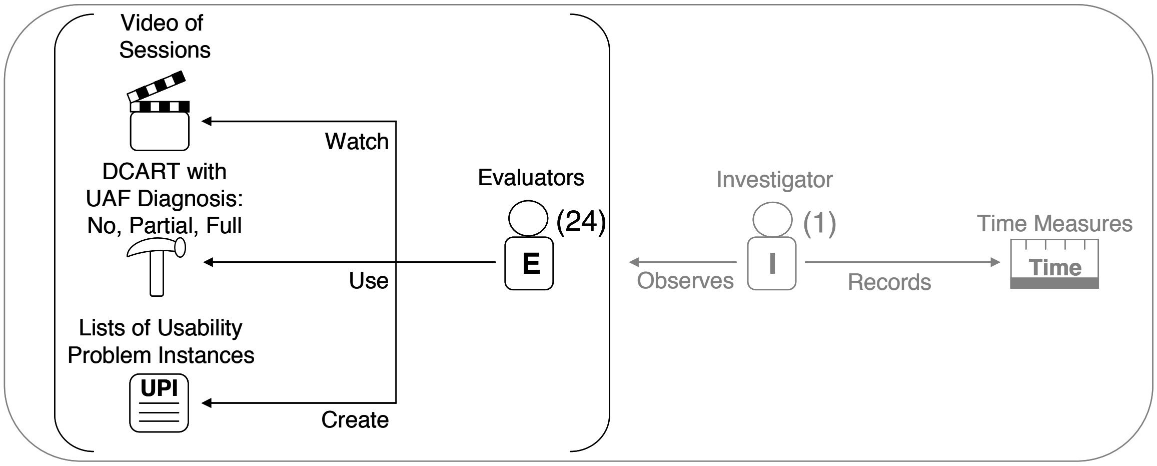


Figure 47: An excerpt from Figure 46 of the evaluator role. The numbers in parentheses indicate how many individuals participated in each role. Activities and objects related to the investigator role are grayed out.

When they arrived for the study, the evaluators read an informed consent form (Appendix B.3) and were given the chance to ask questions about the study. Evaluators who agreed to participate in the study signed the informed consent form.

After they had signed the consent form, the evaluators received a printed instruction booklet that was specific to the level of UAF diagnosis that they would be performing (partial diagnosis – Appendix B.11; full diagnosis – Appendix B.12). During the first hour, the evaluators performed activities to familiarize themselves with DCART and the steps involved with performing a usability evaluation. During the next one and a half hours, the evaluators performed a usability evaluation of Scholar.

The following are the activities that the evaluators performed during the first hour of the study session:

1. The evaluators watched a tutorial video on DCART and the level of diagnosis that they would be performing.
2. I explained the concept of UP instances to evaluators and gave them a printed diagram to show how raw usability data relates to UP instances (Appendix B.4).

3. The evaluators performed a practice usability evaluation of the Internet Movie Database (IMDB) website.

    a. The evaluators watched a video of a correct way to perform a task in the IMDB.

    b. The evaluators watched a video of a user trying to perform the task and used DCART to record UP instances experienced by the user. The evaluators watched the video one time through without pausing or stopping to simulate conducting a usability evaluation in real time. Thereafter, they were allowed to rewind, play, fast forward, pause, and stop the video as much as they needed.

    c. The evaluators in the partial-diagnosis and full-diagnosis treatments diagnosed the UPs described in their UP instances.

    d. The evaluators generated a Word document of their lists of UP instances and compared their list to a sample list specific to their level of UAF diagnosis (partial diagnosis – Appendix B.13; full diagnosis – Appendix B.14). I spoke with them and gave them feedback on the UP instances that they had recorded.

The following are the activities that the evaluators performed during the next one and a half hours of the study session:

- The evaluators performed a usability evaluation of Scholar.

    a. The evaluators watched a video that introduced Scholar, a video of a correct way to add a student to a course, and a video of a correct way to remove a student from a course.

    b. The evaluators watched a video of a user trying to add a student, a video of a second user trying to add a student, and a video of the first user trying to remove a student. The evaluators used DCART to record UP instances experienced by the users. The evaluators watched the three videos one time through without pausing or stopping to simulate conducting a usability evaluation in real time. Thereafter, they were allowed to rewind, play, fast forward, pause, and stop the videos as much as they needed.

    c. The evaluators in the partial-diagnosis and full-diagnosis treatments diagnosed the UPs described in their UP instances.

    d. The evaluators generated a Word document of their list of UP instances.

The evaluators used the session, task run, UP instance collection, UP instance review, and UP record forms in DCART (Section 4.3.2.2). I created the necessary session and task run objects for the evaluators. For this study, the UP collection and review forms had fields for capturing immediate intention for both the partial-diagnosis and full-diagnosis treatments, and the UP record had a field for UAF diagnosis in the full-diagnosis treatment. The evaluators used a function built into DCART to generate a Word document of UP instances from the UP instance records that they had created.

5.2.2.5 Experimental Design

This study was a between-subjects design with the level of diagnosis (no diagnosis, partial diagnosis, or full diagnosis) as the independent variable. The dependent variables were time measures and measures of UP instance discovery and UP instance quality as rated by usability practitioners (judges in this study). See section 5.1.2.5 for additional information on the rationale for this design.

As with study 1, I anticipated that performance would be most closely related to basic knowledge (UE or HCI), experience with course management software, and English language skills. I filtered participants using the online questionnaire mentioned in the procedure (Section 5.2.2.4) and assigned participants, so that they were as evenly distributed between treatments as possible (Table 23).

Table 23: Matching of evaluators for Study 2

Treatment	UE Experience with or without HCI Experience	HCI Experience without UE Experience	CM Software Experience	Fluent in English
No-diagnosis	5	3	6	6
Partial-diagnosis	6	2	7	7
Full-diagnosis	8	0	6	7

HCI = human-computer interaction, CM = course management. The cell values indicate the number of participants that met each criterion. There were 8 participants per treatment. The values in the rows sum to more than 8 because the columns are not mutually exclusive. For example, an individual with UE experience might also have CM software experience and be fluent in English.

5.2.2.6 Data Collection and Analysis

The time measures and measures of UP instance discovery and UP instance quality as rated by usability practitioners (judges in this study) were identical to those of study 1 (Section 5.1.2.6).

5.2.3 Results

5.2.3.1 Hypothesis 3b.1

I hypothesized that tool support for diagnosis would not affect the time that it takes novice evaluators to perform evaluations. I calculated two measures of resource use: the amount of time that the evaluators spent performing the evaluation and whether evaluators finished (Section 5.1.2.6.1).

5.2.3.1.1 Time

Figure 48 illustrates mean time values by treatment.

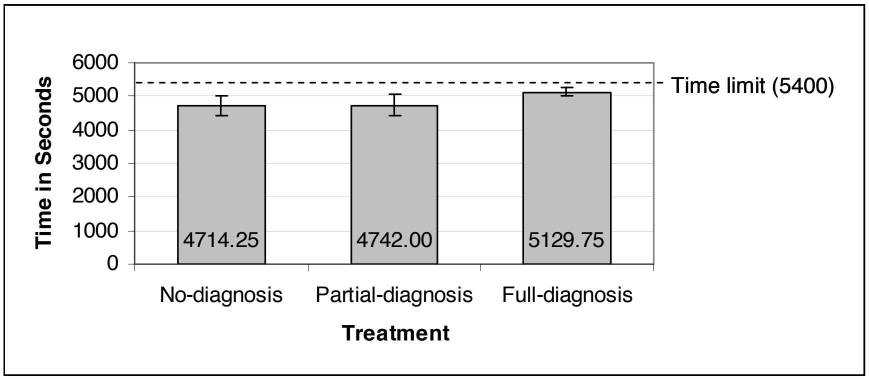


Figure 48: Study 2 - Mean time value by treatment, bars represent standard error

A histogram of the time values suggested that there was a ceiling effect due to the one and a half hour time limit on the evaluators. A Shapiro-Wilk test was performed on the time values; the null hypothesis that they came from a normal distribution was rejected, *W*=0.74, *p*<0.01. As a result, Wilcoxon rank-sum tests, non-parametric tests, were performed instead of *t*-tests, parametric tests. Using a normal approximation procedure, the tests indicated that there were not significant differences in the medians between pairs of treatments: no diagnosis and partial-diagnosis, *W*=67, *p*=0.96; no-diagnosis and full-diagnosis, *W*=73, *p*=0.63; partial-diagnosis and full-diagnosis, *W*=80, *p*=0.23. Table 23 contains descriptive statistics for the time values.

Table 24: Study 2 - Time value by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
No-diagnosis	4714.25	842.97	3386	5400
Partial-diagnosis	4742.00	881.05	2600	5400
Full-diagnosis	5129.75	366.31	4323	5400

Cell values represent time in seconds, n=8 per treatment

5.2.3.1.2 Finished

Figure 49 shows the counts of evaluators who finished by treatment.

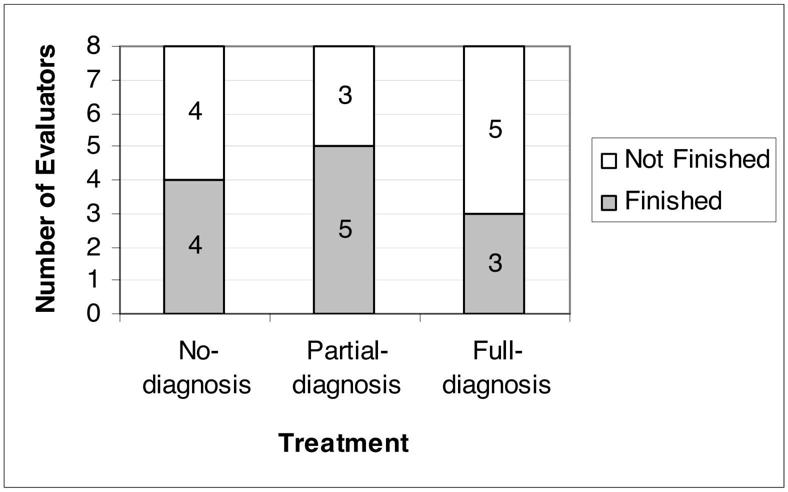


Figure 49: Study 2 - Finished count by treatment

Fisher's Exact test was used to analyze the differences in finishing and not finishing the evaluation among evaluators between treatments. Fisher's Exact test is better suited for this analysis than chi-square tests such as Pearson's Chi-square test or G-tests such as the Likelihood Ratio test because sample sizes are small. The differences in the number of evaluators who finished between the no-diagnosis and partial-diagnosis treatments, *p*=1.00; no-diagnosis and full-diagnosis treatments, *p*=1.00; and partial-diagnosis and full-diagnosis treatments *p*=0.62 were not significant.

5.2.3.2 Hypothesis 3b.2

I hypothesized that tool support for diagnosis would not affect the UP instance discovery of novice evaluators. I calculated four measures of UP instance discovery: discovery thoroughness, discovery validity, discovery effectiveness, and discovery reliability (Section 5.1.2.6.2). Figure 50 illustrates the mean values for these measures.

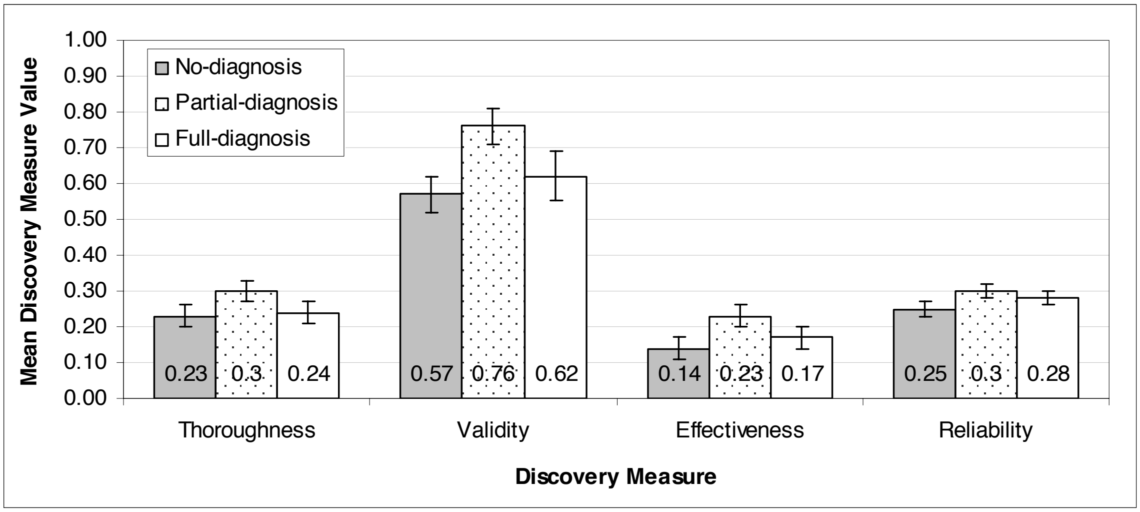


Figure 50: Study 2 - Mean discovery thoroughness, discovery validity, discovery effectiveness, and discovery reliability measures by treatment, bars represent standard error

5.2.3.2.1 Discovery Thoroughness

A Shapiro-Wilk test was performed on the discovery thoroughness values; the null hypothesis that they came from a normal distribution could not be rejected, W=0.96, p=0.39. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, F(2)=0.07, p=0.93. A difference in the mean discovery thoroughness value among all treatments was tested as part of an ANOVA; there was no significant difference among the means, F(2, 21)=1.07, p=0.36. Table 25 contains descriptive statistics for the discovery thoroughness values.

Table 25: Study 2 - Discovery thoroughness by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
No-diagnosis	0.23	0.09	0.11	0.37
Partial-diagnosis	0.30	0.09	0.18	0.47
Full-diagnosis	0.24	0.08	0.11	0.34

n=8 per treatment

5.2.3.2.2 Discovery Validity

A Shapiro-Wilk test was performed on the discovery validity values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.97, *p*=0.69. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(2)=0.43, *p*=0.66. A difference in the mean discovery validity value among all treatments was tested as part of an ANOVA; there was no significant difference among the means, *F*(2, 21)=2.68, *p*=0.09. Table 26 contains descriptive statistics for the discovery validity values.

Table 26: Study 2 - Discovery validity by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
No-diagnosis	0.57	0.14	0.38	0.74
Partial-diagnosis	0.76	0.15	0.57	1.00
Full-diagnosis	0.62	0.21	0.29	0.93

n=8 per treatment

5.2.3.2.3 Discovery Effectiveness

A Shapiro-Wilk test was performed on the discovery effectiveness values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.96, *p*=0.41. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(2)=0.05, *p*=0.95. A difference in the mean discovery effectiveness value among all treatments was tested as part of an ANOVA; there was no significant difference among the means, *F*(2, 21)=1.88, *p*=0.18. Table 27 contains descriptive statistics for the discovery effectiveness values.

Table 27: Study 2 - Discovery effectiveness by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
No-diagnosis	0.14	0.09	0.04	0.27
Partial-diagnosis	0.23	0.09	0.12	0.34
Full-diagnosis	0.17	0.10	0.03	0.32

n=8 per treatment

5.2.3.2.4 Discovery Reliability

A Shapiro-Wilk test was performed on the discovery reliability values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.98, *p*=0.21. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(2)=1.89, *p*=0.25. A difference in the mean discovery effectiveness value among all treatments was tested as part of an ANOVA; there was no significant difference among the means, *F*(2, 81)=1.40, *p*=0.25. Table 28 contains descriptive statistics for the discovery effectiveness values.

Table 28: Study 2 - Discovery effectiveness by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
No-diagnosis	0.25	0.11	0.06	0.50
Partial-diagnosis	0.30	0.09	0.12	0.50
Full-diagnosis	0.28	0.13	0.06	0.50

n=8 per treatment

5.2.3.3 Hypothesis 3b.3

I hypothesized that tool support for diagnosis would increase the quality of novice evaluators' descriptions of UP instances as rated by usability practitioners (judges in this study). The judges assigned quality ratings from the perspective of usability practitioners. I calculated two measures of quality: mean rating across all guidelines and mean rating per guideline (Section 5.1.2.6.3).

5.2.3.3.1 How the Judges Rated

Although differences in mean rating are of primary interest, an understanding of how the judges rated the lists of UP instances was useful in interpreting differences. I calculated measures of association, bias, and distribution for the judges.

Association

Association is the tendency of each judge to give higher/lower ratings to the same evaluator. Association was tested using Pearson's product-moment correlation both by treatment and by guideline; the results are in Table 29 and Table 30. Using an alpha level of 0.05, there was a significant correlation between the judges for the no-diagnosis treatment. The correlations were not significant for the partial-diagnosis and full-diagnosis treatments, which suggests that the judges had different underlying understandings of the role of diagnosis. Using an alpha level of 0.05, there was a significant correlation between the judges for all guidelines and *describe a solution*. The correlations were not significant for *be clear and precise*, *describe the impact*, *support with data*, *describe the cause*, and *describe observed actions*, which suggests that the judges used different underlying traits to form their judgments for these guidelines.

Table 29: Study 2 - Judge association by treatment, tested using Pearson's product-moment correlation

Treatment	*r*	*p*
No-diagnosis	0.32	0.03*
Partial-diagnosis	0.22	0.14
Full-diagnosis	0.23	0.12

n=48 per treatment, * *p*<0.05

Table 30: Study 2 - Judge association by guideline, tested using Pearson's product-moment correlation

Guideline	*r*	*P*
All guidelines	0.26	<0.01*
Be clear and precise	0.20	0.33
Describe the impact	-0.15	0.48
Support with data	0.12	0.57
Describe the cause	0.08	0.70
Describe observed actions	0.25	0.24
Describe a solution	0.74	<0.01*

n=24 per guideline, * $p<0.05$

Bias

Bias is the tendency of each judge to give higher or lower ratings overall. Bias was tested using a 3x6x2 mixed-factor ANOVA, with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure. The results of the same ANOVA were used to compare mean rating scores; see Section 5.2.3.3.2 for the full ANOVA results.

Using an alpha level of 0.05, the judge x treatment x guideline effect was not significant, *F*(10, 252)=0.71. The judge main effect was significant, *F*(1, 252)=13.41, *p*<0.01, but the judge x treatment interaction and judge x guideline interaction effects were also significant.

The judge x treatment interaction was significant, *F*(2, 252)=5.91, *p*<0.01. The judge x treatment interaction was explored using slices to test for simple effects due to judge for each treatment. There was an effect due to judge in the no-diagnosis and partial-diagnosis treatments; judge j1 gave higher ratings in both treatments. Figure 51 shows the judges' mean ratings by treatment.

Additionally, the judge x guideline interaction was significant, *F*(5, 252)=6.16, *p*<0.01. The judge x guideline interaction was explored using slices to test for simple effects due to judge for each guideline (Table 31). Judge j1 gave significantly higher ratings for the following guidelines (Table 32): *be clear and precise*, *support with data*, and *describe a solution*.

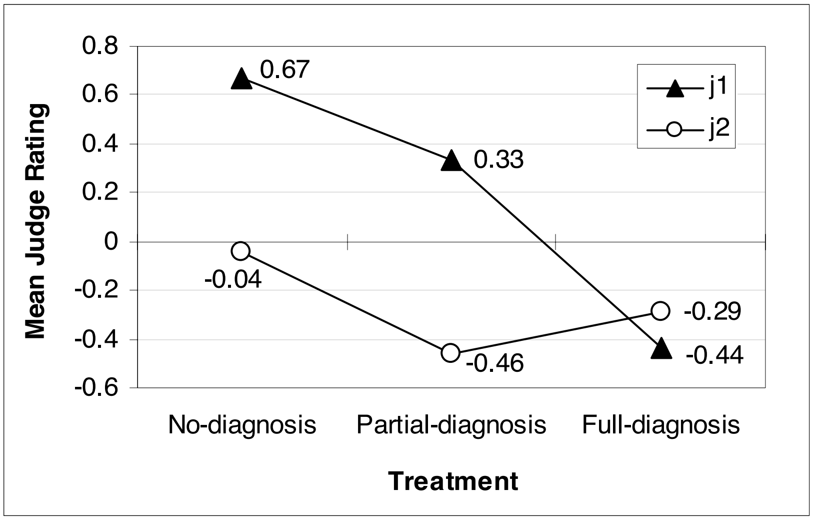


Figure 51: Study 2 - Judge bias, mean rating by judge by treatment

Table 31: Study 2 - Judge bias, simple effects due to judge for each guideline explored using slices

Guideline	*F*	*p*
Be clear and precise	4.29	0.04*
Describe the impact	1.54	0.22
Support with data	28.97	<0.01*
Describe the cause	0.17	0.68
Describe observed actions	0.02	0.89
Describe a solution	9.22	<0.01*

DF Num=1, DF Den=252, * *p*<0.05

Table 32: Study 2 - Judge bias, mean rating by judge by guideline

Judge	Be clear and precise	Support with data	Describe a solution
j1	0.33	0.38	1.21
j2	-0.29	-1.25	-1.25

Cell values represent mean ratings

Distribution

Distribution is the tendency of each judge to use each point in the scale. Distribution was assessed using visual inspection of the judges' ratings by treatment (Figure 52) and by guideline (Figure 53). Each judge gave 48 ratings per treatment. Judge j1 used the endpoints (strongly disagree and strongly agree) more frequently (n=28) than judge j2 (n=3). Judge j2 used the innermost points (somewhat disagree and somewhat agree) more frequently (n=104) than judge j1 (n=61). Judge j1 gave more positive ratings (any of the agree ratings) (n=80) than judge j2 (n=55).

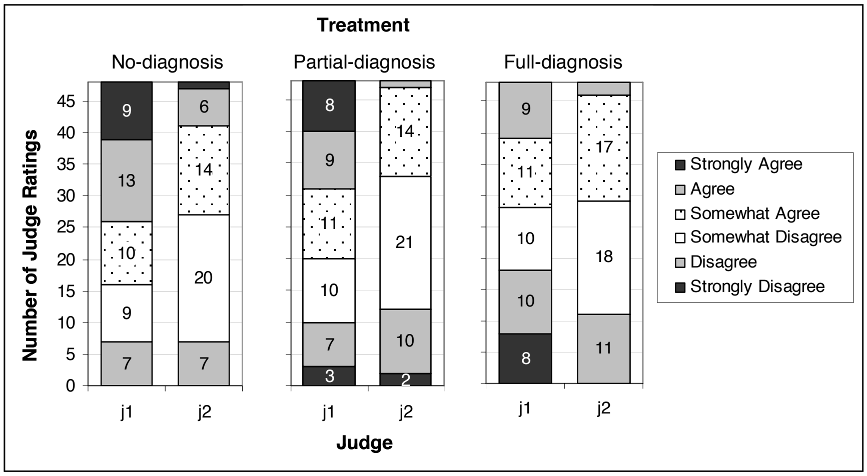


Figure 52: Study 2 - Judge distribution by treatment

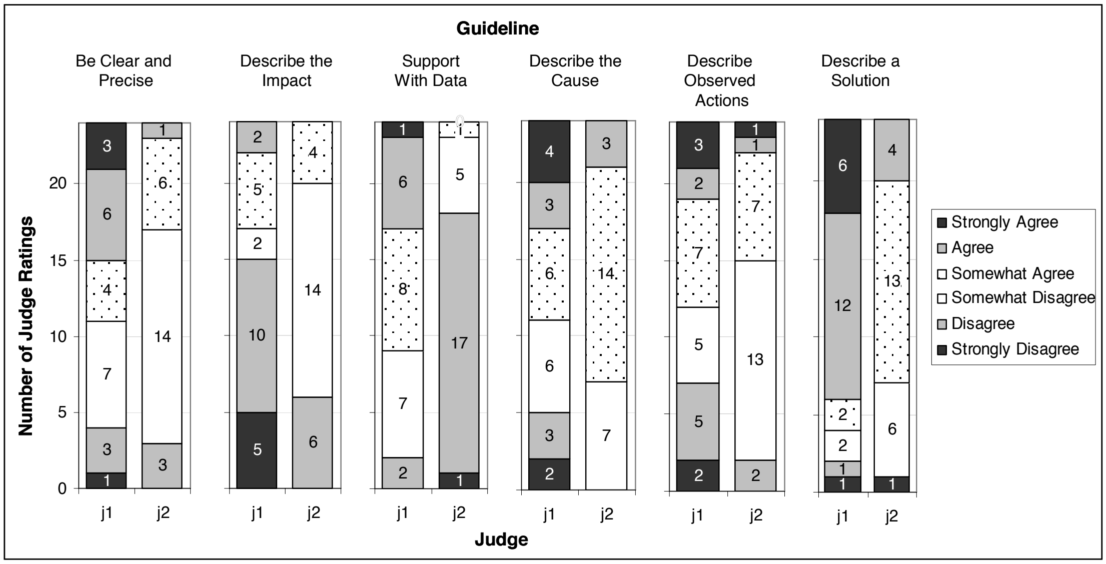


Figure 53: Study 2 - Judge distribution by guideline

5.2.3.3.2 Mean Ratings

The rating data are interval and can have only six values, so it is not possible to test for normality using the Shapiro-Wilk test. Instead, both a histogram and a normal quantile plot suggest that the rating data is jointly normally distributed.

Means are based on individual ratings given by each judge, rather than the sum of the two ratings. Judges rated on a 6-point scale, which has been adjusted to a rating from –2.5 to 2.5. Differences in mean rating across all guidelines by treatment were tested as part of a 3x6x2 mixed-factor ANOVA, with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure (Table 33). The effects specific to judge are discussed in the analysis of judge bias in Section 5.2.3.3.1.

Table 33: Study 2 - Quality as rated by judges, results of a 3x6x2 mixed factor ANOVA with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure

Source	*F*	DF Num	DF Den	*p*
Judge	13.41	1	252	<0.01*
Treatment	10.10	2	252	<0.01*
Guideline	12.44	5	252	<0.01*
Judge x Treatment	5.91	2	252	<0.01*
Judge x Guideline	6.16	5	252	<0.01*
Treatment x Guideline	0.62	10	252	0.80
Judge x Treatment x Guideline	0.71	10	252	0.71

N=288, * *p*<0.05

Treatment

The treatment main effect was explored using a Tukey test of least square means (Table 34). Using an alpha level of 0.05, the mean rating for the no-diagnosis treatment (*M*=0.31, *SD*=1.22) was significantly greater than for the partial-diagnosis (*M*=-0.06, *SD*=1.28) and full-diagnosis (*M*=-0.36, *SD*=1.14) treatments.

Table 34: Study 2 - Quality as rated by judges, treatment main effect explored using a Tukey test of least square means

Treatment	Least Square Mean
No-diagnosis	0.31_{a}
Partial-diagnosis	-0.06_{b}
Full-diagnosis	-0.36_{b}

Means that do not share a common letter differed significantly, α=0.05

Guideline

The guideline main effect indicated that some guidelines had mean ratings that were significantly different from other guidelines. This result was expected and was not of particular interest for this study.

Treatment x Guideline

Even though the treatment x guideline interaction effect was not significant, it was explored using slices for the purposes of the mean rating per guideline measure (Table 35). The judges gave significantly different ratings for *describe observed actions*. Figure 54 shows the mean ratings per guideline. The mean rating by treatment for the guideline were explored using a Tukey test of least square means. Using an alpha level of 0.05, the mean rating for the no-diagnosis treatment (*M*=0.56, *SD*=1.18) was significantly greater than for the partial-diagnosis (*M*=-0.25, *SD*=1.18) and full-diagnosis (*M*=-0.50, *SD*=1.03) treatments.

Table 35: Study 2 - Quality as rated by judges, simple effects due to guideline for each treatment explored using slices

Guideline	*F*	*p*
Be clear and precise	1.28	0.28
Describe the impact	1.16	0.31
Support with data	0.91	0.40
Describe the cause	2.53	0.08
Describe observed actions	4.51	0.01*
Describe a solution	2.80	0.06

DF Num=2, DF Den=252, * $p<0.05$

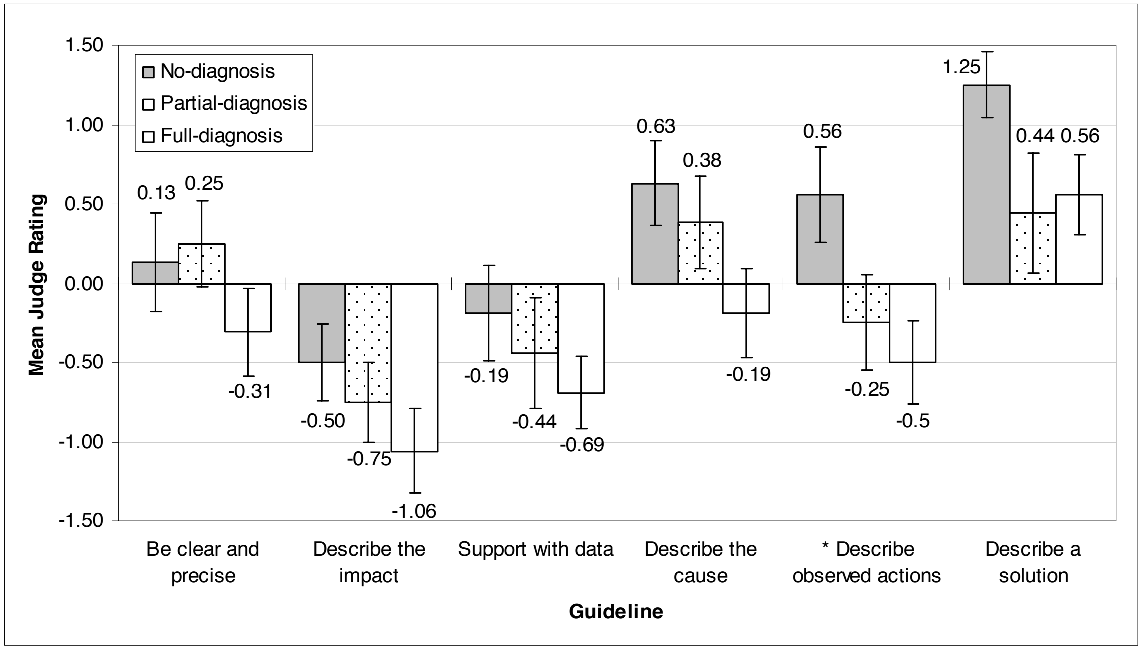


Figure 54: Study 2 - Quality as rated by judges, mean rating per guideline, bars represent standard error, * *p*<0.05

5.2.4 Discussion

Table 36 contains a summary of hypothesis testing results for study 2.

Table 36: Study 2 - Summary of Hypothesis Testing Results

Hypothesis	Result
H3b.1 – Tool support for diagnosis will not affect the time that it takes novice evaluators to perform evaluations.	**Supported** – There was not a significant difference in the time that it took evaluators to perform the evaluation or the number of evaluators who finished between treatments.
H3b.2 – Tool support for diagnosis will not affect the UP instance discovery of novice evaluators.	UP instance discovery measures were calculated by matching lists of UP instances produced by evaluators to a master list of UP instances. **Supported** – There were not significant differences between treatments for the thoroughness, validity, effectiveness, and reliability measures.
H3b.3 – Tool support for diagnosis will increase the quality of novice evaluators' descriptions of UP instances as rated by usability practitioners (judges in this study).	Measures of quality are based on Capra's guidelines [2006]. Higher mean ratings map to more agreement with the guideline(s) thereby indicating higher quality. **Not Supported** – The lists of UP instances produced by evaluators in the no-diagnosis treatment were rated by the judges to be of higher quality overall than those produced by evaluators in the partial-diagnosis and full-diagnosis treatments. There was not a significant difference in quality between the partial-diagnosis and full-diagnosis treatments.

5.2.4.1 Time

There was not a significant difference in the time that it took evaluators to perform the evaluation or the number of evaluators who finished between treatments. Diagnosis can be time consuming, particularly for individuals who are not UAF experts as was the case for the evaluators in this study. Nonetheless, I hypothesized that there would be no significant difference because I anticipated that evaluators who performed diagnosis would write shorter, more focused descriptions of UP instances. An ANOVA was used to test for a difference in mean description word count among all treatments (no-diagnosis *M*=22.09, *SD*=8.28; partial-diagnosis *M*=20.18, *SD*=3.47; full-diagnosis *M*=16.40, *SD*=4.37); there was not a significant difference, *F*(2, 21)=2.02, *p*=0.16. The results suggest that the evaluators in the partial-diagnosis and full-diagnosis treatments performed diagnosis and wrote UP instance descriptions in the same amount of time that it took evaluators in the no-diagnosis treatment to do just the writing.

One explanation is that performing diagnosis helped the evaluators in the partial-diagnosis and full-diagnosis treatments understand the UP instances, so that they were able to quickly describe them. Another explanation is that the evaluators focused on a certain subset of aspects of the description and excluded other aspects; a narrower focus would have made it easier for them to quickly describe UP instances. The discussion of the UP instance discovery and quality results suggests that the second explanation is more probable. Diagnosis with the UAF helps an evaluator focus on the cause of a UP; the evaluators in the partial-diagnosis and full-diagnosis treatments focused more on the underlying UP and less on the specific details unique to a given instance of the UP.

5.2.4.2 Usability Problem Instance Discovery

UP instance discovery measures were calculated by matching lists of UP instances produced by evaluators to a master list of UP instances. There were not significant differences between treatments for the thoroughness, validity, effectiveness, or reliability measures. The lack of a significant difference among treatments for the reliability measure supports the explanation of reliability from study 1; the evaluators in all three treatment conditions used DCART and its built-in support for UP instances and therefore had similar reliability. The model of the usability data collection stage (Figure 45) can be used to explain the lack of differences for validity, effectiveness, and reliability. The evaluators in the partial-diagnosis and full-diagnosis treatments were novice UAF users. As a result, they were unable to use diagnosis as a tool to help them understand the problem until the description stage. UAF experts, on the other hand, have internalized the UAF and use it even as early as recognition and definitely in formulation; knowledge of the UAF influences the number and types of critical incidents that they notice.

Piaget's work on schemas [Kalat, 1996] provides support for this explanation. Piaget defined schemas as mental representations of ideas, perceptions, and actions and considered them to be the fundamental building blocks of thinking. There are two basic processes involving schemas: assimilation and accommodation. Assimilation involves organizing existing schemas to better represent the external world, while accommodation involves modifying existing schemas or creating new ones to account for new ideas, perceptions, and actions. UAF experts who have internalized the UAF have developed schemas for understanding usability data based on the organization of usability concepts in the UAF. Novice UAF users who are also novice usability practitioners, however, have not developed these schemas. As a result, UAF experts are able to quickly organize usability data that fits into their existing schemas and assimilate new usability data that does not exactly fit. On the other hand, novice UAF users who are also novice usability practitioners may create new schemas as they work to accommodate usability data. The UAF experts' schemas increase their ability to notice critical incidents because they spend more time anticipating and less time accommodating usability data.

5.2.4.3 Quality as Rated by Judges

Measures of quality are based on Capra's guidelines [2006]. Higher mean ratings map to more agreement with the guideline(s) thereby indicating higher quality. The lists of UP instances produced by evaluators in the no-diagnosis treatment were rated by the judges to be of higher quality than those produced by evaluators in the partial-diagnosis and full-diagnosis treatments. There was not a significant difference in quality between the partial-diagnosis and full-diagnosis treatments.

I had not expected the partial-diagnosis and full-diagnosis treatments to receive lower ratings overall than the no-diagnosis treatment. In fact, to the contrary, one might expect that UP reports guided by more structure would yield more quality in the report. One explanation for this result is that the evaluators in the partial-diagnosis and full-diagnosis treatments focused on the cause and underlying type of the UPs documented in the UP instance records and did not provide details unique to the instances of the UPs. The UAF diagnosis path was included in UP instance records in the partial-diagnosis and full-diagnosis treatments. The evaluators in these treatments may have felt that this diagnostic information was adequate for describing a problem, while the judges expected the information to be integrated in the description of the UP instance. A post-hoc analysis of the guidelines by treatment provided support for this explanation; the judges rated the partial-diagnosis and full-diagnosis treatments significantly lower than the no-diagnosis treatment for *describe observed actions*.

An examination of the lists of UP instances provided specific examples of the lack of descriptions of observed actions. One representative user had difficulty determining whether a student's PID (personal ID number) was the same as the student's username in Scholar. An evaluator in the no-diagnosis treatment created a record for the UP instance that contained the following description: "While entering the new student into the system, the user is confused by the 'Username' field. All he has is the student's PID, and he doesn't know if that is the same thing as the username". An evaluator in the full-diagnosis treatment created a UP instance record with the following description: "The user was confused whether he had to add the pid or the name". The description from the evaluator in the no-diagnosis treatment provides information on what the representative user was doing when he encountered the UP, but the description from the evaluator in the full-diagnosis treatment only describes the cause of the problem.

5.2.4.4 Limitation of the Study

This study was subject to the same limitation as study 1. Please see Section 5.1.4.4 for details.

5.3 Study 3: Support for Merging and Grouping

Studies 1 and 2 focused on the ability of evaluators to identify and describe UP instances effectively. The primary outputs that were produced by evaluators in these studies were lists of UP instances. As discussed in Section 4.1.3, however, these lists are of limited value. Evaluators need to merge UP instances into UPs (Figure 5) and group UPs (Figure 5) to create usability evaluation reports that facilitate understanding of key usability issues by other individuals involved in the UE process. This study explored evaluator effectiveness from the perspective of both usability practitioners and developers.

As with studies 1 and 2, evaluator effectiveness was of primary interest. Because I assumed a fixed-resources environment, I wanted to remove efficiency as a point of consideration. To confirm this operating assumption, I recorded the amount of time that it took evaluators to perform the evaluations. Of interest with regard to effectiveness were measures of report quality as rated by usability practitioners and quality as rated by developers. Additionally, I interviewed each developer to get qualitative feedback on the usability evaluation reports.

Figure 55 is an overview of study 3; it shows roles and the tools and objects that people in the roles interacted with and produced. This figure is referenced in future sections that describe the various roles in more detail.

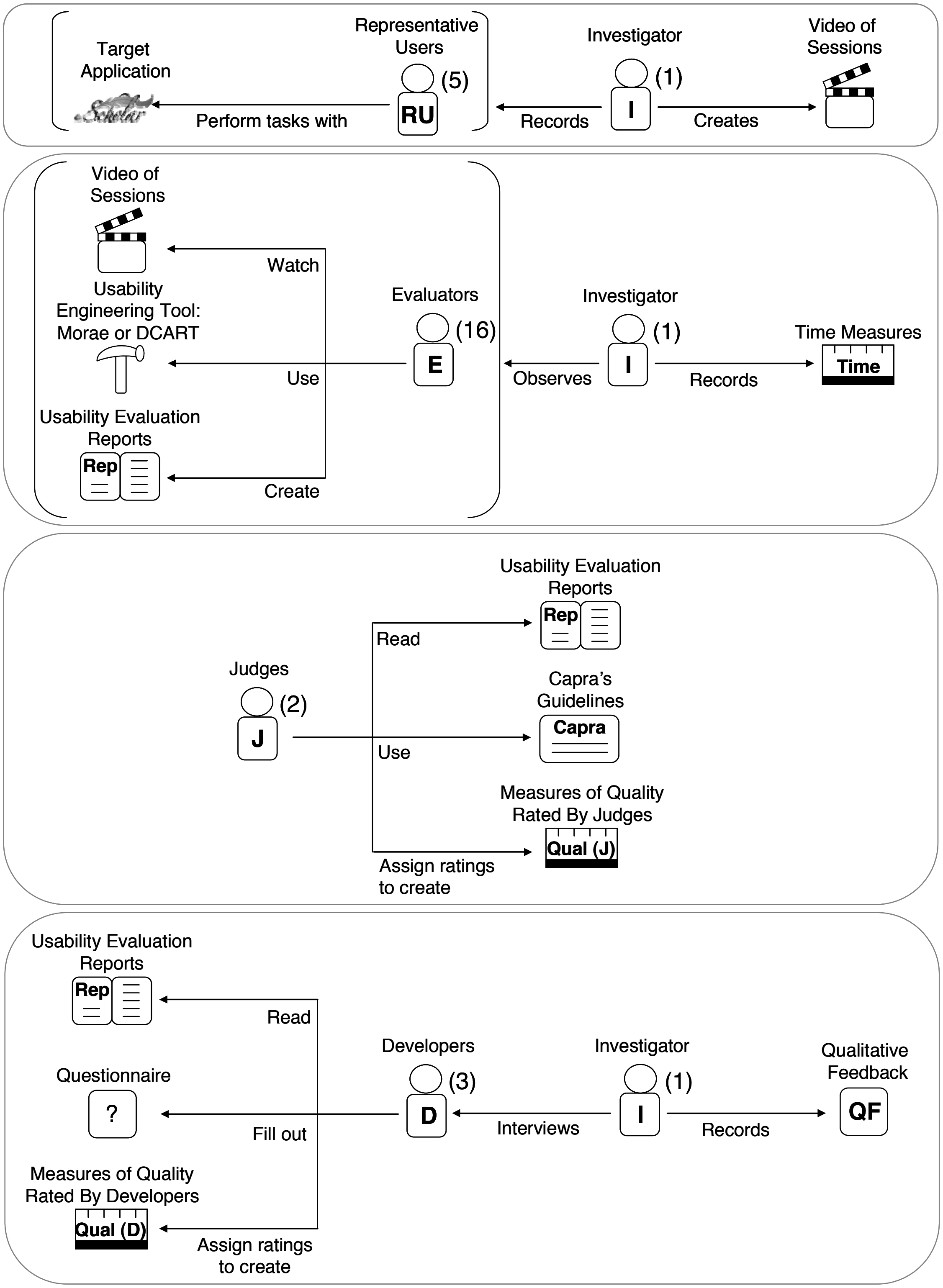


Figure 55: An overview of study 3. The numbers in parentheses indicate the number of individuals in each role.

5.3.1 Research Question and Hypotheses

The research question addressed by this study is directly related to RG3 in Section 1.5.

- RQ3c – How does tool support for merging UP instances and grouping UPs affect the effectiveness of evaluators?

The following hypotheses apply to RQ3c:

- Hypothesis 3c.1 (H3c.1) – Tool support for merging UP instances and grouping UPs will not affect the time that it takes novice evaluators to perform evaluations.
- Hypothesis 3c.2 (H3c.2) – Tool support for merging UP instances and grouping UPs will increase the quality of novice evaluators' usability evaluation reports as rated by usability practitioners (judges in this study).
- Hypothesis 3c.3 (H3c.3) – Tool support for merging UP instances and grouping UPs will increase the quality of novice evaluators' usability evaluation reports as rated by developers.

5.3.2 Method

5.3.2.1 Overview

The participants in this study watched videos of representative users performing tasks with Scholar, a course management system. These participants, whom I refer to as evaluators, produced usability evaluation reports using one of two usability engineering tools: Morae or DCART. Morae did not have support for merging UP instances and grouping UPs; DCART did. I recorded time data while the evaluators created their reports. Individuals with usability experience, whom I refer to as judges, rated the usability evaluation reports from the perspective of a usability practitioner to create measures of quality. The developers of Scholar also reviewed the reports and rated them to create measures of quality. I interviewed the developers after they had finished assigning ratings to get qualitative feedback on the usability evaluation reports.

5.3.2.2 Participants

I solicited participants for this study in the same manner as I did for study 1 (including the same requirements regarding UE experience); see Section 5.1.2.2 for details. I recruited a total of 16 participants. Fourteen of the participants were students in the Department of Computer Science and 2 were students in the Department of Electrical and Computer Engineering. Fourteen had experience with course management systems, but none had ever used Scholar, the course management system used in the study.

5.3.2.3 Materials and Equipment

The materials and equipment are identical to those of study 1 (Section 5.1.2.3).

5.3.2.4 Procedure

I filtered evaluators and placed them into one of two treatment conditions via a background survey (Appendix B.2). In one treatment, evaluators used Morae to conduct a usability evaluation; in the other treatment, evaluators used DCART to conduct a usability evaluation. I notified evaluators who had been selected to participate in the study via email and had them choose a date and time that was convenient for them from a list of available dates and times. Each evaluator participated in one study session that lasted no more than two and a half hours. Evaluators participated individually; each study session consisted of only one evaluator. Figure 56 is an excerpt from Figure 55 that shows only the evaluator role.

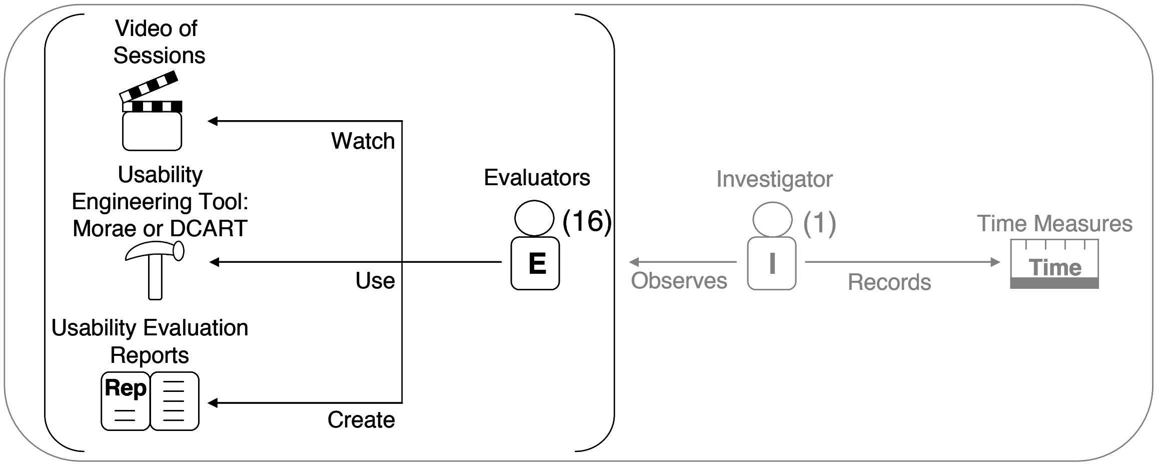


Figure 56: An excerpt from Figure 55 of the evaluator role. The numbers in parentheses indicate how many individuals participated in each role. Activities and objects related to the investigator role are grayed out.

When they arrived for the study, the evaluators read an informed consent form (Appendix B.3) and were given the chance to ask questions about the study. Evaluators who agreed to participate in the study signed the informed consent form.

After they had signed the consent form, the evaluators received a printed instruction booklet that was specific to the tool that they would be using (Morae – Appendix B.15; DCART – Appendix B.16). Regardless of the tool that they would be using, the evaluators followed the same basic process. During the first hour, the evaluators performed activities to familiarize themselves with their tool and the steps involved with performing a usability evaluation. During the next one and a half hours, the evaluators performed a usability evaluation of Scholar.

The following are the activities that the evaluators performed during the first hour of the study session:

1. The evaluators watched a tutorial video on their tool.
2. I explained the concept of UP instances to evaluators who used DCART and gave them a printed diagram to show how raw usability data relates to UP instances, how UP instances can be merged into UPs, and how UPs can be grouped for reporting purposes (Appendix B.5).
3. The evaluators performed a practice usability evaluation of the Internet Movie Database (IMDB) website.
    a. The evaluators watched a video of a correct way to perform a task in the IMDB.
    b. The evaluators watched a video of a user trying to perform the task and used their tool to record raw comments (Morae) or UP instances experienced by the user (DCART). The evaluators watched the video one time through without pausing or stopping to simulate conducting a usability evaluation in real time. Thereafter, they were allowed to rewind, play, fast forward, pause, and stop the video as much as they needed.
    c. The evaluators who used Morae consolidated their comments, and the evaluators who used DCART merged UP instances and grouped UPs.
    d. The evaluators created a usability evaluation report in a Word document and compared their report to a sample report specific to their tool (Morae – Appendix B.17; DCART – Appendix B.18). I spoke with them and gave them feedback on their reports.

The following are the activities that the evaluators performed during the next one and a half hours of the study session:

- The evaluators performed a usability evaluation of Scholar.
    a. The evaluators watched a video that introduced Scholar, a video of a correct way to add a student to a course, and a video of a correct way to remove a student from a course.
    b. The evaluators watched a video of a user trying to add a student, a video of a second user trying to add a student, and a video of the first user trying to remove a student. The evaluators used their tool to record raw comments (Morae) or UP instances experienced by

the user (DCART). The evaluators watched the three videos one time through without pausing or stopping to simulate conducting a usability evaluation in real time. Thereafter, they were allowed to rewind, play, fast forward, pause, and stop the videos as much as they needed.

c. The evaluators who used Morae consolidated their comments, and the evaluators who used DCART merged UP instances and grouped UPs.

d. The evaluators created usability evaluation reports in a Word document.

The Morae group evaluators made time-stamped comments using the observational capture features of Morae Remote Viewer while they watched the videos of representative users. They reviewed their comments, added new comments, and reviewed the video using Morae Manager. The evaluators edited and consolidated comments in Morae and then exported them to a Word document, exported comments to a Word document and then edited and consolidated them, or directly created the usability evaluation report in Word.

The DCART group evaluators used the session, task run, UP instance collection, UP instance review, and UP record forms (Section 4.3.2.2). I created the necessary session and task run objects for the DCART group evaluators. For this study, the UP collection and review forms did not have a field for capturing immediate intention, and the UP record did not have fields for immediate intention or UAF diagnosis. DCART users used built-in functions to merge UP instances into UPs and group UPs. They also used a function built into DCART to generate a usability evaluation report based on the UPs and groups of UPs that they had created. The majority of the evaluators in the DCART group modified the report generated by DCART.

5.3.2.5 Experimental Design

This study was a between-subjects design with the level of support for merging UP instances and grouping UPs (no support = freeform, used Morae or support = structured, used DCART) as the independent variable. The dependent variables were time measures and measures of usability evaluation report quality as rated by usability practitioners (judges in this study) and by developers. See section 5.1.2.5 for additional information on the rationale for this design.

As with study 1, I anticipated that performance would be most closely related to basic knowledge (UE or HCI), experience with course management software, and English language skills. I filtered participants using the online questionnaire mentioned in the procedure (Section 5.2.2.4) and assigned participants, so that they were as evenly distributed between treatments as possible (Table 37).

Table 37: Matching of evaluators for Study 3

Treatment	UE Experience with or without HCI Experience	HCI Experience without UE Experience	CM Software Experience	Fluent in English
Freeform	4	4	7	7
Structured	3	5	7	7

HCI = human-computer interaction, CM = course management. The cell values indicate the number of participants that met each criterion. There were 8 participants per treatment. The values in the rows sum to more than 8 because the columns are not mutually exclusive. For example, an individual with UE experience might also have CM software experience and be fluent in English.

5.3.2.6 Data Collection and Analysis

The time measures were identical to those of study 1 (Section 5.1.2.6). Measures of quality rated by usability practitioners (judges in this study) were also identical, except they were applied to usability evaluation reports instead of lists of UP instances. There were no measures of UP instance discovery. Additionally, this study included a measure of quality as rated by developers and qualitative data from semi-structured interviews with developers.

5.3.2.6.1 Measures of Quality as Rated by Developers

As discussed in Section 4.1.3, a more recent focus of usability research is communicating usability information in a manner that is useful to other members of the usability engineering process. I included developer input via quality ratings to get the developers' feedback on the usability evaluation reports produced by the evaluators.

Previous studies have included developer input. Hoegh et al. [2006] (which also includes previous work by Nielsen et al. [2005]) interviewed developers to obtain feedback on observation of user tests and usability evaluation reports. Hornbæk and Frøkjær [2005] interviewed developers regarding the utility of redesign proposals. Additionally, Law [2006] worked with developers to gather feedback on factors that influenced which usability problems the developers fixed. This study is similar to previous studies in that I am interested in the developers' feedback on the utility of the usability evaluation reports. This study differs from the ones performed by Hoegh et al. and Hornbæk and Frøkjær in that I am comparing different processes for producing usability evaluation reports instead of comparing usability evaluation reports to other forms of feedback. This study differs from the work by Law in that it focuses more on how the usability evaluation reports are produced as opposed to why developers interpret some reports to be better than others.

A number of steps were involved in calculating measures of quality as rated by the developers. First, I created a questionnaire based on the modified set of Capra's guidelines introduced in Section 5.1.2.6.3. Next, three developers from the Scholar development team used the questionnaires to rate the usability evaluation reports produced by evaluators. Finally, the ratings were used as inputs to calculate the measures. Figure 57 is an excerpt from Figure 55 that shows the developer role.

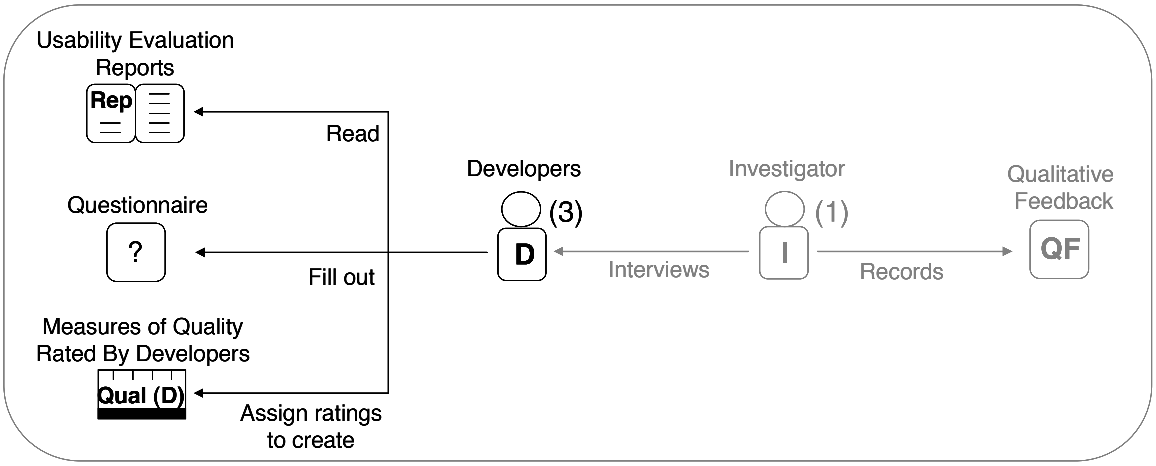


Figure 57: An excerpt from Figure 55 of the developer role. The number in parentheses indicates the number of individuals in the role. Activities and objects related to the investigator role are grayed out.

I did not pay the developers who participated in the study. In exchange for their involvement in my dissertation studies, I performed a formative usability evaluation of Scholar, produced a report, and presented the results at a Sakai conference.

After performing the usability evaluation and presenting the results, I met with the developers to explain the process that they would use to rate the usability evaluation reports produced by the evaluators. I asked the developers to read and sign a consent form (Appendix G.1). After they had signed the consent form, I gave them an instruction booklet (Appendix G.2) that detailed the task that they would perform for the study.

The developers watched the same videos of Scholar as the evaluators watched during their study sessions (Section 5.1.2.4). The developers' questionnaire was in the form of a spreadsheet (Appendix G.3). The developers worked independently and viewed the evaluators’ usability evaluation reports in different orders.

The questionnaire was designed to provide a view of the quality of the usability evaluation reports from the perspective of the developers. Questions 1 through 6 provided information on the quality and mapped to the six guidelines in Section 5.1.2.6.3. Question 7 was a summary question that was intended to get a measure of a developer’s overall opinion of the usefulness of a usability evaluation report. I calculated measures of quality across all six questions and per question. Additionally, I calculated the developer's mean rating on the summary question.

5.3.2.6.2 Qualitative Feedback

I interviewed the developers after they had finished assigning ratings. I interviewed each developer individually; each interview lasted between 30 and 45 minutes. I used a semi-structured interview approach consisting of the following topics: overall impressions, what the developer looked for in good usability evaluation reports, thoughts on the use of video data to accompany textual descriptions, and thoughts on Capra's guidelines as they were included in the questionnaire. Figure 58 is an excerpt from Figure 55 that shows the investigator role in conducting the interviews.

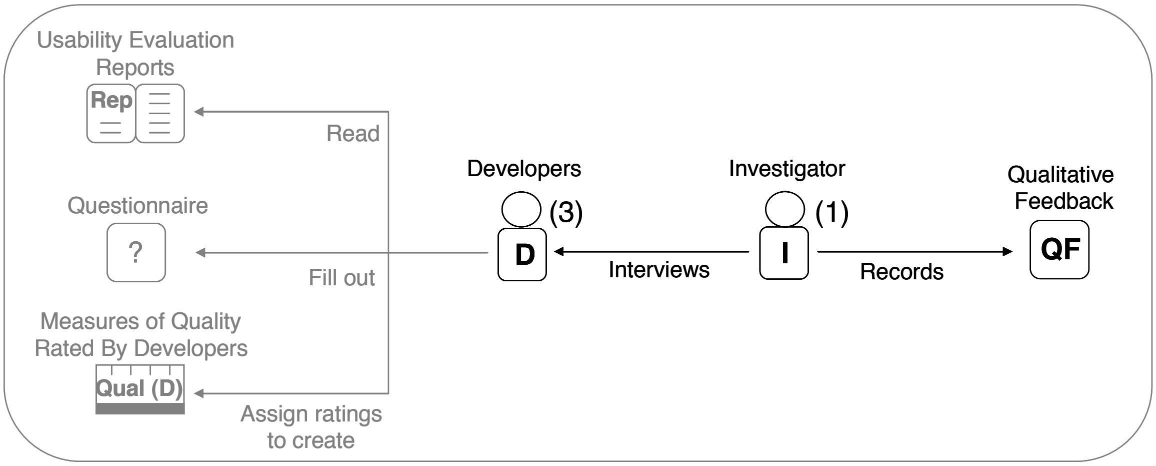


Figure 58: An excerpt from Figure 54 of the investigator role. The number in parentheses indicates the number of individuals in the role. Activities and objects related to the investigator role are grayed out.

I included interviews as a way to confirm or cross-validate the results of the quantitative analyses on the developers' ratings. I followed the general procedures for qualitative data collection and analysis recommended by Creswell [2003]. I first developed an interview protocol for recording data during the interviews. I then conducted an interview with each of the three developers. Thereafter, I typed up my handwritten notes from the interviews and read through them to obtain a general sense of the data. Next, I identified major themes and grouped the data by the themes. Finally, I developed each theme and made specific references to the interview data as appropriate.

5.3.3 Results

5.3.3.1 Hypothesis 3c.1

I hypothesized that tool support for merging UP instances and grouping UPs would not affect the time that it takes novice evaluators to perform evaluations. I calculated two measures of resource use: the amount of time that the evaluators spent performing the evaluation and whether evaluators finished (Section 5.1.2.6.1).

5.3.3.1.1 Time

Figure 59 illustrates mean time values by treatment.

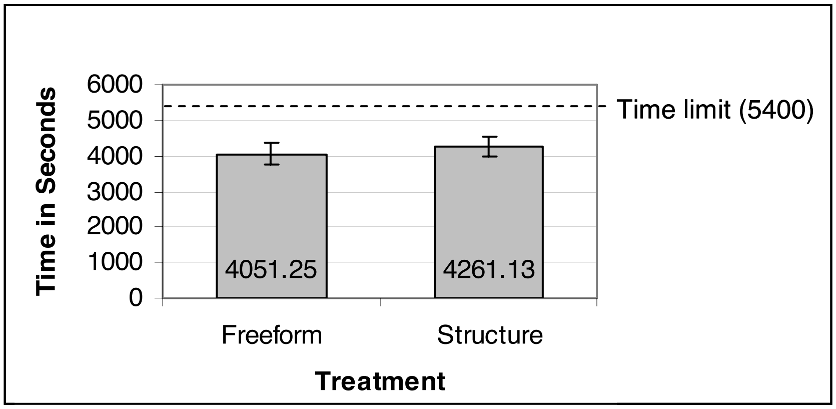


Figure 59: Study 3 - Mean time value by treatment, bars represent standard error

A Shapiro-Wilk test was performed on the time values; the null hypothesis that they came from a normal distribution could not be rejected, *W*=0.93, *p*=0.29. Additionally, Bartlett's test was performed; the null hypothesis that the error variances were equal could not be rejected, *F*(1)=<0.01, *p*=0.92. A t-test assuming equal variances indicated that there was not a significant difference between the treatment means, *t*(14)=0.48, *p*=0.64. Table 38 contains descriptive statistics for the time values.

Table 38: Study 3 - Time value by treatment, descriptive statistics

Treatment	M	SD	Lower	Upper
Freeform	4051.25	901.60	2301	5122
Structure	4261.13	864.87	3109	5243

Cell values represent time in seconds, n=8 per treatment

5.3.3.1.2 Finished

Figure 60 shows the counts of evaluators who finished by treatment.

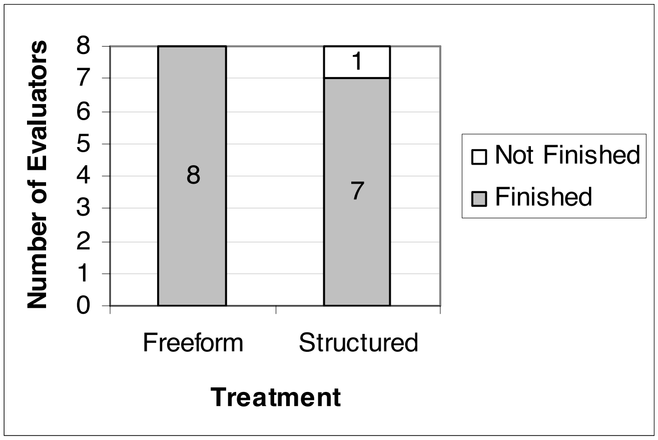


Figure 60: Study 3 - Finished count by treatment

Fisher’s Exact test was used to analyze the differences in finishing and not finishing the evaluation among evaluators between treatments. Fisher’s Exact test is better suited for this analysis than chi-square tests such as Pearson’s Chi-square test or G-tests such as the Likelihood Ratio test because sample sizes are small. The differences in the number of evaluators who finished between the freeform and structure treatments, *p*=1.00, was not significant.

5.3.3.2 Hypothesis 3c.2

I hypothesized that tool support for merging UP instances and grouping UPs would increase the quality of novice evaluators’ usability evaluation reports as rated by usability practitioners (judges in this study). I calculated two measures of quality: mean rating across all guidelines and mean rating per guideline (Section 5.1.2.6.3).

5.3.3.2.1 How the Judges Rated

Although differences in mean rating are of primary interest, an understanding of how the judges rated the lists of UP instances was useful in interpreting differences. I calculated measures of association, bias, and distribution for the judges.

Association

Association is the tendency of each judge to give higher/lower ratings to the same evaluator. Association was tested using Pearson's product-moment correlation both by treatment and by guideline; the results are in Table 39 and Table 40. Using an alpha level of 0.05, there was a significant correlation between the judges for both treatments. Using an alpha level of 0.05, there was a significant correlation between the judges for all guidelines, *describe the impact*, *describe observed actions*, and *describe a solution*. The correlations were not significant for *be clear and precise*, *support with data*, and *describe the cause*, which suggests that the judges used different underlying traits to form their judgments for these guidelines.

Table 39: Study 3 - Judge association by treatment, tested using Pearson's product-moment correlation

Treatment	*r*	*p*
Freeform	0.71	<0.01*
Structured	0.55	<0.01*

n=48 per treatment, * *p*<0.05

Table 40: Study 3 - Judge association by guideline, tested using Pearson's product-moment correlation

Guideline	*r*	*p*
All guidelines	0.64	<0.01*
Be clear and precise	0.33	0.21
Describe the impact	0.64	<0.01*
Support with data	0.46	0.07
Describe the cause	0.46	0.07
Describe observed actions	0.66	0.01*
Describe a solution	0.77	<0.01*

n=16 per guideline, * *p*<0.05

Bias

Bias is the tendency of each judge to give higher or lower ratings overall. Bias was tested using a 2x6x2 mixed-factor ANOVA, with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure. The results of the same ANOVA were used to compare mean rating scores; see Section 5.3.3.2.2 for the full ANOVA results.

The judge main effect was significant, $F(1, 168)=24.97$, $p<0.01$. The effect was explored using a *t*-test of least square means; the mean rating for judge j1 ($M=0.81$, $SD=1.34$) was significantly greater than the mean rating for judge j2 ($M=0.08$, $SD=0.85$), $t(168)=-5.00$, $p<0.01$.

Using an alpha level of 0.05, the judge x treatment interaction was not significant, $F(1, 168)=0.61$, nor was the judge x guideline interaction, $F(5, 168)=0.57$. Additionally, the judge x treatment x guideline interaction was not significant, $F(5, 168)=0.52$.

Distribution

Distribution is the tendency of each judge to use each point in the scale. Distribution was assessed using visual inspection of the judges' ratings by treatment (Figure 61) and by guideline (Figure 62). Each judge gave 48 ratings per treatment. Judge j1 used the endpoints (strongly disagree and strongly agree) more frequently (n=23) than judge j2 (n=8). Judge j2 used the innermost points (somewhat disagree and somewhat agree) more frequently (n=59) than judge j1 (n=31). Judge j1 gave more positive ratings (any of the agree ratings) (n=68) than judge j2 (n=48).

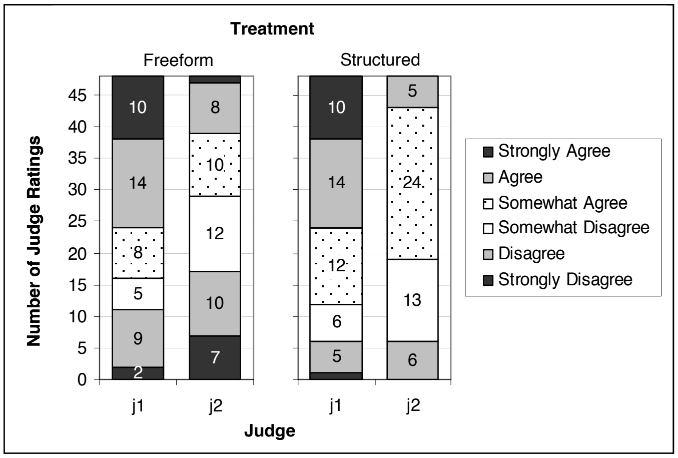


Figure 61: Study 3 - Judge distribution by treatment

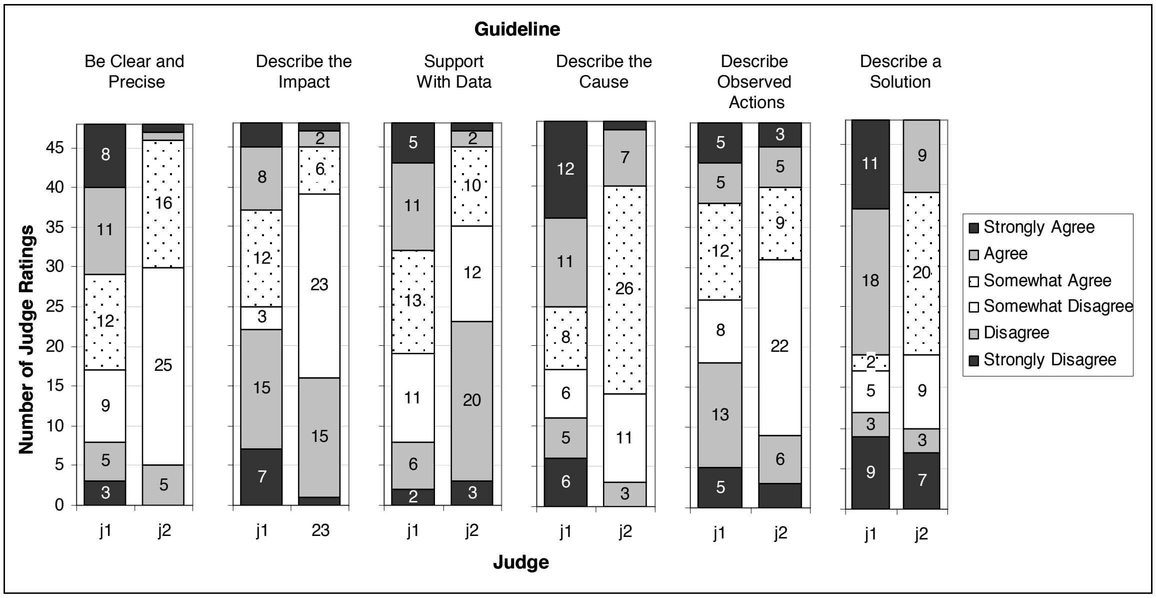


Figure 62: Study 3 - Judge distribution by guideline

5.3.3.2.2 Mean Ratings

The rating data are interval and can have only six values, so it is not possible to test for normality using the Shapiro-Wilk test. Instead, both a histogram and a normal quantile plot suggest that the rating data is jointly normally distributed.

Means are based on individual ratings given by each judge, rather than the sum of the two ratings. Judges rated on a 6-point scale, which has been adjusted to a rating from –2.5 to 2.5. Differences in mean rating across all guidelines by treatment were tested as part of a 2x6x2 mixed-factor ANOVA, with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure (Table 41). The effects specific to judge are discussed in the analysis of judge bias in Section 5.3.3.2.1.

Table 41: Study 3 - Quality as rated by judges, results of a 2x6x2 mixed factor ANOVA with treatment as a between-subject factor, guideline and judge as within-subject factors, and evaluator as a repeated measure

Source	*F*	DF Num	DF Den	*p*
Judge	24.97	1	168	< 0.01*
Treatment	3.95	1	168	< 0.05*
Guideline	7.36	5	168	< 0.01*
Judge x Treatment	0.61	1	168	0.43
Judge x Guideline	0.57	5	168	0.72
Treatment x Guideline	2.02	5	168	0.08
Judge x Treatment x Guideline	0.85	5	168	0.52

N=192, * *p*<0.05

Treatment

The mean rating for structured treatment (*M*=0.45, *SD*=1.17) was significantly greater than for the freeform treatment (*M*=0.10, *SD*=1.54) (Figure 63).

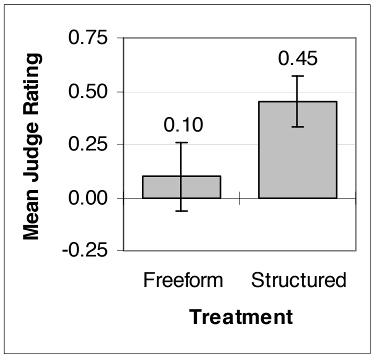


Figure 63: Study 3 - Quality as rated by judges, mean rating per treatment, bars represent standard error

Guideline

The guideline main effect indicated that some guidelines had mean ratings that were significantly different from other guidelines. This result was expected and was not of particular interest for this study.

Treatment x Guideline

Even though the treatment x guideline interaction effect was not significant, it was explored using slices for the purposes of the mean rating per guideline measure (Table 42). The judges gave significantly higher ratings for *support with data* and *describe a solution* for the structured treatment. Figure 64 shows the mean ratings per guideline.

Table 42: Study 3 - Quality as rated by judges, simple effects due to guideline for each treatment explored using slices

Guideline	*F*	*p*
Be clear and precise	0.20	0.66
Describe the impact	0.09	0.77
Support with data	4.89	0.03*
Describe the cause	1.07	0.30
Describe observed actions	0.78	0.38
Describe a solution	7.05	<0.01*

DF Num=1, DF Den=168, * *p*<0.05

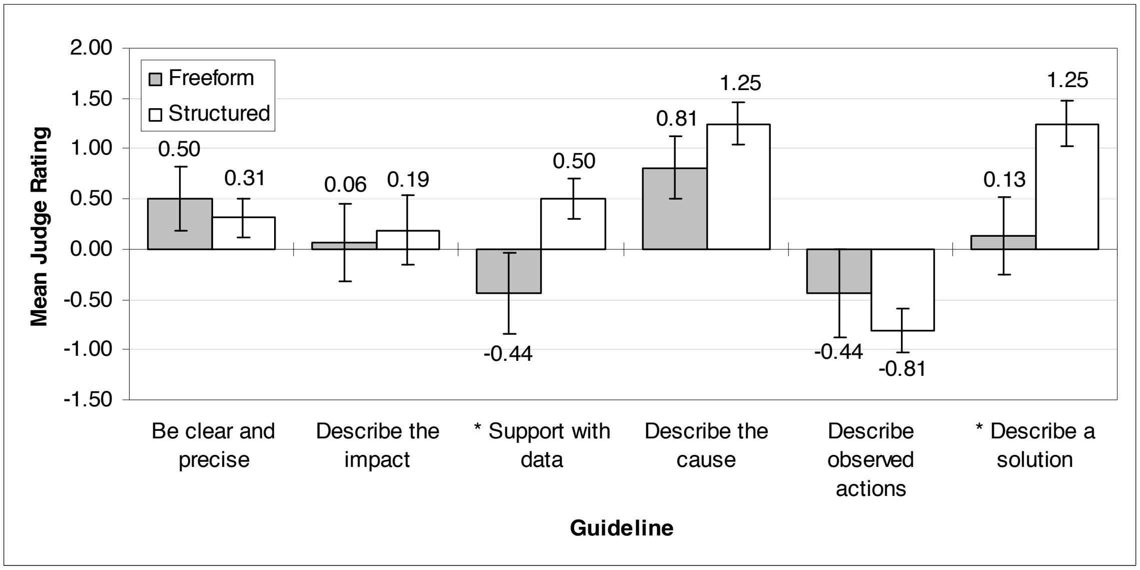


Figure 64: Study 3 - Quality as rated by judges, mean rating by treatment by guideline, bars represent standard error, * *p*<0.05

5.3.3.3 Hypothesis 3c.3

I hypothesized that tool support for merging UP instances and grouping UPs would increase the quality of novice evaluators' usability evaluation reports as rated by developers. I calculated three measures of quality: mean rating across all questions, mean rating per question, and mean summary rating (Section 5.3.2.6.1).

5.3.3.3.1 How the Developers Rated

Although differences in mean rating are of primary interest, an understanding of how the developers rated the lists of UP instances was useful in interpreting differences. I calculated measures of association, bias, and distribution for the developers.

Association

Association is the tendency of each developer to give higher/lower ratings to the same evaluator. Association was tested using Pearson's product-moment correlation both by treatment and by question; the results are in Table 43 and Table 44. Using an alpha level of 0.05, there was a significant correlation between all pairs of developers for the freeform treatment; there was not a significant correlation for the structured treatment. There was a significant correlation between all pairs of developers for all questions. Developers d1 and d2 gave ratings that were significantly correlated on 1 question, developers d1 and d3 on 3 questions, and developers d2 and d3 on 5 questions.

Table 43: Study 3 - Developer association by treatment, tested using Pearson's product-moment correlation

Treatment		d1 and d2	d1 and d3	d2 and d3
Freeform	*r*	0.46	0.61	0.59
	p	<0.01*	<0.01*	<0.01*
Structured	*r*	0.09	-0.09	-0.12
	p	0.54	0.56	0.54

n=48 per treatment; * *p*<0.05

Table 44: Study 3 - Developer association by question, tested using Pearson's product-moment correlation

Question		d1 and d2	d1 and d3	d2 and d3
All questions	*r*	0.27	0.37	0.55
	p	<0.01*	<0.01*	<0.01*
Be clear and precise	*r*	-0.05	-0.35	0.45
	p	0.85	0.18	0.08
Describe the impact	*r*	0.26	0.54	0.60
	p	0.33	0.03*	0.01*
Support with data	*r*	0.73	0.52	0.66
	p	<0.01*	0.04*	0.01*
Describe the cause	*r*	0.13	0.20	0.51
	p	0.65	0.45	<0.05*
Describe observed actions	*r*	0.44	0.47	0.73
	p	0.09	0.06	<0.01*
Describe a solution	*r*	0.20	0.63	0.62
	p	0.47	0.01*	0.01*

n=16 per question; * *p*<0.05

Bias

Bias is the tendency of each developer to give higher or lower ratings overall. Bias was tested using a 2x6x3 mixed-factor ANOVA, with treatment as a between-subject factor, question and developer as within-subject factors, and evaluator as a repeated measure. The results of the same ANOVA were used to compare mean rating scores; see Section 5.3.3.3.2 for the full ANOVA results.

Using an alpha level of 0.05, the developer main effect was not significant *F*(2, 252)=0.89. The developer x treatment interaction was not significant, *F*(2, 252)=0.67, nor was the developer x question interaction, *F*(10, 252)=0.94. Additionally, the developer x treatment x question interaction was not significant, *F*(10, 252)=0.83.

Distribution

Distribution is the tendency of each developer to use each point in the scale. Distribution was assessed using visual inspection of the developers' ratings by treatment (Figure 65); a visual inspection of developers' ratings by question is not included because it is difficult to visually analyze the ratings of three developers using a stacked bar graph like Figure 65. Each developer gave 48 ratings per treatment. Developer d3 used the endpoints (strongly disagree and strongly agree) more frequently (n=44) than developers d1 (n=11) and d2 (n=7). Developers d1 and d2 both used the innermost points (somewhat disagree and somewhat agree) equally (n=49), and developer d3 used them less (n=21). All three developers gave similar numbers of positive ratings (any of the agree ratings) (d1=73, d2=74, d3=78).

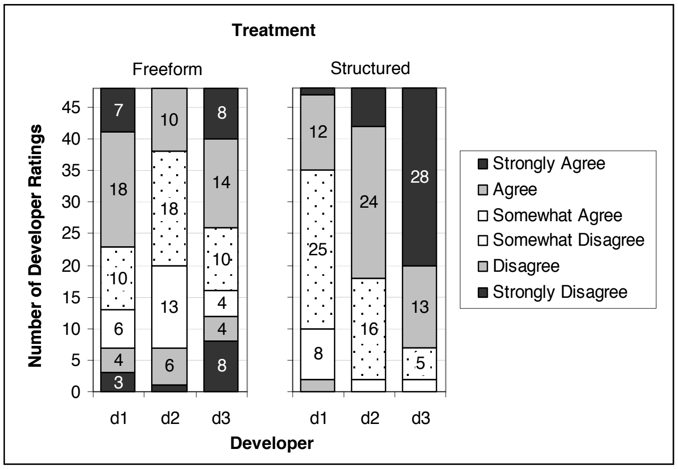


Figure 65: Study 3 - Developer distribution by treatment

5.3.3.3.2 Mean Ratings

The rating data are interval and can have only six values, so it is not possible to test for normality using the Shapiro-Wilk test. Instead, both a histogram and a normal quantile plot suggest that the rating data is jointly normally distributed.

Means are based on individual ratings given by each developer, rather than the sum of the two ratings. Developers rated on a 6-point scale, which has been adjusted to a rating from –2.5 to 2.5. Differences in mean rating across all questions by treatment were tested as part of a 2x6x2 mixed-factor ANOVA, with treatment as a between-subject factor, question and developer as within-subject factors, and evaluator as a repeated measure (Table 33). The effects specific to developer are discussed in the analysis of developer bias in Section 5.3.3.2.1.

Table 45: Study 3 - Quality as rated by developers, results of a 2x6x3 mixed factor ANOVA with treatment as a between-subject factor, question and developer as within-subject factors, and evaluator as a repeated measure

Source	*F*	DF Num	DF Den	*p*
Developer	0.89	2	252	0.41
Treatment	4.49	1	252	0.03*
Question	0.05	5	252	0.99
Developer x Treatment	0.67	2	252	0.51
Developer x Question	0.94	10	252	0.50
Treatment x Question	1.86	5	252	0.10
Developer x Treatment x Question	0.83	10	252	0.60

N=288, * *p*<0.05

Treatment

The mean rating for the structured treatment (*M*=1.21, *SD*=0.97) was significantly greater than for the freeform treatment (*M*=0.39, *SD*=1.43) (Figure 66).

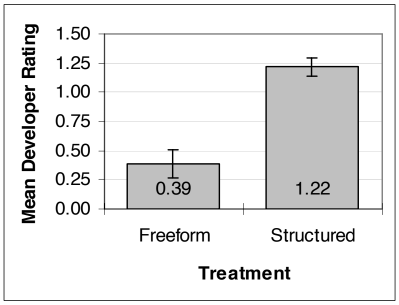


Figure 66: Study 3 - Quality as rated by developers, mean rating by treatment, bars represent standard error

Treatment x Question

Even though the treatment x question interaction effect was not significant, it was explored using slices for the purposes of the mean rating per question measure (Table 46). The developers gave significantly higher ratings for *be clear and precise*, *support with data*, *describe the cause*, and *describe a solution* for the structured treatment. Figure 67 shows the mean ratings per guideline.

Table 46: Study 3 - Quality as rated by developers, simple effects due to question for each treatment explored using slices

Question	*F*	*p*
Be clear and precise	4.49	0.04*
Describe the impact	1.55	0.21
Support with data	5.03	0.03*
Describe the cause	9.71	<0.01*
Describe observed actions	1.55	0.21
Describe a solution	23.63	<0.01*

DF Num=1, DF Den=252, * *p*<0.05

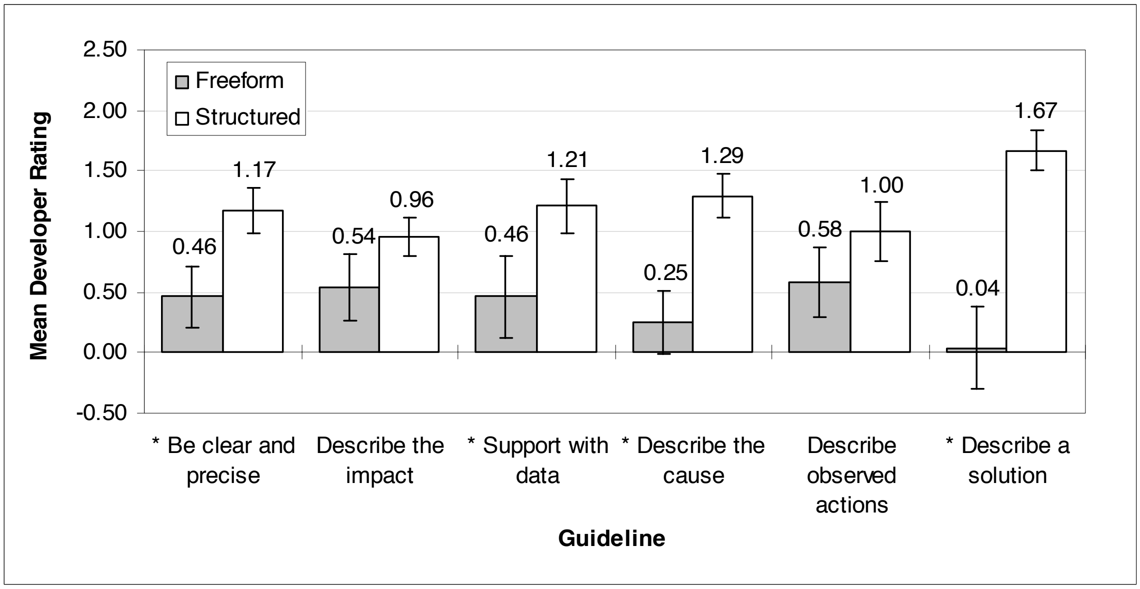


Figure 67: Study 3 - Quality as rated by developers, mean rating by treatment by question, bars represent standard error, * *p*<0.05

5.3.3.3.3 Mean Summary Rating

Figure 68 illustrates mean summary rating by treatment. The rating data are interval and can have only six values, so it is not possible to test for normality using the Shapiro-Wilk test. Instead, both a histogram and a normal quantile plot suggest that the rating data is not normally distributed and has a severe negative skew. As a result, a Wilcoxon rank-sum test, a non-parametric test, was performed instead of a *t*-test, a parametric test. Using a normal approximation procedure, the test indicated that there was a significant difference in the medians between treatments, W=772, *p*<0.01; the median of the structured treatment was greater than the median of the freeform treatment.

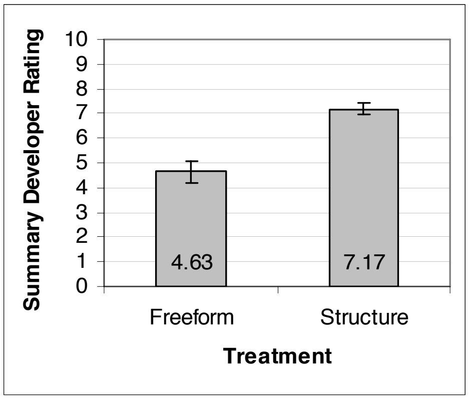


Figure 68: Study 3 - Mean summary rating by treatment, bars represent standard error

5.3.3.3.4 Qualitative Developer Feedback

As discussed in Section 5.3.2.6.2, I interviewed the developers using a semi-structured interview after they had finished assigning ratings. The following sections are a summary of the developers' feedback by topic of conversation.

Overall Observations

The developers agreed that there was a good deal of variance in the quality of the usability evaluation reports. One developer remarked that "some were almost professional grade, while others were almost unreadable". Overall, however, the developers preferred the structured reports to the freeform reports. They felt that the structured reports made it easier to get an overview of all the problems and then look at specific problems in detail.

Importance of Specific Solutions

All three developers talked at length about the importance of specific solutions. They agreed that usability evaluation reports that did not contain specific solutions to usability problems were of less value to them than those that did. Additionally, they all made comments to indicate that they gave reports without specific solutions lower overall ratings even if the reports were of high quality in all other aspects.

All the developers also mentioned that they felt some frustration when reading reports that contained generic solutions. For example, one developer made the following comment:

> "I wrote the software. Obviously I tried to do it right the first time. If the usability engineer doesn't offer me a specific solution to a problem, I may not be able to fix it. It wouldn't have shown up as a problem if I knew how to fix it."

The developer did not know how to fix the interaction design and wanted advice or suggestions from the evaluators. Another developer mentioned that the inclusion of specific solutions made reports seem helpful as opposed to critical attacks on his work.

Grouping

The evaluators in the study generally grouped usability problems according to the interaction design or by importance. Two of the developers commented that the grouping of problems according to the interaction design was useful for discussing the problems with management or other non-technical stakeholders, but it was not too useful in helping them fix the problems. For their own purposes, they preferred having the problems grouped by importance. One developer mentioned that the best grouping would have been based on the organization of the software modules that make up Scholar; the evaluators in the study, however, were not familiar with the organization of Scholar from a software engineering perspective and would not have been able to provide this grouping.

Trust

One of the developers commented on the role of trust when asked whether the videos of the sessions with the representative users were helpful. The developer explained that he did not need to see the video if he had established a working relationship with the usability practitioner who had reported the problem. The video was only of importance if he felt that the usability engineer had been identifying trivial problems or requesting too many conflicting changes to the system.

Capra's Guidelines

The developers were asked to describe the importance of each of Capra's guidelines in terms of the overall quality of a usability evaluation report. Two of the developers mentioned that they had trouble distinguishing between the following two guidelines: *describe the impact*, and *describe observed actions*. Additionally, two of the developers commented that *describe the cause* was a good opportunity for usability practitioners to help educate developers. One of the developers said that developers in general were "constantly in a learning mode" because they have to continually educate themselves on new technologies.

5.3.4 Discussion

Table 47 contains a summary of hypothesis testing results for study 3.

Table 47: Study 3 - Summary of Hypothesis Testing Results

Hypothesis	Result
H3c.1 – Tool support for merging UP instances and grouping UPs will not affect the time that it takes novice evaluators to perform evaluations.	**Supported** – There was not a significant difference in the time that it took evaluators to perform the evaluation or the number of evaluators who finished between treatments.
H3c.2 – Tool support for merging UP instances and grouping UPs will increase the quality of novice evaluators' usability evaluation reports as rated by usability practitioners (judges in this study).	Measures of quality are based on Capra's guidelines [2006]. Higher mean ratings map to more agreement with the guideline(s) thereby indicating higher quality. **Supported** – The usability evaluation reports produced by evaluators in the structured treatment were rated by the judges to be of higher quality than those produced by evaluators in the freeform treatment.
H3c.3 – Tool support for merging UP instances and grouping UPs will increase the quality of novice evaluators' usability evaluation reports as rated by developers.	Measures of quality are based on Capra's guidelines [2006]. Higher mean ratings map to more agreement with the guideline(s) thereby indicating higher quality. **Supported** – The usability evaluation reports produced by evaluators in the structured treatment were rated by the developers to be of higher quality than those produced by evaluators in the freeform treatment.

5.3.4.1 Time

There was not a significant difference in the time that it took evaluators to perform the evaluation or the number of evaluators who finished between treatments. I expected this result because relating and communicating usability data takes an amount of time that is proportional to the amount of usability data. The time associated with any process that is used to facilitate the relating and communicating of the usability data is minor in comparison. In the case of this study, the evaluators employed either a freeform or a structured process.

5.3.4.2 Quality as Rated by Judges and Developers

For this study, it was not possible to directly compare the judge and developer ratings on the evaluators' usability evaluation reports. Although both groups of individuals rated on Capra's guidelines, the guidelines were worded differently for each group. The guidelines for the judges, who had UE experience, were more technical and complete. The judges were asked to rate each usability evaluation report based on how well it achieved or met the guideline. As an example, the following is the *describe observed actions* guideline for the judges:

Describe observed user actions

- Include contextual information about the user and the task.
- Include specific examples, such as the user's navigation flow through the system, user's subjective reactions, screen shots and task success/failure.
- Mention whether the problem was user-reported or experimenter observed.

The developers did not have UE experience and were given more general descriptions of the guideline. The following is the *describe observed actions* guideline as it was worded for the developers: "This usability evaluation report describes what the users were doing when they encountered usability problems".

Schaeffer and Presser [2003] state:

> "There is an intricate relationship among the survey question as it appears in the questionnaire, the rules the interviewer is trained to follow, the cognitive processing of the participants, the interaction between the interviewer and respondent, and the quality of the resulting data" (p. 66).

Accordingly, the judges' ratings cannot be directly compared to the developers' ratings. It is acceptable, however, to compare the overall results between the judges and developers.

The usability evaluation reports produced by evaluators in the structured treatment, the treatment with tool support for merging UP instances and grouping UPs, were rated by both the judges and the developers to be of higher quality than those produced by evaluators in the freeform treatment. The results of this study build on those of study 1, which provided evidence that evaluators can reliably identify UP instances and describe them well. This study suggests that novice evaluators can work with usability data at the UP instance level and then relate and communicate the information through a structured process. The UP instances and the structured process are scaffolding to help the novice evaluators work with usability data and produce usability evaluation reports that are useful to other individuals involved in the UE process.

5.3.4.3 Feedback from Interviews with the Developers

The developers' ratings were generally consistent with the feedback that they provided in the interviews. For example, the developers talked at length about the importance of specific solutions. The evaluators in the structured treatment used a UP instance report format that included a solution field, and the developers' ratings for *describe a solution* for the structured treatment were significantly higher than those for the freeform treatment. Additionally, the developers mentioned that it was important to describe the cause of a UP because it helped them learn about types of UPs. The developers' ratings for *describe the cause* were significantly higher for the structured treatment; one explanation is that the specific process for merging UP instances into UPs helped evaluators find commonalities in terms of cause among UP instances.

The developers were also consistent among themselves in terms of the feedback that they provided during the interviews. For example, one developer discussed the role of trust in the relationship between usability practitioners and developers. The other two developers discussed the difficulties that they had distinguishing between the following two guidelines: *describe the impact* and *describe observed actions*. The developers' comments on trust and the two guidelines are related because the guidelines deal with details associated with specific instances of UPs that are not needed if the developers trust that the usability practitioners with whom they are working. More specifically, they trust the usability practitioners to provide them with relevant information on important UPs and to not waste their time on trivial UPs.

5.3.4.4 Limitation of the Study

This study was subject to the same limitation as study 1. Please see Section 5.1.4.4 for details.

6 Conclusions

This research had three primary goals:

1. Investigate difficulties experienced by usability practitioners and how these difficulties are addressed (or not) by state-of-the-art UE tools
2. Develop a set of desirable features for UE tools targeted at difficulties that are either unaddressed or poorly addressed by existing state-of-the-art tools
3. Evaluate these desirable features with respect to how they affect the effectiveness of novice evaluators

For research goals 1 and 2, I analyzed features provided by state-of-the-art tools with respect to documented difficulties and developed a set of desirable tool features to address these difficulties for novice usability practitioners:

- **UP instance records to address the difficulty of identifying and recording critical usability data:** Using paper and existing UE tools, usability practitioners write notes and raw comments during a lab-based usability evaluation and manually review and relate them to identify instances of UPs. My proposed feature allowed evaluators to work with usability data at the instance level and removed the need for a second pass through the data to consolidate raw comments.
- **UP diagnosis to address the difficulty of understanding and relating usability data:** The need for problem diagnosis is not new with UE; it is central to any domain that involves finding and fixing problems, including automobile repair and the medical field. My proposed feature provided a diagnosis framework of usability concepts to give usability practitioners a common way to understand and relate usability data and a common vocabulary for discussing it.
- **A structured process for combining and associating UP data to address the difficulty of communicating usability information:** Much research has been devoted to developing usability evaluation methods that are used in evaluations of software products. More recently, however, research has shifted away from methods and comparisons of methods to issues of how to use the data generated by these methods. It is no longer enough to simply identify UPs; they must be associated in a meaningful way. My proposed feature supported usability practitioners in merging instances of UPs and grouping UPs to create usability evaluation reports that facilitated understanding of key usability issues by other individuals, such as developers, involved in the UE process.

I developed a UE tool, the Data Collection, Analysis, and Reporting Tool (DCART), which contained these desirable tool features, and used it as a platform for studies for research goal 3 of how these desirable features address the documented difficulties. I discuss the results by desirable feature:

- **Study 1:** Novice usability practitioners who used a tool with support for UP instance records more reliably identified UP instances than those who used tools without support. Additionally, the usability practitioners with support created usability reports of higher quality as rated by judges. The results suggest that UP instances serve as scaffolding to help novice usability practitioners work with raw usability data.
- **Study 2:** Novice usability practitioners who used a tool with support for UP diagnosis were not more reliable in the UP instances that they identified nor did they produce reports of higher quality as rated by judges than those who used tools without support. The results, however, suggest that a diagnosis framework, once internalized, can affect UP instance discovery rates. Additionally, the results indicate that evaluators who perform diagnosis focus more on the cause or type of a UP and less on the details unique to a given instance of a UP.
- **Study 3:** Novice practitioners who used a tool with a structured process for combining and associating UP data produced reports of higher quality as rated by judges and developers than those who used tools without support. The results build upon those of the study of support for UP instances and provide evidence that novice evaluators can work with usability data at the UP instance level and then relate and communicate the information through a structured process

The results of the studies suggest that novice usability practitioners can benefit from appropriate tool support. Specifically, such tool support could help them more consistently produce higher quality usability reports.

6.1 UP instances

Current approaches rely on the expertise of problem analysts to extract UPs from the raw data in the UP analysis stage. The extraction of UPs, however, is not straightforward, particularly for novices. Raw usability data is typically very specific and detailed while UPs are necessarily general. I introduced the concept of UP instances to serve as scaffolding to help novice usability practitioners construct UPs from comments. UP instances have three important qualities.

The first quality is the usefulness of UP instances as scaffolding for novice usability practitioners. The results of studies 1 and 3 indicate that working at the UP instance level instead of the raw usability data level helps novice usability practitioners more consistently interpret the relationship between critical incidents and UPs and synthesize usability data.

The second quality is that UP instances can be relatively easily integrated into state-of-the-art UE tools. A UP instance record documents a UP as experienced by a user at a specific point in time during an evaluation. As a result, UP instance records have time stamps and can be integrated into the logging features of existing UE tools. Additionally, the structured process for combining and associating UP instances is hierarchical in nature and lends itself well to existing data structures and interface widgets (such as tree views).

The third quality is that UP instances offer some advantages for usability studies. A key component of many of the studies documented in the literature, particularly UEM evaluations, is matching lists of UPs produced by evaluators with a master list of UPs. Few of these studies, however, actually describe how they performed the matching [Lavery *et al.*, 1997]. Matching can be difficult at the UP level because UPs can be of any number of levels of granularity. For example, should a UP description that describes a problem with the wording of a specific label be matched with a more general UP description that describes a problem with the wording of all similar labels in the application? UP instances can be more directly matched than UPs because they are all at the same level of granularity; each UP instance only describes one instance of a user experiencing a UP.

To provide additional support for the claim that UP instances can be more directly matched than UPs, I compared the matching of UPs done in a dissertation study conducted by Capra [Capra, 2006, 2007] to the matching of UP instances by instance coders in my studies. In Capra's studies, evaluators created 532 UP descriptions, and 3 coders matched to a master list of 38 UP descriptions. In Capra's study, two UP descriptions matched if fixing the UP described in one UP description would fix the UP described in the other UP description and vice versa. At least 2 coders agreed on 239 (45%) of the evaluators' UP descriptions. At least 1 coder marked 27 of the evaluators' UP descriptions as vague. In my studies, evaluators created 500 UP instance records, and 2 instance coders matched to a master list of 38 UP instance records. The instance coders agreed on 350 (70%) of the evaluators' UP instance records. For 119 of the UP instance records on which they disagreed, only one instance coder rated the UP instance record as being vague. For the remaining 31 UP instance records, the instance coders disagreed as to which UP instance record in the master list a given evaluator's UP instance record matched.

In Capra's study, the coders were encouraged to match UP descriptions and to only rate a UP description as vague in specific circumstances. In my study, I encouraged the instance coders to mark UP instance records as vague if they would not have understood them without having watched the videos. As a result, I had a much higher number of vague ratings. Had I discouraged the instance coders from rating UP instances as vague except in specific circumstances, the agreement rate would have probably been higher. Regardless, the agreement rates were much higher (70%) than in Capra's study (45%), which suggests that UP instances can be more directly matched than UPs.

My research represents a first step in understanding and working with usability data at the UP instance level. Further research is necessary to understand if working with usability data at the UP instance level benefits experienced usability practitioners as well as novice usability practitioners. Additionally, further research is needed to understand how to develop a UP instance "lens" in usability practitioners. What specific skills, competencies, and abilities are necessary to develop in practitioners for them to identify and work with UP instances?

6.2 Usability Engineering Tools

The ultimate purpose of this work was to provide a set of features for working with usability data that could be integrated into UE tools to help novice practitioners perform usability evaluations and create useful reports. I chose UE tools as the focus of my dissertation work because tools enable the translation of theory into practice. A good example is the Unified Modeling Language (UML) [Object Management Group]. A large number of tools are available that allow software engineers to not only design software applications using UML, but also generate the initial code in a number of languages from the design. I would argue that without tool support UML would not have been as widely adopted and used. Existing UE tools represent a good start, particularly considering that the UE tool market is still somewhat of a niche market, but they need to be improved. These tools are primitive in the sense that they facilitate working with low-level data and provide little higher-level analysis and reporting support.

An essay by Lund [2006] on "post-modern usability" provides an additional motivation for a focus on UE tool support. Lund argues that we need to acknowledge the complexity of real world UE efforts and develop a solid theoretical basis for UE to help manage the complexity. Particularly in an engineering discipline, developing theory involves validation against real world data. Tools provide an ideal test bed for validating theory in UE because they enable the use of large usability datasets and provide a way to catalog and exchange usability data.

7 References

Andre, T. S., Hartson, H. R., Belz, S. M., & McCreary, F. A. (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., Hartson, H. R., & Williges, R. C. (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.

ANSI. (2001). *Common Industry Format for usability test reports.* American National Standard for Information Technology.

Arnold, K. Napkin look and feel (emotional responses to match reality). Retrieved December 21, 2005, from http://napkinlaf.sourceforge.net/

Badre, A. N., Hudson, S. E., & Santos, P. J. (1994). Synchronizing video and event logs for usability studies. Paper presented at the *Workshop on Advanced Visual Interfaces*.

Bailey, G. (1993). Iterative methodology and designer training in human-computer interface design. Paper presented at the *Conference on Human Factors in Computing Systems*.

Bias, R., & Mayhew, D. (1994). *Cost-Justifying Usability*. Boston, Massachusetts, USA: Harcourt Brace & Company.

Biobserve. Spectator. Retrieved December 29, 2005, from http://www.usability.biobserve.com/

Bit Debris Solutions. Usability Activity Log. Retrieved December 21, 2005, from http://www.bitdebris.com/products/ulablog/

Borges, J., Morales, I., & Rodriguez, N. (1996). Guidelines for designing usable World Wide Web pages. Paper presented at the *Conference on Human Factors in Computing Systems*.

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

Capra, M. (2006). *Usability Problem Description and the Evaluator Effect in Usability Testing.* Unpublished dissertation. Blacksburg, VA: Virginia Tech.

Capra, M. (2007). Personal communication.

Capra, M. G. (2001). *An Exploration of End-User Critical Incident Classification.* Unpublished master's thesis. Blacksburg, VA: Virginia Tech.

Castillo, J. C., Hartson, H. R., & Hix, D. (1998). Remote usability evaluation: Can users report their own critical incidents? Paper presented at the *Conference on Human Factors in Computing Systems*.

Classic System Solutions Inc. Agility. Retrieved December 21, 2005, from http://www.classicsys.com/classic_site/html/team_design.html

Cockton, G., & Lavery, D. (1999). A framework for usability problem extraction*.* Paper presented at the *International Conference on Human-Computer Interaction*.

Cockton, G., Woolrych, A., Hall, L., & Hindmarch, M. (2003). Changing analysts' tunes: The surprising impact of a new instrument for usability inspection method assessment. In *People & Computers XVII* (pp. 145-161).

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Education and Psychological Measurements,* 20, 37-46.

Creswell, J. W. (2003). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Thousand Oaks, CA: Sage Publications.

Dumas, J. S., Molich, R., & Jeffries, R. (2004). Describing usability problems: Are we sending the right message? *interactions,* 11(4), 24-29.

Etgen, M., & Cantor, J. (1999). What does getting WET (web event-logging tool) mean for web usability*.* Paper presented at the *Conference on Human Factors and the Web*.

Faraday, P. (2000). Visually critiquing web pages*.* Paper presented at the *Conference on Human Factors and the Web*.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin,* 76, 378-382.

Godfrey-Smith, P. (2003). *Theory and Reality: An Introduction to the Philosophy of Science*. Chicago: University of Chicago Press.

Gray, W. D., & Salzman, M. C. (1998). Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction,* 13(3), 203-261.

Griffin, T., Schwartz, S., & Sofronoff, K. (1998). Implicit processes in medical diagnosis. In K. Kirsner, C. Speelman, M. Maybery, A. O'Brien-Malone, M.

Anderson & C. MacLeod (Eds.), *Implicit and Explicit Mental Processes* (pp. 329-342). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Hammontree, M. L., Hendrickson, J. J., & Hensley, B. W. (1992). Integrated data capture and analysis tools for research and testing on graphical user interfaces. Paper presented at the *Conference on Human Factors in Computing Systems*.

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

Hartson, H. R., Andre, T. S., & Williges, R. C. (2001). Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction,* 13(4), 373-410.

Hartson, H. R., Andre, T. S., Williges, R. C., & Rens, L. v. (1999). The User Action Framework: A theory-based foundation for inspection and classification of usability problems. Paper presented at the *International Conference on Human-Computer Interaction*.

Hertzum, M. (1999). User testing in industry: A case study of laboratory, workshop, and field tests. Paper presented at the *5th ERCIM workshop*, Sankt Augustin, Germany.

Hertzum, M., & Jacobsen, N. E. (2003). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction,* 15(1), 183-204.

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

Hix, D., & Hartson, H. R. (1993b). Formative evaluation: Ensuring usability in user interfaces. In L. Bass & P. Dewan (Eds.), *User Interface Software* (pp. 1-30). New York: John Wiley & Sons.

Hoegh, R. T., Nielsen, C., Overgaard, M., Pedersen, M., & Stage, J. (2006). The impact of usability reports and user test observations on developers' understanding of usability data: An exploratory study. *International Journal of Human-Computer Interaction,* 21(2), 173-196.

Hornbæk, K., & Frøkjær, E. (2005). Comparing usability problems and redesign proposals as input to practical systems development. Paper presented at the *Conference on Human Factors in Computing Systems*.

Hornbaek, K., & Stage, J. (2006). The interplay between usability evaluation and user interaction design. *International Journal of Human-Computer Interaction,* 21(2), 117-123.

Howarth, J. (2006). Identifying immediate intention during usability evaluation. Paper presented at the *ACM Southeast Conference*.

ISO. (1998). *9241 - Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability.* Geneva, Switzerland: International Standards Organization.

ISO. (1999). *13407 - Human-centered design processes for interactive systems.* Geneva, Switzerland: International Standards Organization.

Ivory, M. Y., & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys,* 33(4), 470-516.

Jackson, S., Startford, S., Krajcik, J., & Soloway, E. (1996). A learner-centered tool for students building models. *Communications of the ACM,* 39(4), 48-49.

Jacobsen, N. E., Hertzum, M., & John, B. E. (1998). The evaluator effect in usability tests. Paper presented at the *Conference on Human Factors in Computing Systems*.

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

John, B. E., & Packer, H. (1995). Learning and using the cognitive walkthrough method: A case study approach. Paper presented at the *Conference on Human Factors in Computing Systems*.

Kalat, J. W. (1996). The development of thought and knowledge: Piaget's contributions. In *Introduction to Psychology* (pp. 410-419). Pacific Grove, CA: Brooks/Cole Publishing Company.

Kaur, K., Maiden, N., & Sutcliffe, A. (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.* Unpublished dissertation. Blacksburg, VA: Virginia Tech.

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

Kennedy, S. (1989). Using video in the BNR usability lab. *SIGCHI Bulletin,* 21(2), 92-95.

Lavery, D., & Cockton, G. (1997). Representing predicted and actual usability problems. Paper presented at the *International Workshop on Representations in Interactive Software Development*.

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

Law, E. (2006). Evaluating the downstream utility of user tests and examining the developer effect: A case study. *International Journal of Human-Computer Interaction,* 21(2), 147-172.

Lim, K. H., Benbasat, I., & Todd, P. (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.

Lowgren, J., & Nordqvist, T. (1992). Knowledge-based evaluation as design support for graphical user interfaces. Paper presented at the *Conference on Human Factors in Computing Systems*.

Ludi, S. (2000). Macromedia Director as a prototyping and usability testing tool. *ACM Crossroads,* 6(5).

Lund, A. M. (1997). Another approach to justifying the cost of usability. *interactions,* 4(3), 48-56.

Lund, A. M. (2006). Post-Modern Usability. *Journal of Usability Studies,* 2(1), 1-6.

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

Macleod, M., Bowden, R., Bevan, N., & Curson, I. (1997). The MUSiC performance measurement method. *Behaviour and Information Technology,* 16(4-5), 279-293.

Macleod, M., & Rengger, R. (1993). The development of DRUM: A software tool for video-assisted usability evaluation. Paper presented at the *BCS HCI Conference*.

Mahajan, R., & Shneiderman, B. (1997). Visual and textual consistency checking tools for graphical user interfaces. *IEEE Transactions on Software Engineering,* 23(11), 722-735.

Mayhew, D. (1999). *The Usability Engineering Lifecycle*. San Francisco, CA: Morgan Kaufmann.

Mind Design Systems. Bailey's usability testing environment. Retrieved November 30, 2005, from https://www.mindd.com/Content.aspx?pid=2011&region=ute

Nayak, N. P., Mrazek, D., & Smith, D. R. (1995). Analyzing and communicating usability data: Now that you have the data what do you do? *ACM SIGCHI Bulletin,* 27(1), 22-30.

Nielsen, C., Overgaard, M., Pedersen, M., & Stage, J. (2005). Feedback from usability evaluation to user interface design: Are usability reports any good? Paper presented at the *INTERACT*.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. Paper presented at the *Conference on Human Factors in Computing Systems*.

Nielsen, J. (1994). Heuristic evaluation. In J. M. Nielsen, R. L. (Ed.), *Usability Inspection Methods* (pp. 25-62). New York: John Wiley and Sons.

Nielsen, J. (2005). Usability for the masses. *Journal of Usability Studies,* 1(1), 2-3.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. Paper presented at the *Conference on Human Factors in Computing Systems*.

Noldus. Observer. Retrieved December 21, 2005, from http://www.noldus.com/site/doc200401012

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction* (pp. 31-62). Hillsdale, NJ: Erlbaum.

Object Management Group. Unified Modeling Language. Retrieved March 9, 2007, from http://www.uml.org/

Olsen, D. R. (1992). *User Interface Management Systems: Models and Algorithms*. San Mateo, CA, USA: Morgan Kaufman Publishers, Inc.

Olsen, D. R., Dempsey, E. P., & Rogge, R. (1985). Input/output linkage in a user interface management system. Paper presented at the *Conference on Computer Graphics and Interactive Techniques*.

Olsen, D. R., Green, M., Lantz, K. A., Schulert, A., & Sibert, J. L. (1987). Whiter (or wither) UIMS? Paper presented at the *Conference on Human Factors in Computing Systems*.

Open Software Foundation. (1991). *OSF/Motif Style Guide.* Englewood Cliffs, NJ, USA.

Ovo Studios. Ovo Logger. Retrieved December 21, 2005, from http://www.ovostudios.com/ovologger.asp

Parush, A., Nadir, R., & Shtub, A. (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.

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

Quintana, C., Krajcik, J., & Soloway, E. (2002). A case study to distill structural scaffolding guidelines for scaffolded software environments. Paper presented at the *Conference on Human Factors in Computing Systems*.

Rizzo, A., Marchigiani, E., & Andreadis, A. (1997). The AVANTI project: prototyping and evaluation with a cognitive walkthrough based on Norman's model of action. Paper presented at the *Designing Interactive Systems Conference*.

Rosson, M. B., Carroll, J. M., & Bellamy, R. (1990). Smalltalk scaffolding: A case study of minimalist instruction. Paper presented at the *Conference on Human Factors in Computing Systems*.

Rowe, A. L., Lowry, T., Halgren, S. L., & Cooke, N. J. (1994). A comparison of usability evaluations conducted by different teams. Paper presented at the *Conference on Human Factors in Computing Systems*.

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

Schaeffer, N. C., & Presser, S. (2003). The science of asking questions. *Annual Review of Sociology,* 29, 65-88.

Scholtz, J., & Laskowski, S. (1998). Developing usability tools and techniques for designing and testing web sites. Paper presented at the *Conference on Human Factors and the Web*.

Sears, A. (1995). AIDE: A step toward metric-based interface development tools. Paper presented at the *ACM Symposium on User Interface and Software Technology*.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-computer Interaction* (3rd ed.). Reading, MA: Addison-Wesley.

Sink, S. (1985). *Productivity Management: Planning, Measurement and Evaluation, Control and Improvement*: John Wiley and Sons, Inc.

Sink, S., & Tuttle, T. (1989). *Planning and Measurement in Your Organization of the Future*: Industrial Engineering and Management Press.

Smith, S. L. (1986). Standards versus guidelines for designing user interface software. *Behaviour and Information Technology,* 5(1), 47-61.

Soloway, E., Guzdial, M., & Hay, K. (1994). Learner-centered design: The challenge for HCI in the 21st century. *interactions,* 1(2), 36-48.

Springett, M. (1998). Linking surface error characteristics to root problems in user-based evaluation studies. Paper presented at the *Working Conference on Advanced Visual Interfaces*.

Szczur, M. (1994). Usability testing - on a budget: A NASA usability test case study. *Behavior and Information Technology,* 13(1/2), 106-118.

TechSmith. Morae: A complete usability testing solution for web sites and software. Retrieved November 30, 2005, from http://www.techsmith.com/products/morae/default.asp

Theaker, C. J., Phillips, R., Frost, T. M. E., & Love, W. R. (1989). HIMS: A tool for HCI evaluations. In A. Sutcliffe & L. Macaulay (Eds.), *People and Computers V (HCI '89 Conference)* (pp. 427-439).

Theofanos, M. (2005). Towards the design of effective formative test reports. *Journal of Usability Studies,* 1(1), 27-45.

Theofanos, M., Quesenbery, W., Snyder, C., Dayton, D., & Lewis, J. (2005). *Reporting on formative testing: A UPA 2005 workshop report.* Bloomingdale, IL.

Uehling, D. L., & Wolf, K. (1995). User action graphing effort (UsAGE). Paper presented at the *Conference on Human Factors in Computing Systems*.

Usable Net. LIFT. Retrieved December 21, 2005, from http://www.usablenet.com/

Users First. Users First - Portable professional tools to observe users. Retrieved November 30, 2005, from http://www.usersfirst.com/index.jsp?page=products

Uzilla. Uzilla: Tools for a usable web. Retrieved November 30, 2005, from http://uzilla.net/

Vermeeren, A., Kesteren, I. v., & Bekker, M. (2003). Managing the evaluator effect in user testing. Paper presented at the *International Conference on Human-Computer Interaction*.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Pyschological Processes*. Cambridge, MA: Harvard University Press.

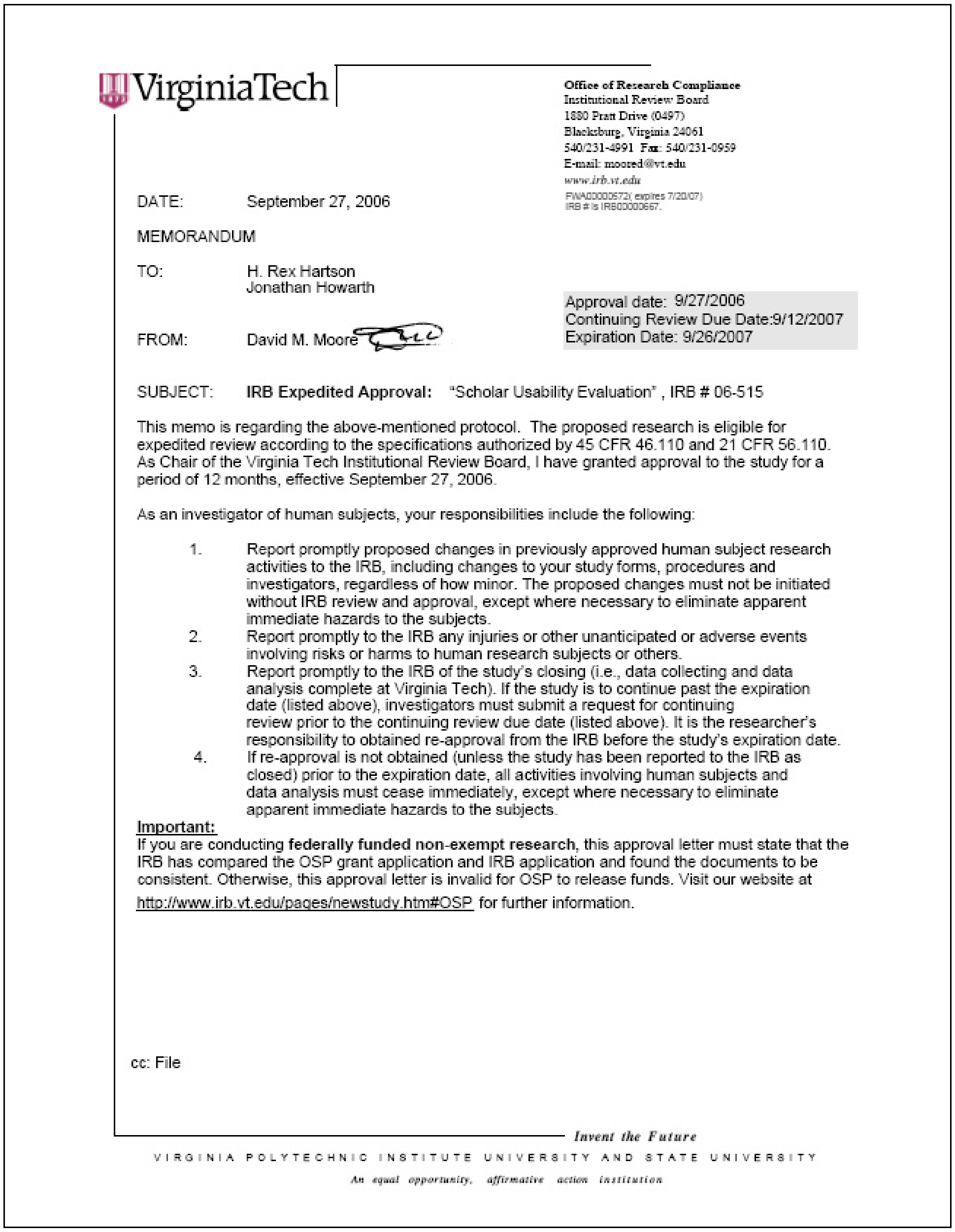
Weiler, P. (1993). Software for the usability lab: a sampling of current tools*.* Paper presented at the *Conference on Human Factors in Computing Systems*.

Wixon, D. (2003). Evaluating usability methods: why the current literature fails the practitioner. *interactions,* 10(4), 28-34.

Working Web. Usability tool: What does it do? Retrieved November 30, 2005, from http://workingweb.com.au/services/UsingUsabilityTool.php

Appendix A IRB Approval For Studies

The following is the approval for recording representative users' audio and screen video.



VirginiaTech

Office of Research Compliance
Institutional Review Board
1880 Pratt Drive (0497)
Blacksburg, Virginia 24061
540/231-4991 Fax: 540/231-0959
E-mail: moored@vt.edu
www.irb.vt.edu
FWA00000572(expires 7/20/07)
IRB # is IRB00000667.

DATE: September 27, 2006

MEMORANDUM

TO: H. Rex Hartson
Jonathan Howarth

Approval date: 9/27/2006
Continuing Review Due Date:9/12/2007
Expiration Date: 9/26/2007

FROM: David M. Moore

SUBJECT: **IRB Expedited Approval:** "Scholar Usability Evaluation" , IRB # 06-515

This memo is regarding the above-mentioned protocol. The proposed research is eligible for expedited review according to the specifications authorized by 45 CFR 46.110 and 21 CFR 56.110. As Chair of the Virginia Tech Institutional Review Board, I have granted approval to the study for a period of 12 months, effective September 27, 2006.

As an investigator of human subjects, your responsibilities include the following:

1. Report promptly proposed changes in previously approved human subject research activities to the IRB, including changes to your study forms, procedures and investigators, regardless of how minor. The proposed changes must not be initiated without IRB review and approval, except where necessary to eliminate apparent immediate hazards to the subjects.
2. Report promptly to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.
3. Report promptly to the IRB of the study's closing (i.e., data collecting and data analysis complete at Virginia Tech). If the study is to continue past the expiration date (listed above), investigators must submit a request for continuing review prior to the continuing review due date (listed above). It is the researcher's responsibility to obtained re-approval from the IRB before the study's expiration date.
4. If re-approval is not obtained (unless the study has been reported to the IRB as closed) prior to the expiration date, all activities involving human subjects and data analysis must cease immediately, except where necessary to eliminate apparent immediate hazards to the subjects.

Important:
If you are conducting **federally funded non-exempt research**, this approval letter must state that the IRB has compared the OSP grant application and IRB application and found the documents to be consistent. Otherwise, this approval letter is invalid for OSP to release funds. Visit our website at http://www.irb.vt.edu/pages/newstudy.htm#OSP for further information.

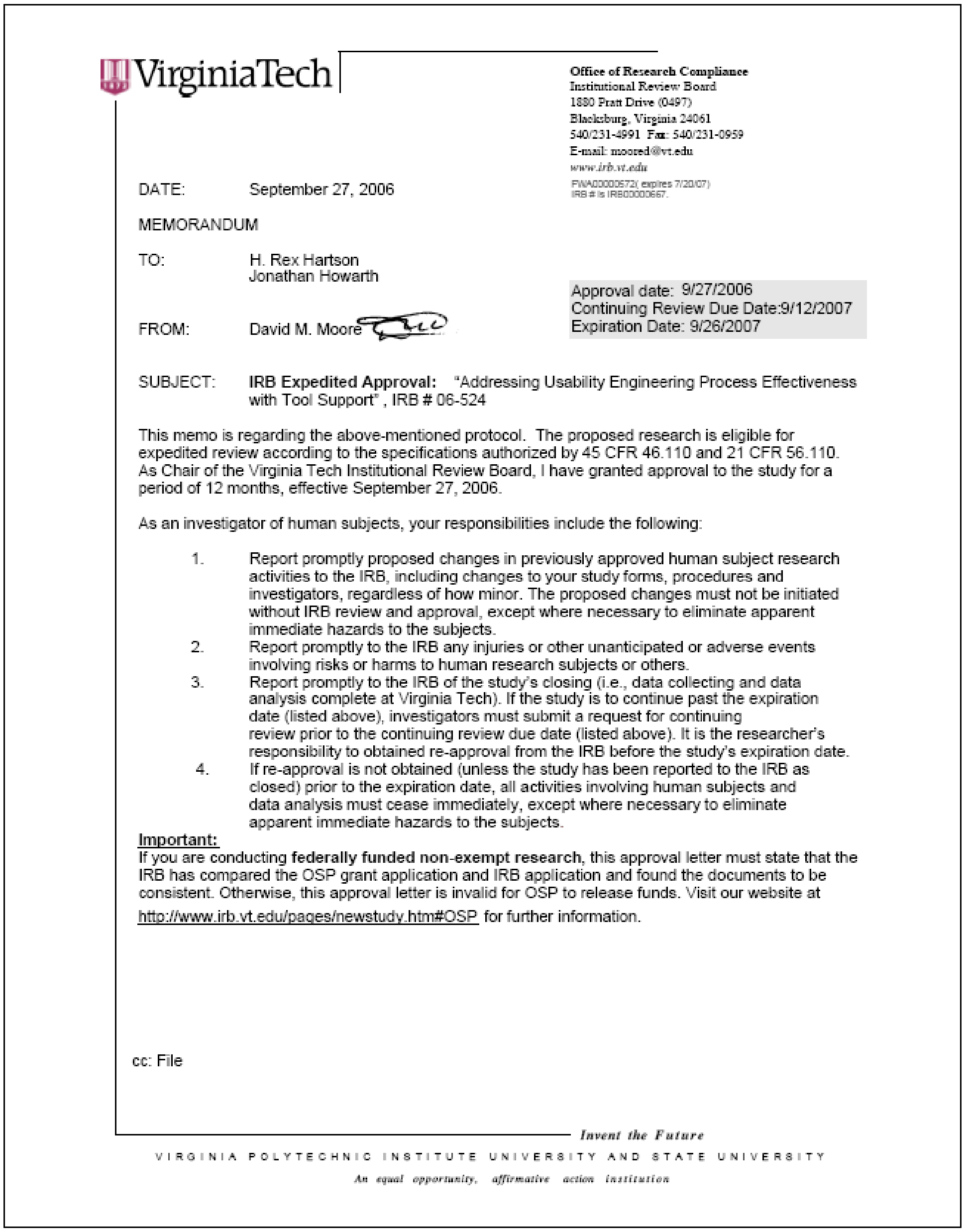
cc: File

Invent the Future

VIRGINIA POLYTECHNIC INSTITUTE UNIVERSITY AND STATE UNIVERSITY

An equal opportunity, affirmative action institution

The following is the approval for conducting studies 1, 2, and 3.



VirginiaTech

Office of Research Compliance
Institutional Review Board
1880 Pratt Drive (0497)
Blacksburg, Virginia 24061
540/231-4991 Fax: 540/231-0959
E-mail: moored@vt.edu
www.irb.vt.edu
FWA00000572(expires 7/20/07)
IRB # is IRB00000667.

DATE: September 27, 2006

MEMORANDUM

TO: H. Rex Hartson
Jonathan Howarth

Approval date: 9/27/2006
Continuing Review Due Date:9/12/2007
Expiration Date: 9/26/2007

FROM: David M. Moore

SUBJECT: **IRB Expedited Approval:** "Addressing Usability Engineering Process Effectiveness with Tool Support" , IRB # 06-524

This memo is regarding the above-mentioned protocol. The proposed research is eligible for expedited review according to the specifications authorized by 45 CFR 46.110 and 21 CFR 56.110. As Chair of the Virginia Tech Institutional Review Board, I have granted approval to the study for a period of 12 months, effective September 27, 2006.

As an investigator of human subjects, your responsibilities include the following:

1. Report promptly proposed changes in previously approved human subject research activities to the IRB, including changes to your study forms, procedures and investigators, regardless of how minor. The proposed changes must not be initiated without IRB review and approval, except where necessary to eliminate apparent immediate hazards to the subjects.
2. Report promptly to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.
3. Report promptly to the IRB of the study's closing (i.e., data collecting and data analysis complete at Virginia Tech). If the study is to continue past the expiration date (listed above), investigators must submit a request for continuing review prior to the continuing review due date (listed above). It is the researcher's responsibility to obtained re-approval from the IRB before the study's expiration date.
4. If re-approval is not obtained (unless the study has been reported to the IRB as closed) prior to the expiration date, all activities involving human subjects and data analysis must cease immediately, except where necessary to eliminate apparent immediate hazards to the subjects.

Important:
If you are conducting **federally funded non-exempt research**, this approval letter must state that the IRB has compared the OSP grant application and IRB application and found the documents to be consistent. Otherwise, this approval letter is invalid for OSP to release funds. Visit our website at http://www.irb.vt.edu/pages/newstudy.htm#OSP for further information.

cc: File

Invent the Future

VIRGINIA POLYTECHNIC INSTITUTE UNIVERSITY AND STATE UNIVERSITY

An equal opportunity, affirmative action institution

Appendix B Evaluator Materials

Appendix B.1 Evaluator Recruitment Email

Hi,

My name is Jonathan Howarth, and I am a graduate student in CS. I am conducting a study of usability engineering tools, and I am looking for participants. Details are provided below:

IRB Approval: This study has been approved by the IRB.

Eligible participants: CS and ISE VT graduate students who meet one of the following requirements are eligible to participate in this study:

- Have taken or are taking a usability engineering course
- Have taken or are taking an HCI course
- Have job experience related to usability engineering

Procedure: Participants will use a usability tool to perform a usability evaluation of a software application.

Date of studies: The study will take place between October 16 and November 17. Participants will be able to choose a date and time that is convenient for them from a list of available dates and times.

Location of study: The study will be conducted in 102 McBryde.

Compensation: All study participants will be paid a fixed fee of $25 in cash.

Time commitment: The study will take 2 to 3 hours.

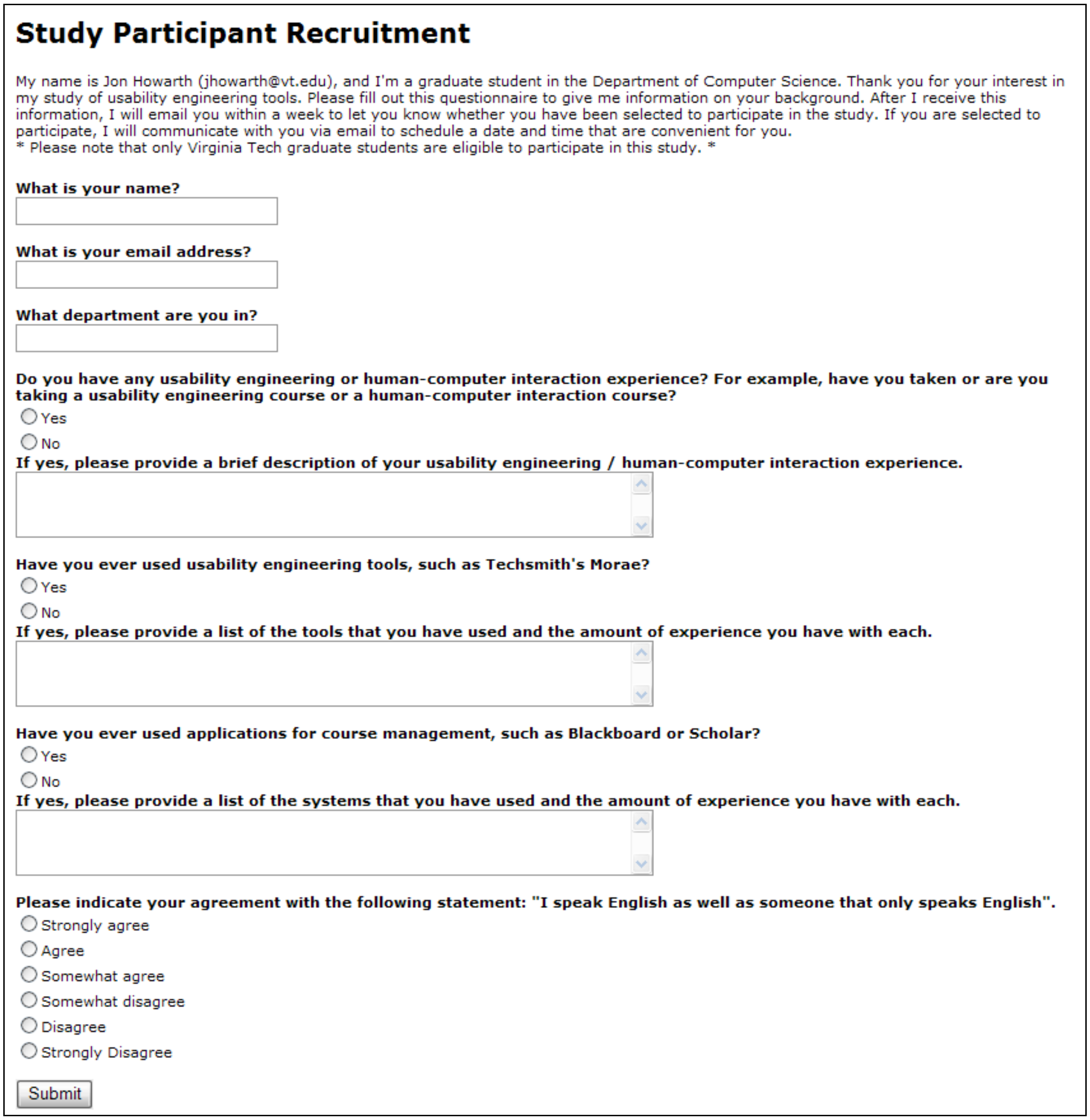
How to apply: Please fill out the survey at https://survey.vt.edu/survey/entry.jsp?id=1158939448396. This survey provides me with information on your background. I will contact you via email within a week of receiving your survey submission.

Thanks,

- Jon Howarth

8

Appendix B.2 Evaluator Background Survey



Study Participant Recruitment

My name is Jon Howarth (jhowarth@vt.edu), and I'm a graduate student in the Department of Computer Science. Thank you for your interest in my study of usability engineering tools. Please fill out this questionnaire to give me information on your background. After I receive this information, I will email you within a week to let you know whether you have been selected to participate in the study. If you are selected to participate, I will communicate with you via email to schedule a date and time that are convenient for you.
* Please note that only Virginia Tech graduate students are eligible to participate in this study. *

What is your name?

What is your email address?

What department are you in?

Do you have any usability engineering or human-computer interaction experience? For example, have you taken or are you taking a usability engineering course or a human-computer interaction course?

○ Yes

○ No

If yes, please provide a brief description of your usability engineering / human-computer interaction experience.

Have you ever used usability engineering tools, such as Techsmith's Morae?

○ Yes

○ No

If yes, please provide a list of the tools that you have used and the amount of experience you have with each.

Have you ever used applications for course management, such as Blackboard or Scholar?

○ Yes

○ No

If yes, please provide a list of the systems that you have used and the amount of experience you have with each.

Please indicate your agreement with the following statement: "I speak English as well as someone that only speaks English".

○ Strongly agree

○ Agree

○ Somewhat agree

○ Somewhat disagree

○ Disagree

○ Strongly Disagree

Appendix B.3 Evaluator Consent Form

Informed Consent for Participant of Investigative Project

Virginia Polytechnic Institute and State University
Department of Computer Science

Title of Project: Addressing Usability Engineering Process Effectiveness with Tool Support

Role of Participant: Evaluator

Investigators:
Dr. Rex Hartson, Professor, Computer Science, Virginia Tech
Jonathan Howarth, Graduate Student, Computer Science, Virginia Tech

I. The Purpose of this Research

You are invited to participate in a research study of usability engineering tools. Specifically, you will be working either with Morae, a commercial tool produced by TechSmith, or the Data Collection, Analysis, and Reporting Tool (DCART), a tool developed at Virginia Tech. There will be no more than 48 other participants in this study performing the same task as you.

II. Procedures

This study will be conducted in McBryde 102 on the Virginia Tech campus. Jonathan Howarth will begin by asking you to complete some sample exercises to familiarize yourself with the technology or technologies that you will be using. These technologies include one or more of the following: Morae, DCART, and/or the User Action Framework (UAF). Jonathan Howarth will then ask you to watch a video of people using a software application and to perform a series of tasks using either Morae or DCART. These tasks consist of identifying and recording usability problem instances encountered by people in the video. Your role in these tests is that of an evaluator of the previously mentioned tools. Jonathan Howarth is not evaluating you or your performance in any way; you are helping him to evaluate these tools. All information that you help him attain will remain anonymous. He may ask you questions while you are working with a tool. The session will last about two to three hours. The task is not very tiring, but you may take breaks if you wish.

III. Risks

There are no more than minimal risks associated with this study.

IV. Benefits of this Project

Your participation in this project will provide information that may be used to improve usability engineering tools. No promise or guarantee of benefits has been made to encourage you to participate. If you would like to receive a synopsis or summary of this research when it is completed, please notify Jonathan Howarth.

V. Extent of Anonymity and Confidentially

The results of this study will be kept strictly confidential. At no time will the results of the study be released to anyone other than individuals working on the project without your written consent. It is possible, however, that the Institutional Review Board (IRB) may view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research. The information you provide will have your name removed and only a participant number will identify you during analyses and any written reports of the research. The only individual that will have access to your name and participant number is Jonathan Howarth. He will destroy any identifying information within three years of completion of the study.

VI. Compensation

Jonathan Howarth will pay you a fixed fee of $25 as compensation. You will receive a payment in cash when you have completed the study.

VII. Freedom to Withdraw

You are free to withdraw from this study at any time for any reason without penalty. If you choose to withdraw from the study and do not complete it, you will still receive $2.50 for each quarter hour that you completed, up to a maximum of $25. You may also choose not to complete any part of the study, such as individual questions on a questionnaire, without penalty.

IX. Participant's Responsibilities

I voluntarily agree to participate in this study. I have the following responsibilities:

- To notify Jonathan Howarth at any time about a desire to discontinue participation.
- After completion of this study, I will not discuss my experiences with any other individual for a period of two months. This will ensure that everyone will begin the study with the same level of knowledge and expectations.

X. Participant's Permission

I have read and understand this informed consent form and the conditions of this study. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty.

____ _______________________

Signature Date

Should I have any questions about this research or its conduct, I may contact:

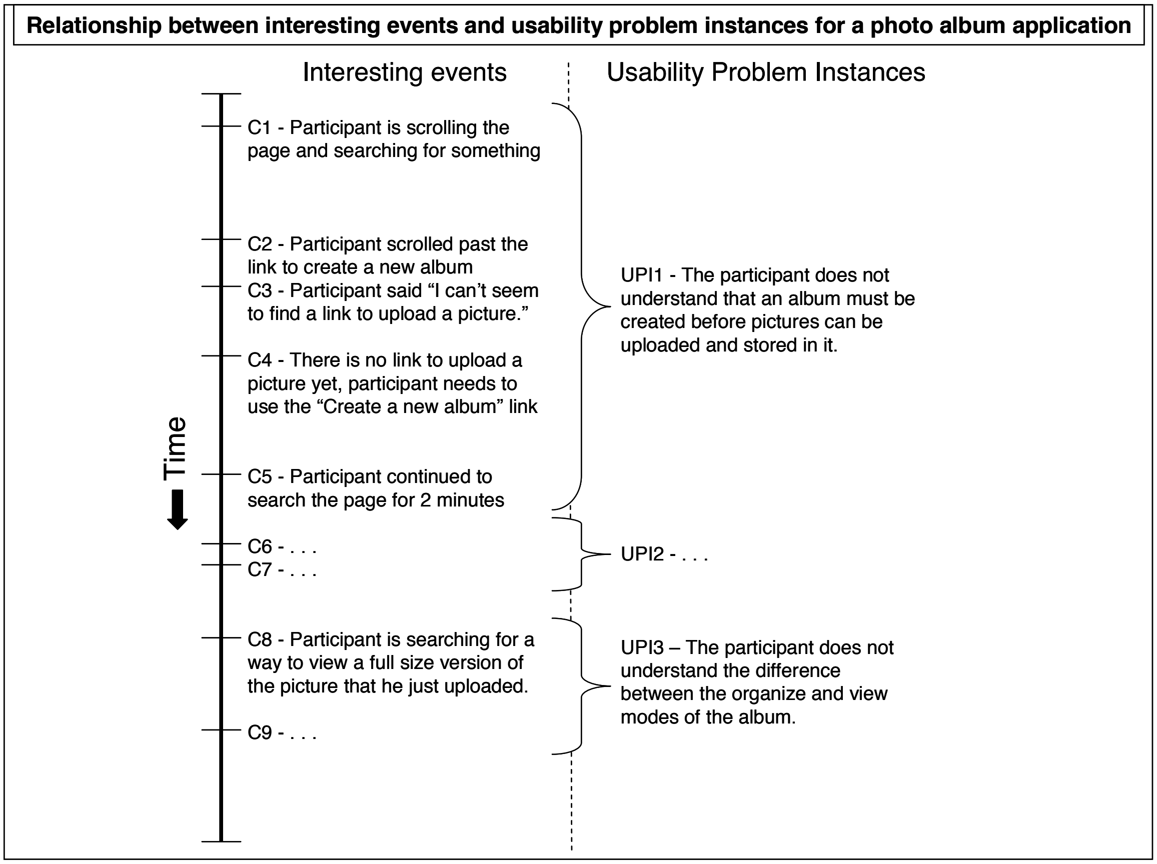
Dr. Rex Hartson, Investigator, hartson@vt.edu, (540)231-4857

Jonathan Howarth, Investigator, jhowarth@vt.edu, (540)961-5231

In addition, if you have detailed questions regarding your rights as a participant in University research, you may contact the following individual:

Dr. David Moore, Institutional Review Board Chair, moored@vt.edu, (540) 231-4991

Appendix B.4 Partial Data Relationship For Evaluators



Relationship between interesting events and usability problem instances for a photo album application

C1 - Participant is scrolling the page and searching for something

C2 - Participant scrolled past the link to create a new album

C3 - Participant said "I can't seem to find a link to upload a picture."

C4 - There is no link to upload a picture yet, participant needs to use the "Create a new album" link

C5 - Participant continued to search the page for 2 minutes

C6 - . . .

C7 - . . .

C8 - Participant is searching for a way to view a full size version of the picture that he just uploaded.

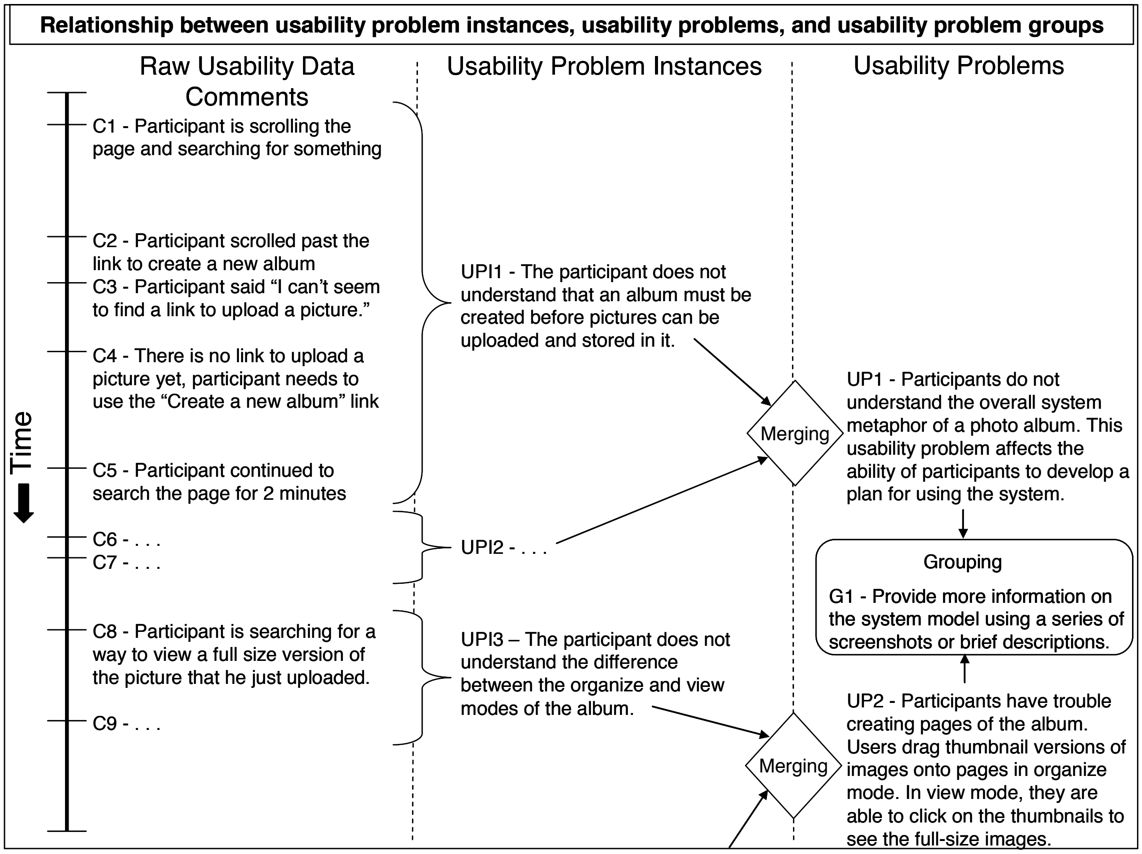
C9 - . . .

UPI1 - The participant does not understand that an album must be created before pictures can be uploaded and stored in it.

UPI2 - . . .

UPI3 – The participant does not understand the difference between the organize and view modes of the album.

Appendix B.5 Full Data Relationship for Evaluators



Relationship between usability problem instances, usability problems, and usability problem groups

C1 - Participant is scrolling the page and searching for something

C2 - Participant scrolled past the link to create a new album

C3 - Participant said "I can't seem to find a link to upload a picture."

C4 - There is no link to upload a picture yet, participant needs to use the "Create a new album" link

C5 - Participant continued to search the page for 2 minutes

C6 - . . .

C7 - . . .

C8 - Participant is searching for a way to view a full size version of the picture that he just uploaded.

C9 - . . .

UPI1 - The participant does not understand that an album must be created before pictures can be uploaded and stored in it.

UPI2 - . . .

UPI3 – The participant does not understand the difference between the organize and view modes of the album.

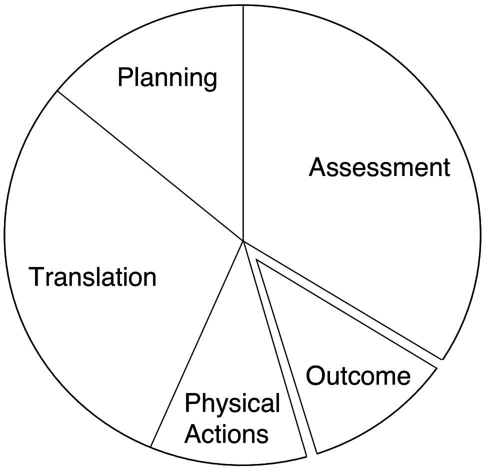
UP1 - Participants do not understand the overall system metaphor of a photo album. This usability problem affects the ability of participants to develop a plan for using the system.

G1 - Provide more information on the system model using a series of screenshots or brief descriptions.

UP2 - Participants have trouble creating pages of the album. Users drag thumbnail versions of images onto pages in organize mode. In view mode, they are able to click on the thumbnails to see the full-size images.

Appendix B.6 UAF Reference Sheet

Immediate intention provides information about what the participant was doing or attempting and why at the time of experiencing a usability problem instance. Immediate intention is expressed in terms of the type of user action involved in the context of the location within the Interaction Cycle of the User Action Framework, a conceptual framework of usability concepts.



Planning - how the design helps users do planning and understanding in general what the system can be used for.

Translation - how the design helps users translate their plans into actions to do specific things, such as what action to make on what interface object.

Physical Actions - how the design helps users do actual physical actions, such as keystrokes, mouse clicks, and dragging.

Outcomes and System Functionality – events or processing in the non-user-interface backend of the system, such as when there is missing functionality or software bugs there.

Assessment - how the design helps users understand feedback, or displays of results, that come back from the system, especially error messages.

Appendix B.7 Evaluator Study 1 Morae Instructions

Overview

During this study, we will ask you to do the following:

1. Watch the Morae tutorial video
2. Perform a familiarization exercise
3. Watch videos of users performing tasks with Scholar, a course management application, and document all usability problem instances that they encounter.

Part 1 – Tutorial Video

Please double click the “Morae Tutorial Video.wmv” icon on the desktop. Please watch the Morae tutorial video; it is approximately 8 minutes long. When you are finished, please close Windows Media Player.

Part 2 – Familiarization Exercise

2.1 Video of the Correct Way to Accomplish a Task

Double click the “Familiarization Video Correct.wmv” icon on the desktop and watch the video. This video will show you the correct way to accomplish the task of finding movies where two individuals are credited alongside one another in the Internet Movie Database (IMDB).

2.2 **Inserting Markers**

The study facilitator will now set up a familiarization session for you in Morae. Please read through the rest of this section before you begin.

Use the Remote Viewer to watch a participant perform the task of finding movies where two individuals are credited alongside one another in the IMDB. You have already seen a video of how to accomplish the task. Create markers for the following:

- The beginning and ending of the task
- Comments for two usability problem instances experienced by the participant

Notify the study facilitator when you have inserted the markers that you need to document two usability problem instances in your report. He will save your Morae recording and import it into a Morae Manager project.

Review your markers in Morae Manager and make any necessary changes, such as editing the text of the markers or adjusting the time of the markers.

Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms "edit" and "revise" are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

2.3 Generation of a Usability Report

When you are done reviewing your markers, create a usability report as demonstrated in the Morae tutorial video or by using a process of your own choosing. There are two requirements for your report:

- The report should contain only usability problem instance records; you may need to combine or separate comments in your markers to achieve this.
- Each usability problem instance record should have a timestamp associated with it.

Save the file to the desktop as "Participant Familiarization Report Morae.doc".

Open the report and compare it to the report on the desktop titled "Practitioner Familiarization Report Morae.1.doc". This report was generated by a practitioner. Please read and answer the following questions out loud to the study facilitator:

1. In comparing the usability problem instances in your report to those in the practitioner's report, which are similar?
2. Does your report contain any usability problem instances that the practitioner's report does not?

Part 3 – Usability Evaluation of Scholar

In this part, you will first watch three videos to become familiar with Scholar and the steps for adding and removing students from courses. You will then conduct a usability evaluation consisting of three tasks; in two of the tasks, participants are adding students, and in the other task, the participant is removing students.

3.1 Familiarization with Scholar

Please double click the following icons on the desktop in the following order and watch the videos:

1. "Scholar Introduction.wmv"
2. "Scholar Add Student Task Correct.wmv"
    a. The text for the task reads: A student emailed you to ask your permission to force add the course. Add him to the course. His pid is "psd_student_1".
3. "Scholar Remove Student Task Correct.wmv"
    a. The text for the task reads: On the first day of class, you realized that John Dewey has not taken the necessary prerequisites and is not eligible for the course. Remove him from the course.

3.2 Usability Evaluation

This section describes how you will perform the usability evaluation. Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms "edit" and "revise" are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Record as many usability problem instances as you can. Even if the one person experiences the same usability problem more than one time, create a usability problem instance record for each one. For example, if a participant were to click on an incorrect link or button a second time, create a second usability problem instance record even if it is almost identical to the first record.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

Use the Remote Viewer to watch the following participants perform the following tasks in the following order:

1. "s27 - Add a student"
2. "s67 - Add a student"
3. "s27 - Remove a student"

You have already seen videos of how to accomplish the tasks. Create markers for the following:

- The beginning and ending of the task
- Comments on usability problem instances experienced by the participant

When the task is finished, the study facilitator will save your Morae recording and import it into a Morae Manager project.

Review your markers in Morae Manager and make any necessary changes, such as editing the text of the markers or adjusting the time of the markers.

When you are done reviewing your markers, create a usability report as demonstrated in the Morae tutorial video or by using a process of your own choosing. There are two requirements for your report:

- The report should contain only usability problem instance records; you may need to combine or separate comments in your markers to achieve this.
- Each usability problem instance record should have a timestamp associated with it.

Save the file to the desktop as “Participant Scholar Report Morae.doc”.

You will have a maximum of 1.5 hours to perform the evaluation and generate a report.

Appendix B.8 Evaluator Study 1 DCART Instructions

Overview

During this study, we will ask you to do the following:

1. Watch the DCART tutorial video
2. Perform a familiarization exercise
3. Watch videos of users performing tasks with Scholar, a course management application, and document all usability problem instances that they encounter.

Part 1 – Tutorial Video

Please double click the “DCART Familiarization” icon on the desktop. Please watch the following video:

- “DCART Tutorial – Express” (16 minutes)

When you are finished, please close the tutorial window and close the welcome window.

Part 2 – Familiarization Exercise

2.1 General Familiarization with DCART

Please read and answer the following questions out loud to the study facilitator using the information in DCART:

1. In the “Familiarization Video” project, what is the participant’s name?
2. In the “Actual IMDB Path” task run, what benchmark task has the user performed?
3. What is the target value for the task in minutes?

2.2 Video of the Correct Way to Accomplish a Task

Go to the “Correct IMDB Path” task run and watch the video. This video will show you the correct way to accomplish the task of finding movies where two individuals are credited alongside one another in the Internet Movie Database (IMDB).

2.3 Identification of Usability Problem Instances

Read through this entire section before beginning. Go to the “Collect and Review” tab of the “Actual IMDB Path” task run and select the “Collect usability record data” link. Create usability records for two usability problem instances encountered by the participant.

When you have finished creating your two usability problem instances, close the "Usability Record Collection Form" and select the "Review usability records" link. Edit the text of your usability records and adjust their starting times as necessary.

When you are done reviewing your records, select the "Usability Records" link in the Collected Data Pools View. Work with your records and fill in the fields as necessary.

Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms "edit" and "revise" are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

2.4 Generation of a Usability Report

Select the "Generate a Usability Report" link in the Collected Data Pools View. Save the file to the desktop as "Participant Familiarization Report DCART.doc". Open the report and compare it to the report on the desktop titled "Practitioner Familiarization Report DCART.1.doc". This report was generated by a practitioner. Please read and answer the following questions out loud to the study facilitator:

1. In comparing the usability problem instances in your report to those in the practitioner's report, which are similar?
2. Does your report contain any usability problem instances that the practitioner's report does not?

Part 3 – Usability Evaluation of Scholar

Please close DCART and then double click the "DCART Scholar Evaluation" icon on the desktop. DCART is set up for a usability evaluation of Scholar, a course management system. First, you will watch some videos to become familiar with Scholar and the steps for adding and removing students from courses. You will then conduct a usability evaluation consisting of three task runs; in two of the task runs, participants are adding students, and in the other task run, the participant is removing a student.

3.1 Familiarization with Scholar

Open the following task runs in the following order and watch the videos associated with them using the “View video” link at the bottom of each task run to become familiar with Scholar:

1. “An introduction to Scholar”
2. “Correct - Add a student”
3. “Correct - Remove a Student”

3.2 Usability Evaluation

This section describes how you will perform the usability evaluation. Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Record as many usability problem instances as you can. Even if the one person experiences the same usability problem more than one time, create a usability problem instance record for each one. For example, if a participant were to click on an incorrect link or button a second time, create a second usability problem instance record even if it is almost identical to the first record.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

Collect usability record data using the “Collect usability record data” link on the Collect and Review tab for the following task runs in the following order:

1. “s27 - Add a student”
2. “s67 - Add a student”
3. “s27 - Remove a student”

Review the usability record data for the task runs using the “Review usability records” link on the Collect and Review tab. You can review the task runs in any order that you like.

When you are done reviewing your records, select the “Version 2.2.x” level in the Levels View and the “Usability Records” link in the Collected Data Pools View. Work with your records and fill in the fields as necessary.

Select the “Generate a Usability Report” link in the Collected Data Pools View. Please save your report on the desktop with the name “Participant Scholar Report DCART.doc”. Edit the document as necessary.

You will have a maximum of 1.5 hours to perform the evaluation and generate a report.

Appendix B.9 Evaluator Study 1 Morae Familiarization Sample

No joint search
Time: 0:1:47.88
The participant tried to use the search operator "and" in the search box at the top left corner of the screen, but search operators are not supported. Because it is a search, the participant expects some form of operators. Two possible options are to either support search operators or to provide an advanced search option that uses a form-based approach to support search operators.

Participant doesn't understand how the results relate to his search query
Time: 0:1:59.98
Because the participant entered a search term with operators, he expected a fairly short list of results. Instead he is presented with an extensive list of results that do not appear to relate to his query. One possible solution is to catch the fact that a user tried to use a search operator and provide feedback on the results page that search operators are not supported.

Not sure what the name search does
Time: 0:2:28.48
There is no explanation as to whether a name in the "More Searches" area is the name of a person, a character, a movie, etc. One option is to provide a more specific term. For example, if the name search searched the real names of actors, then just use the term "Actor".

Overwhelming number of results for a name search
Time: 0:2:22.50
The name search returned almost 1000 results in an uncategorized list. Such a result is overwhelming. Instead, the system could show only the most relevant subset of the results or provide a paging mechanism to allow the user to view only a subset of the results at a time.

Option for the credited alongside search is scrolled of the screen.
Time: 0:3:34.37
The option for the credited alongside search is at the bottom of Owen Wilson's IMDB page. Depending on its frequency of use, it may be appropriate to move it higher on the page. Regardless, it should still appear above the message boards, which typically mark the end of content provided by the site and the beginning of content provided by users.

Links at top of the joint search seem unrelated to the purpose of the search
Time: 0:4:11.28
The links at the top (example "[wilson: 10412]") do not have a readily understandable purpose and do not seem to relate to the results of the joint

search. Without having better knowledge of the purpose of the links, I would suggest removing them from the page.

Character selection checkboxes appear after the Look up joint ventures button
Time: 0:4:34.51
The checkboxes appear after the action button to which they are related. One suggestion is to put the Look up joint ventures button after the actor categories and checkboxes.

Clicked the look up joint ventures button without selecting actors
Time: 0:4:34.51
The participant clicked the Look up joint ventures button without selecting actors. Requiring that users select roles after searching on joint ventures adds an extra step that users are not expecting. One solution is to show all joint ventures and then provide a mechanism for filtering by role.

Match names with any occupation is scrolled of the screen
Time: 0:4:43.43
The option for finding joint ventures regardless of role (actor, director, etc) is below the categories of roles. One solution is to move the checkbox above the roles and actor checkboxes, so that users know that it exists before they take the time to check the checkboxes for several roles.

Can't distinguish between Luke Wilson (I) and (II)
Time: 0:5:01.77
There are two entries for Luke Wilson that are differentiated by roman numerals in parentheses. The (I) and (II) distinguish between the two actors with the same name, but they are not user centered. It might be more appropriate to distinguish between them by middle name, for example.

Appendix B.10 Evaluator Study 1 DCART Familiarization Sample

Actual IMDB path (Task Run)

Location:

Virginia Tech > Familiarization Video > Actual > Session > Actual IMDB path

User Class:

Casual User - This individual uses the IMDB to get reviews for movies and find out information about his favorite movie stars.

Benchmark Task:

Credited alongside - Find all moves where Owen Wilson is credited alongside his brother Luke Wilson.

Usability Specification(s):

- Time on task - How long does it take a participant to complete the task.

Instances

1. No joint search

Start time:

0:0:28.88

Description:

The participant tried to use the search operator "and", but search operators are not supported. Because it is a search, the participant expects some form of operators.

User interface object:

The search at the top left corner of the screen

Designer Knowledge:

The search does not support search operators.

Solution:

Two possible options are to either support search operators or to provide an advanced search option that uses a form-based approach to support search operators.

2. Participant doesn't understand how the results relate to his search query

Start time:

0:0:40.98

Description:

Because the participant entered a search term with operators, he expected a fairly short list of results. Instead he is presented with an extensive list of results that do not appear to relate to his query.

User interface object:

Search results list

Solution:

One possible solution is to catch the fact that a user tried to use a search operator and provide feedback on the results page that search operators are not supported.

3. Not sure what the name search does

Start time:

0:1:9.48

Description:

There is no explanation as to whether a name is the name of a person, a character, a movie, etc.

User interface object:

The searches listed under "More Searches" at the bottom of a search results page.

Designer Knowledge:

I am unable to determine specifically what the "Name" search searches.

Solution:

Provide a more specific term. For example, if the name search searched the real names of actors, then just use the term "Actor".

4. Overwhelming number of results for a name search

Start time:

0:1:23.50

Description:

The name search returned almost 1000 results in an uncategorized list. Such a result is overwhelming.

User interface object:

The search results list for the name search

Solution:

One solution is to show only the most relevant subset of the results. The second is to implement a paging mechanism to allow the user to show a only a subset of the results at a time.

5. Option for the credited alongside search is scrolled off the screen

Start time:

0:2:15.37

Description:

The option for the credited alongside search is at the bottom of Owen Wilson's IMDB page.

User interface object:

The credited alongside search box

Solution:

Depending on its frequency of use, it may be appropriate to move it higher on the page. Regardless, it should still appear above the message boards, which typically mark the end of content provided by the site and the beginning of content provided by users.

6. Links at top of the joint search seem unrelated to the purpose of the search

Start time:

0:2:52.28

Description:

The links at the top do not have a readily understandable purpose and do not seem to relate to the results of the joint search.

User interface object:

Links like the following [wilson: 10412]

Designer Knowledge:

I'm not sure what the links refer to or why they appear on this page.

Solution:

Without having better knowledge of the purpose of the links, I would suggest removing them from the page.

7. Character selection checkboxes appear after the Look up joint ventures button

Start time:

0:3:15.51

Description:

The checkboxes appear after the action button to which they are related.

User interface object:

Look up joint ventures button and actor checkboxes

Solution:

One suggestion is to put the Look up joint ventures button after the actor categories and checkboxes.

8. Clicked the look up joint ventures button without selecting actors

Start time:

0:3:15.51

Description:

The participant clicked the Look up joint ventures button without selecting actors.

User interface object:

Look up joint ventures button and actor checkboxes

Solution:

Requiring that users select roles after searching on joint ventures adds an extra step that users are not expecting. One solution is to show all joint ventures and then provide a mechanism for filtering by role.

9. Match names with any occupation is scrolled of the screen

Start time:

0:3:24.43

Description:

The option for finding joint ventures regardless of role (actor, director, etc) is below the categories of roles.

User interface object:

Match names with any occupation checkbox

Solution:

One solution is to move the checkbox above the roles and actor checkboxes, so that users know that it exists before they take the time to check the checkboxes for several roles.

10. Can't distinguish between Luke Wilson (I) and (II)

Start time:

0:3:42.77

Description:

There are two entries for Luke Wilson that are differentiated by roman numerals in parantheses.

User interface object:

Actor name link with roman numerals

Designer Knowledge:

There are two different actors with the name Luke Wilson, so the (I) and (II) are used to distinguish between them.

Solution:

The (I) and (II) distinguish between the two actors with the same name, but they are not user centered. It might be more appropriate to distinguish between them by middle name, for example.

Appendix B.11 Evaluator Study 2 Partial Diagnosis Instructions

Overview

During this study, we will ask you to do the following:

1. Watch the DCART and immediate intention tutorial videos
2. Perform a familiarization exercise
3. Watch videos of users performing tasks with Scholar, a course management application, and document all usability problem instances that they encounter.

Part 1 – Tutorial Videos

Please double click the “DCART Familiarization” icon on the desktop. Please watch the following videos:

- “DCART Tutorial” (16 minutes)
- “Immediate Intention Tutorial” (4 minutes)

When you are finished, please close the tutorial window and close the welcome window.

Part 2 – Familiarization Exercise

2.1 General Familiarization with DCART

Please read and answer the following questions out loud to the study facilitator using the information in DCART:

1. In the “Familiarization Video” project, what is the participant’s name?
2. In the “Actual IMDB Path” task run, what benchmark task has the user performed?
3. What is the target value for the task in minutes?

2.2 Video of the Correct Way to Accomplish a Task

Go to the “Correct IMDB Path” task run and watch the video. This video will show you the correct way to accomplish the task of finding movies where two individuals are credited alongside one another in the Internet Movie Database (IMDB).

2.3 Identification of Usability Problem Instances

Read through this entire section before beginning. Go to the “Collect and Review” tab of the “Actual IMDB Path” task run and select the “Collect usability record data" link. Create usability records for two usability problem instances encountered by the participant.

When you have finished creating your two usability problem instances, close the "Usability Record Collection Form" and select the "Review usability records" link. Edit the text of your usability records and adjust their starting times as necessary. Be sure to specify immediate intention for each record.

When you are done reviewing your records, select the "Usability Records" link in the Collected Data Pools View. Work with your records and fill in the fields as necessary.

Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms "edit" and "revise" are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

2.4 Generation of a Usability Report

Select the "Generate a Usability Report" link in the Collected Data Pools View. Save the file to the desktop as "Participant Familiarization Report DCART.doc". Open the report and compare it to the report on the desktop titled "Practitioner Familiarization Report DCART.2Partial.doc". This report was generated by a practitioner. Please read and answer the following questions out loud to the study facilitator:

1. In comparing the usability problem instances in your report to those in the practitioner's report, which are similar?
2. Does your report contain any usability problem instances that the practitioner's report does not?

Part 3 – Usability Evaluation of Scholar

Please close DCART and then double click the "DCART Scholar Evaluation" icon on the desktop. DCART is set up for a usability evaluation of Scholar, a course management system. First, you will watch some videos to become familiar with

Scholar and the steps for adding and removing students from courses. You will then conduct a usability evaluation consisting of three task runs; in two of the task runs, participants are adding students, and in the other task run, the participant is removing a student.

3.1 Familiarization with Scholar

Open the following task runs in the following order and watch the videos associated with them using the “View video” link at the bottom of each task run to become familiar with Scholar:

1. “An introduction to Scholar”
2. “Correct - Add a student”
3. “Correct - Remove a Student”

3.2 Usability Evaluation

This section describes how you will perform the usability evaluation. Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Record as many usability problem instances as you can. Even if the one person experiences the same usability problem more than one time, create a usability problem instance record for each one. For example, if a participant were to click on an incorrect link or button a second time, create a second usability problem instance record even if it is almost identical to the first record.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

Collect usability record data using the “Collect usability record data” link on the Collect and Review tab for the following task runs in the following order:

1. “s27 - Add a student”
2. “s67 - Add a student”
3. “s27 - Remove a student”

Review the usability record data for the task runs using the “Review usability records” link on the Collect and Review tab. You can review the task runs in any order that you like. Be sure to specify immediate intention for each record.

When you are done reviewing your records, select the “Version 2.2.x” level in the Levels View and the “Usability Records” link in the Collected Data Pools View. Work with your records and fill in the fields as necessary.

Select the “Generate a Usability Report” link in the Collected Data Pools View. Please save your report on the desktop with the name “Participant Scholar Report DCART.doc”. Edit the document as necessary.

You will have a maximum of 1.5 hours to perform the evaluation and generate a report.

Appendix B.12 Evaluator Study 2 Full Diagnosis Instructions

Overview

During this study, we will ask you to do the following:

1. Watch the DCART and immediate intention tutorial videos
2. Perform a familiarization exercise
3. Watch videos of users performing tasks with Scholar, a course management application, and document all usability problem instances that they encounter.

Part 1 – Tutorial Videos

Please double click the “DCART Familiarization” icon on the desktop. Please watch the following videos:

- “DCART Tutorial” (16 minutes)
- “UAF Diagnosis Tutorial” (10 minutes)

When you are finished, please close the tutorial window and close the welcome window.

Part 2 – Familiarization Exercise

2.1 General Familiarization with DCART

Please read and answer the following questions out loud to the study facilitator using the information in DCART:

1. In the “Familiarization Video” project, what is the participant’s name?
2. In the “Actual IMDB Path” task run, what benchmark task has the user performed?
3. What is the target value for the task in minutes?

2.2 Video of the Correct Way to Accomplish a Task

Go to the “Correct IMDB Path” task run and watch the video. This video will show you the correct way to accomplish the task of finding movies where two individuals are credited alongside one another in the Internet Movie Database (IMDB).

2.3 Identification of Usability Problem Instances

Read through this entire section before beginning. Go to the “Collect and Review” tab of the “Actual IMDB Path” task run and select the “Collect usability record data” link. Create usability records for two usability problem instances encountered by the participant.

When you have finished creating your two usability problem instances, close the “Usability Record Collection Form” and select the “Review usability records” link. Edit the text of your usability records and adjust their starting times as necessary. Be sure to specify immediate intention for each record.

When you are done reviewing your records, select the “Usability Records” link in the Collected Data Pools View. Work with your records and fill in the fields as necessary. Be sure to diagnose each record to 3 levels in the UAF.

Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

2.4 Generation of a Usability Report

Select the “Generate a Usability Report” link in the Collected Data Pools View. Save the file to the desktop as “Participant Familiarization Report DCART.doc”. Open the report and compare it to the report on the desktop titled “Practitioner Familiarization Report DCART.2Partial.doc”. This report was generated by a practitioner. Please read and answer the following questions out loud to the study facilitator:

1. In comparing the usability problem instances in your report to those in the practitioner’s report, which are similar?
2. Does your report contain any usability problem instances that the practitioner’s report does not?

Part 3 – Usability Evaluation of Scholar

Please close DCART and then double click the “DCART Scholar Evaluation” icon on the desktop. DCART is set up for a usability evaluation of Scholar, a course management system. First, you will watch some videos to become familiar with Scholar and the steps for adding and removing students from courses. You will then conduct a usability evaluation consisting of three task runs; in two of the task runs, participants are adding students, and in the other task run, the participant is removing a student.

3.1 Familiarization with Scholar

Open the following task runs in the following order and watch the videos associated with them using the “View video” link at the bottom of each task run to become familiar with Scholar:

1. “An introduction to Scholar”
2. “Correct - Add a student”
3. “Correct - Remove a Student”

3.2 Usability Evaluation

This section describes how you will perform the usability evaluation. Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Record as many usability problem instances as you can. Even if the one person experiences the same usability problem more than one time, create a usability problem instance record for each one. For example, if a participant were to click on an incorrect link or button a second time, create a second usability problem instance record even if it is almost identical to the first record.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

Collect usability record data using the “Collect usability record data” link on the Collect and Review tab for the following task runs in the following order:

1. “s27 - Add a student”
2. “s67 - Add a student”
3. “s27 - Remove a student”

Review the usability record data for the task runs using the “Review usability records” link on the Collect and Review tab. You can review the task runs in any order that you like. Be sure to specify immediate intention for each record.

When you are done reviewing your records, select the “Version 2.2.x” level in the Levels View and the “Usability Records” link in the Collected Data Pools View. Work with your records and fill in the fields as necessary. Be sure to diagnose each record to 3 levels in the UAF.

Select the “Generate a Usability Report” link in the Collected Data Pools View. Please save your report on the desktop with the name “Participant Scholar Report DCART.doc”. Edit the document as necessary.

You will have a maximum of 1.5 hours to perform the evaluation and generate a report.

Appendix B.13 Evaluator Study 2 Partial Diagnosis Familiarization Sample

Actual IMDB path (Task Run)
Location:
Virginia Tech > Familiarization Video > Actual > Session > Actual IMDB path
User Class:
Casual User - This individual uses the IMDB to get reviews for movies and find out information about his favorite movie stars.
Benchmark Task:
Credited alongside - Find all moves where Owen Wilson is credited alongside his brother Luke Wilson.
Usability Specification(s):

- Time on task - How long does it take a participant to complete the task.

Instances

1. No joint search
Start time:
0:0:28.88
Description:
The participant tried to use the search operator "and", but search operators are not supported. Because it is a search, the participant expects some form of operators.
User interface object:
The search at the top left corner of the screen
Designer Knowledge:
The search does not support search operators.
Stage of the interaction cycle:
Translation
Type of action:
Cognitive
Solution:
Two possible options are to either support search operators or to provide an advanced search option that uses a form-based approach to support search operators.

2. Participant doesn't understand how the results relate to his search query
Start time:
0:0:40.98
Description:
Because the participant entered a search term with operators, he

expected a fairly short list of results. Instead he is presented with an extensive list of results that do not appear to relate to his query.

User interface object:

Search results list

Stage of the interaction cycle:

Assessment

Type of action:

Cognitive

Solution:

One possible solution is to catch the fact that a user tried to use a search operator and provide feedback on the results page that search operators are not supported.

3. Not sure what the name search does

Start time:

0:1:9.48

Description:

There is no explanation as to whether a name is the name of a person, a character, a movie, etc.

User interface object:

The searches listed under "More Searches" at the bottom of a search results page.

Designer Knowledge:

I am unable to determine specifically what the "Name" search searches.

Stage of the interaction cycle:

Translation

Type of action:

Cognitive

Solution:

Provide a more specific term. For example, if the name search searched the real names of actors, then just use the term "Actor".

4. Overwhelming number of results for a name search

Start time:

0:1:23.50

Description:

The name search returned almost 1000 results in an uncategorized list. Such a result is overwhelming.

User interface object:

The search results list for the name search

Stage of the interaction cycle:

Assessment

Type of action:

Cognitive

Solution:
One solution is to show only the most relevant subset of the results. The second is to implement a paging mechanism to allow the user to show only a subset of the results at a time.

5. Option for the credited alongside search is scrolled off the screen

Start time:
0:2:15.37
Description:
The option for the credited alongside search is at the bottom of Owen Wilson's IMDB page.
User interface object:
The credited alongside search box
Stage of the interaction cycle:
Translation
Type of action:
Sensory
Solution:
Depending on its frequency of use, it may be appropriate to move it higher on the page. Regardless, it should still appear above the message boards, which typically mark the end of content provided by the site and the beginning of content provided by users.

6. Links at top of the joint search seem unrelated to the purpose of the search

Start time:
0:2:52.28
Description:
The links at the top do not have a readily understandable purpose and do not seem to relate to the results of the joint search.
User interface object:
Links like the following [wilson: 10412]
Designer Knowledge:
I'm not sure what the links refer to or why they appear on this page.
Stage of the interaction cycle:
Planning
Type of action:
Cognitive
Solution:
Without having better knowledge of the purpose of the links, I would suggest removing them from the page.

7. Character selection checkboxes appear after the Look up joint ventures button

Start time:

0:3:15.51

Description:

The checkboxes appear after the action button to which they are related.

User interface object:

Look up joint ventures button and actor checkboxes

Stage of the interaction cycle:

Translation

Type of action:

Cognitive

Solution:

One suggestion is to put the Look up joint ventures button after the actor categories and checkboxes.

8. Clicked the look up joint ventures button without selecting actors

Start time:

0:3:15.51

Description:

The participant clicked the Look up joint ventures button without selecting actors.

User interface object:

Look up joint ventures button and actor checkboxes

Stage of the interaction cycle:

Planning

Type of action:

Cognitive

Solution:

Requiring that users select roles after searching on joint ventures adds an extra step that users are not expecting. One solution is to show all joint ventures and then provide a mechanism for filtering by role.

9. Match names with any occupation is scrolled of the screen

Start time:

0:3:24.43

Description:

The option for finding joint ventures regardless of role (actor, director, etc) is below the categories of roles.

User interface object:

Match names with any occupation checkbox

Stage of the interaction cycle:

Translation

Type of action:

Cognitive

Solution:

One solution is to move the checkbox above the roles and actor checkboxes, so that users know that it exists before they take the time to check the checkboxes for several roles.

10. Can't distinguish between Luke Wilson (I) and (II)

Start time:

0:3:42.77

Description:

There are two entries for Luke Wilson that are differentiated by roman numerals in parantheses.

User interface object:

Actor name link with roman numerals

Designer Knowledge:

There are two different actors with the name Luke Wilson, so the (I) and (II) are used to distinguish between them.

Stage of the interaction cycle:

Translation

Type of action:

Cognitive

Solution:

The (I) and (II) distinguish between the two actors with the same name, but they are not user centered. It might be more appropriate to distinguish between them by middle name, for example.

Appendix B.14 Evaluator Study 2 Full Diagnosis Familiarization Sample

Actual IMDB path (Task Run)
Location:
Virginia Tech > Familiarization Video > Actual > Session > Actual IMDB path
User Class:
Casual User - This individual uses the IMDB to get reviews for movies and find out information about his favorite movie stars.
Benchmark Task:
Credited alongside - Find all moves where Owen Wilson is credited alongside his brother Luke Wilson.
Usability Specification(s):

- Time on task - How long does it take a participant to complete the task.

Instances

1. **No joint search**
   Start time:
   0:0:28.88
   Description:
   The participant tried to use the search operator "and", but search operators are not supported. Because it is a search, the participant expects some form of operators.
   User interface object:
   The search at the top left corner of the screen
   Designer Knowledge:
   The search does not support search operators.
   Stage of the interaction cycle:
   Translation
   Type of action:
   Cognitive
   Diagnosis:
   - User Action Framework [1]
     - Translation (design helping user know what physical action to make on what UI object) [3]
       - Content, meaning (of a cognitive affordance) -- clarity, precision, predictability, effectiveness [25]
         - Precise, correct, distinguishable, relevant expression of meaning (of cognitive affordance) [777]
           - Correct expression of meaning (of cognitive affordance) [651]

   Solution:

Two possible options are to either support search operators or to provide an advanced search option that uses a form-based approach to support search operators.

2. Participant doesn't understand how the results relate to his search query

Start time:

0:0:40.98

Description:

Because the participant entered a search term with operators, he expected a fairly short list of results. Instead he is presented with an extensive list of results that do not appear to relate to his query.

User interface object:

Search results list

Stage of the interaction cycle:

Assessment

Type of action:

Cognitive

Diagnosis:

- User Action Framework [1]
  - Assessment (Design of feedback and display of results helping user know if it worked) [6]
    - Issues about feedback (dialogue about interaction for task) [459]
      - Existence of feedback or indication of state or mode [213]
        - Existence of necessary or desirable feedback dialogue (rather than indicators of state, mode) [216]

Solution:

One possible solution is to catch the fact that a user tried to use a search operator and provide feedback on the results page that search operators are not supported.

3. Not sure what the name search does

Start time:

0:1:9.48

Description:

There is no explanation as to whether a name is the name of a person, a character, a movie, etc.

User interface object:

The searches listed under "More Searches" at the bottom of a search results page.

Designer Knowledge:

I am unable to determine specifically what the "Name" search searches.

Stage of the interaction cycle:

Translation
Type of action:
Cognitive
Diagnosis:
- User Action Framework [1]
  - Translation (design helping user know what physical action to make on what UI object) [3]
    - Content, meaning (of a cognitive affordance) -- clarity, precision, predictability, effectiveness [25]
      - User-centered, convincing expression of meaning (of cognitive affordance) [778]
        - User-centered content expression, design (of cognitive affordance) [382]

Solution:
Provide a more specific term. For example, if the name search searched the real names of actors, then just use the term "Actor".

4. Overwhelming number of results for a name search

Start time:
0:1:23.50
Description:
The name search returned almost 1000 results in an uncategorized list. Such a result is overwhelming.
User interface object:
The search results list for the name search
Stage of the interaction cycle:
Assessment
Type of action:
Cognitive
Diagnosis:
- User Action Framework [1]
  - Assessment (Design of feedback and display of results helping user know if it worked) [6]
    - Issues about information displays (results for task) [460]
      - Presentation (of information displays, results) [462]
        - Layout, spatial grouping by function, clutter in display [509]
          - Eliminate unnecessary information [700]

Solution:
One solution is to show only the most relevant subset of the results. The second is to implement a paging mechanism to allow the user to show only a subset of the results at a time.

5. Option for the credited alongside search is scrolled off the screen

Start time:

0:2:15.37

Description:

The option for the credited alongside search is at the bottom of Owen Wilson's IMDB page.

User interface object:

The credited alongside search box

Stage of the interaction cycle:

Translation

Type of action:

Sensory

Diagnosis:

- User Action Framework [1]
  - Translation (design helping user know what physical action to make on what UI object) [3]
    - Presentation (of a cognitive affordance) [26]
      - Sensory issues (of cognitive affordance) [67]
        - Findability/locatability of visible cognitive affordance [810]
          - Visibility of cognitive affordance [349]

Solution:

Depending on its frequency of use, it may be appropriate to move it higher on the page. Regardless, it should still appear above the message boards, which typically mark the end of content provided by the site and the beginning of content provided by users.

7. Links at top of the joint search seem unrelated to the purpose of the search

Start time:

0:2:52.28

Description:

The links at the top do not have a readily understandable purpose and do not seem to relate to the results of the joint search.

User interface object:

Links like the following [wilson: 10412]

Designer Knowledge:

I'm not sure what the links refer to or why they appear on this page.

Stage of the interaction cycle:

Planning

Type of action:

Cognitive

Diagnosis:

- User Action Framework [1]
  - Planning (Design helping user plan goals, tasks, how to use system) [2]
    - Task/step structuring and sequencing, work flow [769]
      - Matching work flow to user view of task structure [784]

- Flow of task in individual screen layout design [786]

Solution:

Without having better knowledge of the purpose of the links, I would suggest removing them from the page.

7. Character selection checkboxes appear after the Look up joint ventures button

Start time:

0:3:15.51

Description:

The checkboxes appear after the action button to which they are related.

User interface object:

Look up joint ventures button and actor checkboxes

Stage of the interaction cycle:

Translation

Type of action:

Cognitive

Diagnosis:

- User Action Framework [1]
  - Planning (Design helping user plan goals, tasks, how to use system) [2]
    - Task/step structuring and sequencing, work flow [769]
      - Matching work flow to user view of task structure [784]
        - Flow of task in individual screen layout design [786]

Solution:

One suggestion is to put the Look up joint ventures button after the actor categories and checkboxes.

8. Clicked the look up joint ventures button without selecting actors

Start time:

0:3:15.51

Description:

The participant clicked the Look up joint ventures button without selecting actors.

User interface object:

Look up joint ventures button and actor checkboxes

Stage of the interaction cycle:

Planning

Type of action:

Cognitive

Diagnosis:

- User Action Framework [1]
  - Planning (Design helping user plan goals, tasks, how to use system) [2]

- Goal decomposition [10]
  - Matching user conception of task organization [314]

Solution:

Requiring that users select roles after searching on joint ventures adds an extra step that users are not expecting. One solution is to show all joint ventures and then provide a mechanism for filtering by role.

9. Match names with any occupation is scrolled of the screen

Start time:

0:3:24.43

Description:

The option for finding joint ventures regardless of role (actor, director, etc) is below the categories of roles.

User interface object:

Match names with any occupation checkbox

Stage of the interaction cycle:

Translation

Type of action:

Cognitive

Diagnosis:

- User Action Framework [1]
  - Translation (design helping user know what physical action to make on what UI object) [3]
    - Presentation (of a cognitive affordance) [26]
      - Sensory issues (of cognitive affordance) [67]
        - Findability/locatability of visible cognitive affordance [810]
          - Visibility of cognitive affordance [349]

Solution:

One solution is to move the checkbox above the roles and actor checkboxes, so that users know that it exists before they take the time to check the checkboxes for several roles.

10. Can't distinguish between Luke Wilson (I) and (II)

Start time:

0:3:42.77

Description:

There are two entries for Luke Wilson that are differentiated by roman numerals in parantheses.

User interface object:

Actor name link with roman numerals

Designer Knowledge:

There are two different actors with the name Luke Wilson, so the (I) and (II) are used to distinguish between them.

Stage of the interaction cycle:

Translation

Type of action:

Cognitive

Diagnosis:

- User Action Framework [1]
    - Translation (design helping user know what physical action to make on what UI object) [3]
        - Content, meaning (of a cognitive affordance) -- clarity, precision, predictability, effectiveness [25]
            - User-centered, convincing expression of meaning (of cognitive affordance) [778]
                - User-centered content expression, design (of cognitive affordance) [382]

Solution:

The (I) and (II) distinguish between the two actors with the same name, but they are not user centered. It might be more appropriate to distinguish between them by middle name, for example.

Appendix B.15 Evaluator Study 3 Morae Instructions

Overview

During this study, we will ask you to do the following:

1. Watch the Morae tutorial video
2. Perform a familiarization exercise
3. Watch videos of users performing tasks with Scholar, a course management application, and document all usability problems that they encounter.

Part 1 – Tutorial Video

Please double click the “Morae Tutorial Video.wmv” icon on the desktop. Please watch the Morae tutorial video; it is approximately 8 minutes long. When you are finished, please close Windows Media Player.

Part 2 – Familiarization Exercise

2.1 Video of the Correct Way to Accomplish a Task

Double click the “Familiarization Video Correct.wmv” icon on the desktop and watch the video. This video will show you the correct way to accomplish the task of finding movies where two individuals are credited alongside one another in the Internet Movie Database (IMDB).

2.2 **Inserting Markers**

The study facilitator will now set up a familiarization session for you in Morae. Please read through the rest of this section before you begin.

Use the Remote Viewer to watch a participant perform the task of finding movies where two individuals are credited alongside one another in the IMDB. You have already seen a video of how to accomplish the task. Create markers for the following:

- The beginning and ending of the task
- Comments for two usability problem instances experienced by the participant

Notify the study facilitator when you have inserted the markers that you need to document two usability problem instances in your report. He will save your Morae recording and import it into a Morae Manager project.

Review your markers in Morae Manager and make any necessary changes, such as editing the text of the markers or adjusting the time of the markers.

Keep the following in mind as you work:

- Only describe usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not record it as a usability problem unless the participant in the video actually does experience it. For example, the terms "edit" and "revise" are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not document it as a usability problem.
- Be specific and provide detail in your usability problems. Someone who has not seen the task run video should be able to understand the usability problem from your description.

2.3 Generation of a Usability Report

When you are done reviewing your markers, create a usability report as demonstrated in the Morae tutorial video or by using a process of your own choosing. There are two requirements for your report:

- The report should contain only usability problems; you may need to combine or separate comments in your markers to achieve this.
- Each usability problem should have a timestamp associated with it.

Save the file to the desktop as "Participant Familiarization Report Morae.doc".

Open the report and compare it to the report on the desktop titled "Practitioner Familiarization Report Morae.3.doc". This report was generated by a practitioner. Please read and answer the following questions out loud to the study facilitator:

1. In comparing the usability problems in your report to those in the practitioner's report, which are similar?
2. Does your report contain any usability problems that the practitioner's report does not?

Part 3 – Usability Evaluation of Scholar

In this part, you will first watch three videos to become familiar with Scholar and the steps for adding and removing students from courses. You will then conduct a usability evaluation consisting of three tasks; in two of the tasks, participants are adding students, and in the other task, the participant is removing students.

3.1 Familiarization with Scholar

Please double click the following icons on the desktop in the following order and watch the videos:

1. "Scholar Introduction.wmv"

2. "Scholar Add Student Task Correct.wmv"
    b. The text for the task reads: A student emailed you to ask your permission to force add the course. Add him to the course. His pid is "psd_student_1".
3. "Scholar Remove Student Task Correct.wmv"
    c. The text for the task reads: On the first day of class, you realized that John Dewey has not taken the necessary prerequisites and is not eligible for the course. Remove him from the course.

3.2 Usability Evaluation

This section describes how you will perform the usability evaluation. Keep the following in mind as you work:

- This report will be given to the developers of Scholar. They will be reviewing it and assigning a measure of quality to it. They will be interested in how you consolidate and present your data. Consult the "Practitioner Familiarization Report Consolidated Morae.3.doc" on the desktop for ideas.
- Only describe usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not record it as a usability problem unless the participant in the video actually does experience it. For example, the terms "edit" and "revise" are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not document it as a usability problem.
- Be specific and provide detail in your usability problems. Someone who has not seen the task run video should be able to understand the usability problem from your description.

Use the Remote Viewer to watch the following participants perform the following tasks in the following order:

1. "s27 - Add a student"
2. "s67 - Add a student"
3. "s27 - Remove a student"

You have already seen videos of how to accomplish the tasks. Create markers for the following:

- The beginning and ending of the task
- Comments on usability problems experienced by the participant

When the task is finished, the study facilitator will save your Morae recording and import it into a Morae Manager project.

Review your markers in Morae Manager and make any necessary changes, such as editing the text of the markers or adjusting the time of the markers.

When you are done reviewing your markers, create a usability report as demonstrated in the Morae tutorial video or by using a process of your own choosing. Remember, this report will be seen by the developers of Scholar. Include information in your report in a format that will help them fix the usability problems in Scholar.

Save the file to the desktop as “Participant Scholar Report Morae.doc”.

You will have a maximum of 1.5 hours to perform the evaluation and generate a report.

Appendix B.16 Evaluator Study 3 DCART Instructions

Overview

During this study, we will ask you to do the following:

1. Watch the DCART tutorial video
2. Perform a familiarization exercise
3. Watch videos of users performing tasks with Scholar, a course management application, and document all usability problem instances that they encounter.

Part 1 – Tutorial Video

Please double click the “DCART Training” icon on the desktop. Please watch the following video:

- “DCART Tutorial – Full” (20 minutes)

When you are finished, please close the tutorial window and close the welcome window.

Part 2 – Familiarization Exercise

2.1 General Familiarization with DCART

Please read and answer the following questions out loud to the study facilitator using the information in DCART:

1. In the “Familiarization Video” project, what is the participant’s name?
2. In the “Actual IMDB Path” task run, what benchmark task has the user performed?
3. What is the target value for the task in minutes?

2.2 Video of the Correct Way to Accomplish a Task

Go to the “Correct IMDB Path” task run and watch the video. This video will show you the correct way to accomplish the task of finding movies where two individuals are credited alongside one another in the Internet Movie Database (IMDB).

2.3 Identification of Usability Problem Instances

Read through this entire section before beginning. Go to the “Collect and Review” tab of the “Actual IMDB Path” task run and select the “Collect usability record data” link. Create usability records for two usability problem instances encountered by the participant.

When you have finished creating your two usability problem instances, close the “Usability Record Collection Form” and select the “Review usability records” link. Edit the text of your usability records and adjust their starting times as necessary.

When you are done reviewing your records, select the “Usability Records” link in the Collected Data Pools View. Work with your records and fill in the fields as necessary.

Create an additional four sample usability problem instance records named “Instance 1”, “Instance 2”, “Instance 3”, and “Instance 4”. Merge “Instance 1” and “Instance 2” to form “Problem 1”. Merge “Instance 3” and “Instance 4” to form “Problem 2”. Now group “Problem 1” and “Problem 2” to form “Group 1”.

Keep the following in mind as you work:

- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

2.4 Generation of a Usability Report

Select the “Generate a Usability Report” link in the Collected Data Pools View. Save the file to the desktop as “Participant Familiarization Report DCART.doc”. Open the report and compare it to the report on the desktop titled “Practitioner Familiarization Report DCART.3.doc”. This report was generated by a practitioner. Please read and answer the following questions out loud to the study facilitator:

1. In comparing the usability problem instances in your report to those in the practitioner’s report, which are similar?
2. Does your report contain any usability problem instances that the practitioner’s report does not?

Part 3 – Usability Evaluation of Scholar

Please close DCART and then double click the “DCART Scholar Evaluation” icon on the desktop. DCART is set up for a usability evaluation of Scholar, a course management system. First, you will watch some videos to become familiar with Scholar and the steps for adding and removing students from courses. You will then conduct a usability evaluation consisting of three task runs; in two of the task runs, participants are adding students, and in the other task run, the participant is removing a student.

3.1 Familiarization with Scholar

Open the following task runs in the following order and watch the videos associated with them using the “View video” link at the bottom of each task run to become familiar with Scholar:

1. “An introduction to Scholar”
2. “Correct - Add a student”
3. “Correct - Remove a Student”

3.2 Usability Evaluation

This section describes how you will perform the usability evaluation. Keep the following in mind as you work:

- This report will be given to the developers of Scholar. They will be reviewing it and assigning a measure of quality to it. Usability problem instances are generally too specific for their needs – they will be interested in how you have merged and grouped usability problem instances into usability problems and usability problem groups. You will need to consolidate your findings into these two types of records.
- Only create usability problem instance records for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance record unless the participant in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the participants featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance record for these terms.
- Record as many usability problem instances as you can. Even if the one person experiences the same usability problem more than one time, create a usability problem instance record for each one. For example, if a participant were to click on an incorrect link or button a second time, create a second usability problem instance record even if it is almost identical to the first record.
- Be specific and provide detail in your usability problem instance records. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance record.

Collect usability record data using the “Collect usability record data” link on the Collect and Review tab for the following task runs in the following order:

1. “s27 - Add a student”
2. “s67 - Add a student”
3. “s27 - Remove a student”

Review the usability record data for the task runs using the “Review usability records” link on the Collect and Review tab. You can review the task runs in any order that you like.

When you are done reviewing your records, select the “Version 2.2.x” level in the Levels View and the “Usability Records” link in the Collected Data Pools View. Work with your records and fill in the fields as necessary. Be sure to merge similar usability problem instances into usability problems and group usability problems.

Select the “Generate a Usability Report” link in the Collected Data Pools View. Please save your report on the desktop with the name “Participant Scholar Report DCART.doc”. Remove any task run details and any usability problem instances. Only leave usability problem groups and usability problems in the report.

You will have a maximum of 1.5 hours to perform the evaluation and generate a report.

Appendix B.17 Evaluator Study 3 Morae Familiarization Sample

Individual Usability Problems

No joint search
Time: 0:1:47.88
The participant tried to use the search operator "and" in the search box at the top left corner of the screen, but search operators are not supported. Because it is a search, the participant expects some form of operators. Two possible options are to either support search operators or to provide an advanced search option that uses a form-based approach to support search operators.

Participant doesn't understand how the results relate to his search query
Time: 0:1:59.98
Because the participant entered a search term with operators, he expected a fairly short list of results. Instead he is presented with an extensive list of results that do not appear to relate to his query. One possible solution is to catch the fact that a user tried to use a search operator and provide feedback on the results page that search operators are not supported.

Not sure what the name search does
Time: 0:2:28.48
There is no explanation as to whether a name in the "More Searches" area is the name of a person, a character, a movie, etc. One option is to provide a more specific term. For example, if the name search searched the real names of actors, then just use the term "Actor".

Overwhelming number of results for a name search
Time: 0:2:22.50
The name search returned almost 1000 results in an uncategorized list. Such a result is overwhelming. Instead, the system could show only the most relevant subset of the results or provide a paging mechanism to allow the user to view only a subset of the results at a time.

Option for the credited alongside search is scrolled of the screen.
Time: 0:3:34.37
The option for the credited alongside search is at the bottom of Owen Wilson's IMDB page. Depending on its frequency of use, it may be appropriate to move it higher on the page. Regardless, it should still appear above the message boards, which typically mark the end of content provided by the site and the beginning of content provided by users.

Links at top of the joint search seem unrelated to the purpose of the search
Time: 0:4:11.28

The links at the top (example "[wilson: 10412]") do not have a readily understandable purpose and do not seem to relate to the results of the joint search. Without having better knowledge of the purpose of the links, I would suggest removing them from the page.

Character selection checkboxes appear after the Look up joint ventures button
Time: 0:4:34.51
The checkboxes appear after the action button to which they are related. One suggestion is to put the Look up joint ventures button after the actor categories and checkboxes.

Clicked the look up joint ventures button without selecting actors
Time: 0:4:34.51
The participant clicked the Look up joint ventures button without selecting actors. Requiring that users select roles after searching on joint ventures adds an extra step that users are not expecting. One solution is to show all joint ventures and then provide a mechanism for filtering by role.

Match names with any occupation is scrolled of the screen
Time: 0:4:43.43
The option for finding joint ventures regardless of role (actor, director, etc) is below the categories of roles. One solution is to move the checkbox above the roles and actor checkboxes, so that users know that it exists before they take the time to check the checkboxes for several roles.

Can't distinguish between Luke Wilson (I) and (II)
Time: 0:5:01.77
There are two entries for Luke Wilson that are differentiated by roman numerals in parentheses. The (I) and (II) distinguish between the two actors with the same name, but they are not user centered. It might be more appropriate to distinguish between them by middle name, for example.

Consolidated

This report is essentially based on one task. It would be necessary for us to run a number of different tasks before we could make any substantial claims or suggest system-wide changes. Based on our limited view of the system, however, it appears that layout and content of search result pages limits the ability of users to find information.

Search feature

Two major areas requiring improvement are as follows:

1. Providing explanations of what types of searches are supported. For example, search operators such as "and", "+", and "&" are not supported, but the user is never informed that they are not supported even if he uses them.

2. There are a number of categories of searches that are ambiguous. For example, does a name search find actor's names, character's names, or movie names?

Search results

Currently searches return large numbers of poorly organized results. Some suggestions for improving the search results would be returning fewer results with some obvious ordering (such as a relevance scale based on certain criteria) or providing a paging mechansim to allow users to view only a subset of the results at a time.

Visibility of UI objects

There are multiple instances of the participant having trouble finding needed UI objects (such as the joint ventures search) because they are scrolled off of the screen. See the individual problems in this group for solution suggestions.

Appendix B.18 Evaluator Study 3 DCART Familiarization Sample

Executive Summary

This report is essentially based on one task. It would be necessary for us to run a number of different tasks before we could make any substantial claims or suggest system-wide changes. Based on our limited view of the system, however, it appears that layout and content of search result pages limits the ability of users to find information.

Groups

1. Search feature

Description:

Two major areas requiring improvement are as follows:
1. Providing explanations of what types of searches are supported. For example, search operators such as "and", "+", and "&" are not supported, but the user is never informed that they are not supported even if he uses them.
2. There are a number of categories of searches that are ambiguous. For example, does a name search find actor's names, character's names, or movie names?

Usability problems contained in this group:

- Not sure what the name search does (from task run Actual IMDB path)
- Participant doesn't understand how the results relate to his search query (from task run Actual IMDB path)
- No joint search (from task run Actual IMDB path)

2. Search results

Description:

Currently searches return large numbers of poorly organized results. Some suggestions for improving the search results would be returning fewer results with some obvious ordering (such as a relevance scale based on certain criteria) or providing a paging mechansim to allow users to view only a subset of the results at a time.

Usability problems contained in this group:

- Overwhelming number of results for a name search (from task run Actual IMDB path)

3. Visibility of UI objects

Description:

There are multiple instances of the participant having trouble finding needed UI objects (such as the joint ventures search) because they are scrolled off of the screen. See the individual problems in this group for solution suggestions.

Usability problems contained in this group:

- Option for the credited alongside search is scrolled off the screen (from task run Actual IMDB path)
- Match names with any occupation is scrolled off the screen (from task run Actual IMDB path)

Problems

1. No joint search

Description:

The participant tried to use the search operator "and", but search operators are not supported. Because it is a search, the participant expects some form of operators.

User interface object:

The search at the top left corner of the screen

Designer Knowledge:

The search does not support search operators.

Solution:

Two possible options are to either support search operators or to provide an advanced search option that uses a form-based approach to support search operators.

2. Participant doesn't understand how the results relate to his search query

Description:

Because the participant entered a search term with operators, he expected a fairly short list of results. Instead he is presented with an extensive list of results that do not appear to relate to his query.

User interface object:

Search results list

Solution:

One possible solution is to catch the fact that a user tried to use a search operator and provide feedback on the results page that search operators are not supported.

3. Not sure what the name search does

Description:

There is no explanation as to whether a name is the name of a person, a character, a movie, etc.

User interface object:

The searches listed under "More Searches" at the bottom of a search results page.

Designer Knowledge:

I am unable to determine specifically what the "Name" search searches.

Solution:

Provide a more specific term. For example, if the name search searched the real names of actors, then just use the term "Actor".

4. Overwhelming number of results for a name search

Description:

The name search returned almost 1000 results in an uncategorized list. Such a result is overwhelming.

User interface object:

The search results list for the name search

Solution:

One solution is to show only the most relevant subset of the results. The second is to implement a paging mechanism to allow the user to show a only a subset of the results at a time.

5. Option for the credited alongside search is scrolled off the screen

Description:

The option for the credited alongside search is at the bottom of Owen Wilson's IMDB page.

User interface object:

The credited alongside search box

Solution:

Depending on its frequency of use, it may be appropriate to move it higher on the page. Regardless, it should still appear above the message boards, which typically mark the end of content provided by the site and the beginning of content provided by users.

6. Clicked the look up joint ventures button without selecting actors

Description:

The participant clicked the Look up joint ventures button without selecting actors.

User interface object:

Look up joint ventures button and actor checkboxes

Solution:

Requiring that users select roles after searching on joint ventures adds an extra step that users are not expecting. One solution is to show all joint ventures and then provide a mechanism for filtering by role.

7. Match names with any occupation is scrolled of the screen

Description:

The option for finding joint ventures regardless of role (actor, director, etc) is below the categories of roles.

User interface object:

Match names with any occupation checkbox

Solution:

One solution is to move the checkbox above the roles and actor checkboxes, so that users know that it exists before they take the time to check the checkboxes for several roles.

8. Can't distinguish between Luke Wilson (I) and (II)

Description:

There are two entries for Luke Wilson that are differentiated by roman numerals in parantheses.

User interface object:

Actor name link with roman numerals

Designer Knowledge:

There are two different actors with the name Luke Wilson, so the (I) and (II) are used to distinguish between them.

Solution:

The (I) and (II) distinguish between the two actors with the same name, but they are not user centered. It might be more appropriate to distinguish between them by middle name, for example.

9. Layout as it affects planning in a joint ventures search

Description:

The layout of certain interface objects on the look up joint ventures search makes it difficult for users to plan how to perform the search.

Designer Knowledge:

The user should first select actors using the checkboxes and then click the look up joint ventures button.

Solution:

The following are suggestions for fixing the problem

- Remove unnecessary search options (such as those like [wilson: 10412] from the main body of the page
- Place the actor checkboxes before the look up joint ventures button

Usability problem instances merged to form this problem:

- Character selection checkboxes appear after the Look up joint ventures button (from task run Actual IMDB path)
- Links at top of the joint search seem unrelated to the purpose of the search (from task run Actual IMDB path)

Appendix C Representative User Materials

Appendix C.1 Representative User Recruitment Email

Hi,

My name is Jonathan Howarth, and I am conducting a usability study of a course-management application and am looking for participants. Details are provided below:

IRB Approval: This study has been approved by the IRB.

Eligible participants: All VT graduate students are eligible to participate in this study.

Procedure: Participants will perform a series of tasks with the course-management application.

Date of studies: The study will take place between September 25 and October 6. Participants will be able to choose a date and time that is convenient for them from a list of available dates and times.

Location of study: The study will be conducted in 102 McBryde.

Compensation: All study participants will be paid a fixed fee of $20 in cash.

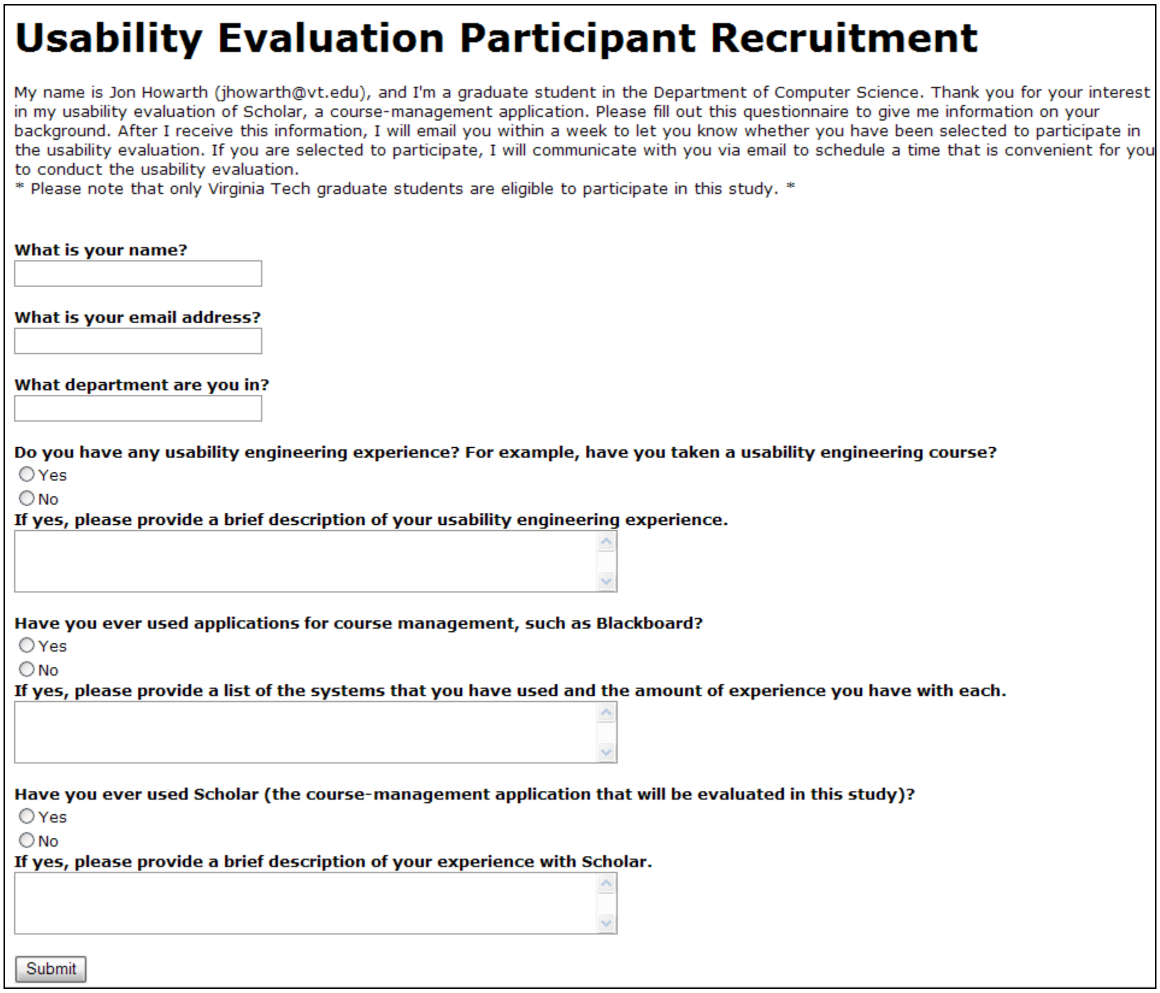
Time commitment: The study will take approximately 2 hours.

How to apply: Please fill out the survey at https://survey.vt.edu/survey/entry.jsp?surveyId=1158861528640. This survey provides me with information on your background. I will contact you via email within a week of receiving your survey submission.

Thanks,

- Jon Howarth

Appendix C.2 Representative User Background Survey



Usability Evaluation Participant Recruitment

My name is Jon Howarth (jhowarth@vt.edu), and I'm a graduate student in the Department of Computer Science. Thank you for your interest in my usability evaluation of Scholar, a course-management application. Please fill out this questionnaire to give me information on your background. After I receive this information, I will email you within a week to let you know whether you have been selected to participate in the usability evaluation. If you are selected to participate, I will communicate with you via email to schedule a time that is convenient for you to conduct the usability evaluation.
* Please note that only Virginia Tech graduate students are eligible to participate in this study. *

What is your name?

What is your email address?

What department are you in?

Do you have any usability engineering experience? For example, have you taken a usability engineering course?

○ Yes
○ No

If yes, please provide a brief description of your usability engineering experience.

Have you ever used applications for course management, such as Blackboard?

○ Yes
○ No

If yes, please provide a list of the systems that you have used and the amount of experience you have with each.

Have you ever used Scholar (the course-management application that will be evaluated in this study)?

○ Yes
○ No

If yes, please provide a brief description of your experience with Scholar.

Appendix C.3 Representative User Consent Form

Informed Consent for Participant of Investigative Project

Virginia Polytechnic Institute and State University
Department of Computer Science

Title of Project: Scholar Usability Evaluation

Investigators:
Dr. Rex Hartson, Professor, Computer Science, Virginia Tech
Jonathan Howarth, Graduate Student, Computer Science, Virginia Tech

I. The Purpose of this Research

You are invited to participate in a research study of Scholar, a course management system. Specifically, you will be performing tasks using Scholar to help me improve the system. Two to four other individuals will be performing the same tasks.

II. Procedures

This study will be conducted in McBryde 102 on the Virginia Tech campus. Jonathan Howarth will record audio and screen video as you perform tasks with Scholar. Some sample tasks include setting up a new course and adding students to the course. Jonathan Howarth is not evaluating you or your performance in any way; you are helping to find usability problems in Scholar. All information that you help attain will remain anonymous. Jonathan Howarth may ask you questions while you are working with the application. The session will last about two hours. The task is not very tiring, but you may take breaks if you wish.

III. Risks

There are no more than minimal risks associated with this study.

IV. Benefits of this Project

Your participation in this project will provide information that may be used to improve Scholar. No promise or guarantee of benefits has been made to encourage you to participate. If you would like to receive a synopsis or summary of this research when it is completed, please notify Jonathan Howarth.

V. Extent of Anonymity and Confidentially

The results of this study will be kept strictly confidential. The information you provide will have your name removed and only a participant number will identify you during analyses and any reports (written or video) of the research. Jonathan Howarth is the only individual that will have access to your name and participant number. He will generate a compilation video consisting of segments of your session and the sessions of the other participants for the developers of Scholar. Additionally, he may use this compilation video in future research studies. The Institutional Review Board (IRB) may also view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research. Jonathan Howarth will destroy any identifying information within three years of completion of the study.

VI. Compensation

Jonathan Howarth will pay you a fixed fee of $20 as compensation. You will receive a payment in cash when you have completed the study.

VII. Freedom to Withdraw

You are free to withdraw from this study at any time for any reason without penalty. If you choose to withdraw from the study and do not complete it, you will still receive $2.50 for each quarter hour that you have completed, up to a maximum of $20. You may also choose not to complete any part of the study, such as individual questions on a questionnaire, without penalty.

IX. Participant's Responsibilities

I voluntarily agree to participate in this study. I have the following responsibilities:

- To notify Jonathan Howarth at any time about a desire to discontinue participation.
- After completion of this study, I will not discuss my experiences with any other individual for a period of two months. This will ensure that everyone will begin the study with the same level of knowledge and expectations.

X. Participant's Permission

I have read and understand this informed consent form and the conditions of this study. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty.

______________________________________ ______________________

Signature Date

Should I have any questions about this research or its conduct, I may contact:

Dr. Rex Hartson, Investigator, hartson@vt.edu, (540)231-4857

Jonathan Howarth, Investigator, jhowarth@vt.edu, (540)961-5231

In addition, if you have detailed questions regarding your rights as a participant in University research, you may contact the following individual:

Dr. David Moore, Institutional Review Board Chair, moored@vt.edu, (540) 231-4991

Appendix C.4 Representative User Tasks

Task 1

You will be teaching a course entitled “The Philosophy of Software Design” in the upcoming semester. You have decided to use Scholar to administer the course. Please start up Internet Explorer and go to http://scholar.vt.edu. Please enter the information to create the course in Scholar, but do not submit it.

Task 2

A student emailed you to ask your permission to force add the course. Add him to the course. His pid is “psd_student_1”.

Task 3

Create a syllabus for the course that can be viewed by anyone, including visitors to the course site that are not enrolled in the class. The text for the syllabus is on the desktop: “Syllabus Text.doc”.

Task 4

The course will have two labs. One lab will be held Monday afternoons from 2:00 to 3:00 in McBryde 136. The other lab will be held on Tuesday afternoons from 3:30 to 4:30 in McBryde 126. Create the labs and put half of the students in the Monday lab and the other half of the students in Tuesday lab.

Task 5

You plan to have students work together in teams on certain assignments. Create five teams of six students each.

Task 6

Each team will lead an in-class discussion on a topic related to the course. The discussions will take place on Fridays. On the Monday before a team’s presentation, the team is expected to email the other students in the class with papers and resources describing the topic. Set up a class email address, so that students can send and receive email.

Task 7

On the first day of class, you realized that John Dewey has not taken the necessary prerequisites and is not eligible for the course. Remove him from the course.

Task 8

Add the class meeting times and lab meeting times to the course calendar. Consult the syllabus for the days, times, and locations. These calendar entries should began this week and end by December 1st.

Task 9

A student attended the first class and asked your permission to force add the course. Add him to the course. His pid is psd_student_17. Make sure to assign him to a lab section and put him on a team.

Task 10

One of your students raised an interesting question concerning the role of creativity in design. Set up a discussion thread for the students to continue the discussion.

Task 11

For the first homework, the students should read two journal papers that you have selected and write a one-page summary. The pdf files for the papers are on the desktop of your computer: “Paper 1.pdf” and “Paper 2.pdf”. The specification for the summary is also on the desktop: “Homework #1 Spec.doc”. Post both of these with the homework. Make the homework due one week from today. Make sure that this homework appears on the course calendar. You will assign a point value from 0 to 100 as the grade. Additionally, students can either paste their summary into the submission form or attach a Word document.

Task 12

You gave a quiz in class today. Post the grades from the quiz, so that the students can only see their own grades. The grades are in a spreadsheet on your desktop: “Quiz Grades.xls”.

Task 13

You prefer to grade homework submissions as you receive them. Two students have submitted homework #1. Give both students a grade of 95 and post the grades, so that the students can only see their own grades.

Task 14

Remove the student that dropped the course. His name is Blaise Pascal.

Task 15

Your second homework is a worksheet on software engineering. Make the homework due two weeks from today. Students shouldn’t see the assignment until one week from now. The worksheet document is on the desktop: “Worksheet.doc”. Additionally, make sure that the homework due date is on the calendar.

Task 16

Two more students have submitted homework #1. Give both students a grade of 95 and post the grades, so that the students can only see their own grades.

Task 17

You are scheduling a help session to take place the evening before the midterm, which is one month from today. Add a calendar entry for the session in McBryde 236 from 6:00 to 8:00 pm.

Appendix D   Modified SUPEX Outputs

Appendix D.1   Modified SUPEX Output for Representative User 1, Task 1

ID	Sub task	ID	Step	Time-stamp	Content
1.1	Find the page for adding students				
		1.1.1	Declare intention	00:14.4	"Now I'm going to look for how do I administer a course"
		1.1.2	Explore course options	00:20.3	Clicks on the Philosophy of Software Design 1 tab "Let me go to this here"
				00:37.5	Moused over Recent Announcements, Recent Discussion Items, Recent Chat Messages on the right hand side and Syllabus, Announcements, Gradebook, Email Archive, and Presentation menu bar items as well as the Users Present box "Recent Announcements, recent chat, syllabus, announcements, grade book, email archive, presentation, users present"
		1.1.3	Explore workspace options	00:37.5	Clicks the My Workspace tab
		1.1.4	Explore membership options	00:39.5	Clicks Membership menu bar item
				00:41.2	Highlights text on the Membership page "Just below includes all sites now"
		1.1.5	Explore worksite request options	00:45.3	Clicks on the Worksite Request menu bar item
				00:47.7	Clicks on the Yes button on the Security Alert dialog
				00:50.7	"I don't need to request a worksite. I need to go back here."
		1.1.6	Explore home options	00:51.0	Clicks the Home menu bar item
				00:56.6	Scrolls down and back up the Home page
		1.1.7 -> 1.1.2	Further explore course options	00:59.2	Clicks the Philosophy of Software Design 1 tab
		1.1.8	Explore resource options	01:02.8	Clicks on the Resources menu bar item

					"Resources"
				01:06.0	Upload no, permissions no, drop box, chat room"
		1.1.9	Explore gradebook options	01:08.1	Clicks on the Gradebook menu bar item "Gradebook"
				01:10.8	"Ok, can I here add assignment no, course grade no, ok there we are"
		1.1.10	Explore course grade options in the gradebook	01:13.2	Clicks on the Course Grade link
				01:18.5	Scrolled down and back up the page
				01:20.3	"And then maybe I can add a student here"
				01:22.0	No this is all, the number 1 is missing"
				01:26.9	"Add assignment"
		1.1.11	Explore roster options in the gradebook	01:28.5	Clicks on Roster link "Roster"
				01:30.7	"Find here, export, no, I still don't see add"
				01:35.8	Mouses over add assignment "Add assignment"
				01:37.6	Mouses over the top half of the menu bar
				01:39.3	"How do I add this guy"
		1.1.12 -> 1.1.6	Further explore home options	01:40.8	Clicks on the Home menu bar item "Home"
				01:41.2	Mouses over the right side of the Home screen and then back to the menu bar
				01:41.7	"We'll go back here"
		1.1.13	Explore syllabus options even though he knows that they are incorrect	01:48.3	Clicks on the syllabus menu bar item "Syllabus, obviously no"
		1.1.14 -> 1.1.8	Further explore resource options	01:50.3	Clicks on the Resources menu bar item "Resources"
				01:52.1	"Site resources, upload download, permissions"
				01:56.2	"Actions add"
		1.1.15	Explore the add a file options in resources	01:58.8	Clicked on the add link on the Resources page "Add a file"
				02:03.4	Clicks the browser"s back button "No, go back"

		1.1.16	Explain current state to the facilitator	02:05.8	Mouses over the menu bar items "The next thing that makes sense is"
				02:12.1	Facilitator: "So what are you looking for"
				02:13.5	"I'm trying to add a student, I want to go, um, I mean, um"
				02:16.6	Clicks on the Home menu bar item
				02:18.2	"I went to the roster"
				02:20.5	Clicks on the Gradebook menu bar item "So I went to the gradebook"
				02:22.0	Clicks on the Roster link in the gradebook "I was able to go to the roster"
				02:24.7	"These are all my students, but now I'm trying to add one of these kids"
				02:28.4	Scrolls down the page and over to the menu bar items
				02:30.1	"So I don't see a straight up add student"
				02:35.5	"So we are going to show all, there"s 28 guys"
				02:35.6	Clicks on the drop down box and selects the Show all option "So show all, there's 28 guys, obviously student 1 is missing, we need to add him"
				02:43.9	Clicks the sort icon next to the Student Name link "What is this, no"
				02:45.0	"This is sort"
				02:46.8	Add assignment"
				02:48.7	Clicks on the Grade Options link "Grade options"
				02:51.1	Scrolls down and back up the grade options page
				02:51.1	Clicks the add assignment link on the Gradebook page "No"
				02:53.3	Clicks on the Roster link
				02:55.1	"So I don"t see a way that I can add this guy, not yet, let me look around a little more"
		1.1.17	Explore help	03:02.7	Clicks on the Help menu bar item "Maybe I can look at help"
				03:05.7	Scrolls down the the list of help topics "Maybe I could look at"
		1.1.18	Explore gradebook entries in help	03:14.8	Clicks the Gradebook link "Gradebook"

				03:15.5	"Sorting gradebook tables, creating, adding, or editing assignments, entering or editing gradebook grades, details, no"
				03:28.2	Clicks on the Gradebook link to close it
		1.1.19	Explore permissions and roles entries in help	03:33.0	Clicked on the Permissions and Rules link "Permissions and Roles"
				03:36.2	Clicks on the Add/Edit/Delete Participant from Worksite Setup link "Add, edit participants to the worksite, there we go"
				03:40.8	Highlights text
				03:41.6	Highlights text
				03:42.6	"To view this, see permission, rules, etc"
				03:46.0	Clicks on the Permission, roles, and tools link
				03:47.8	Typed a backspace to return to the previous help page "Let's go back one second"
				03:51.4	Highlights text on the help page "Click worksite setup, ok"
				03:54.7	Closes help
		1.1.20	Attempt to follow directions specified in help	03:56.6	Clicks on the Section Info menu bar item "Section Info"
				04:02.0	Clicks on the My Workspace tab "My workspace"
				04:04.8	"And there will be a worksite setup, right there"
				04:05.7	Clicks on the worksite setup menu bar item
				04:07.2	"And here in this one"
				04:08.5	Clicks on the Philosophy of Software Design 1 link
				04:11.0	Presses the backspace key and returns to the previous page "Go back one second"
				04:17.0	Clicks on the Philosophy of Software Design 1 checkbox "Click on this guy"
		1.1.21	Return to help to reread the instructions	04:20.4	Clicks on the Help menu bar item "Obviously I didn"t clearly see it"
				04:24.0	Clicks on the Permissions and Rules link "Permissions and rules"

				04:25.6	Clicks on the Add/Edit/Delete Participants from Worksite Setup link "Add, edit, delete participants"
				04:29.5	"And then check the box, click revise"
				04:32.0	Highlights text on the help page
				04:35.1	"Where revise, revise"
				04:36.1	Switches focus to the main Scholar window
				04:38.4	"This is stupid"
				04:39.9	Clicks the Add Participants link "Add participants"
1.2	Add the student				
		1.2.1	Enter the pid	04:41.6	"User names, finally we are here"
				04:43.0	Clicks on the Username(s) text box and types in psd_student_1
				04:43.3	"Psd_student_1, that's his user name"
		1.2.2	Select how to assign a role	04:53.7	"Same role"
				04:54.1	Clicks on the radio button beside Assign each participant a role individually "Just make sure that I can assign him a student"s role"
		1.2.3	Determine the purpose of the guest email address text box	04:58.3	Clicks on the Guest Email Address text box
				04:59.4	Highlights a portion of the text "Multiple usernames are allowed"
				05:03.6	Clicks on the Guest Email Address text box
				05:04.7	"Email address, I don"t know it, doesn't matter, I won't put it in"
				05:08.7	Clicks the continue button "Continue"
		1.2.4	Assign the student role	05:10.5	Clicks on the Please select a role drop down box and selects the Student entry "Select role, student"
				05:12.1	Clicks on the continue button "And then continue"
		1.2.5	Select whether to send an email	05:14.7	"Ok an email can automatically be sent, don"t send"
				05:17.6	Clicks on the continue button
		1.2.6	Finish the addition of the student	05:19.2	"Ok, his name is Peter Abelard, I think that I"m done with this task 2"
				05:26.0	Facilitator: "Are you done with the task"

				05:28.1	Ok, finish and add him to the course"
				05:32.1	Clicks on the Finish button "Finish"
				05:34.6	"Yes, I am done with the course, done with the task"

Appendix D.2 Modified SUPEX Output for Representative User 2, Task 1

ID	Subtask	ID	Step	Time-stamp	Content
2.1	Find the page for adding students				
		2.1.1	Declare intention	00:11.6	"Ok, I guess I have to learn how to add people"
		2.1.2	Explore workspace options	00:22.0	"Oh, the front page is just telling me that it is working on this"
				00:24.9	Mouses over several menu bar items and pauses on Membership
		2.1.3	Explore worksite setup options	00:32.8	Clicks on the Worksite Setup menu bar item "I'm going to assume that it is worksite setup"
				00:37.3	Clicks on the new link "New"
				00:43.9	Clicks on the browser"s back button "D***it, back"
				00:46.6	"Awhh (a grunt)"
		2.1.4	Add a worksite	00:47.8	Clicks on the Worksite Setup menu bar item "Worksite setup"
				00:51.5	"I just create this, is there nothing else that I can do"
				00:55.7	Clicks on the Continue button
				00:56.3	"Let's see"
				01:01.1	Clicks on the text Subject text box "Oh, add more to the roster, yes"
				01:08.4	"But I don't have the CRN number and stuff"
				01:11.2	"So what am I supposed to do"
				01:14.2	Clicks on the Add More Roster(s) drop down menu and selects the 1 more entry
				01:17.0	Clicks on the Add More Roster(s) drop down menu and selects the 1 more entry
				01:17.1	"Ah, I just want to add one person"
				01:20.8	Facilitator: "Do you believe you are adding a person here"
		2.1.5	Realize that he is adding a course and not a person	01:23.7	"I"m adding a class aren't I, ya, d****it"

				01:32.2	"But do I already have a class"
				01:34.7	Facilitator: "You do have a class, you just created the class in the last task"
				01:35.3	Clicks on the Home menu bar item
		2.1.6	Search for a list of classes	01:41.5	"Then where is my list of classes, or just a class list"
				01:47.1	"Where's the button that says class list"
				01:51.3	Clicks on the Membership menu bar item
				01:51.5	"Maybe I belong to a class"
				01:53.0	"Ah, I"m a member of a class"
				01:53.3	Clicks on the Philosophy of Software Design 3 link
				01:59.0	"OK"
		2.1.7	Determine if there is a way to perform the task	02:03.1	Mouses over several menu bar items
				02:08.4	Scrolls down the home page
				02:11.6	"See I could say that it would be great if there was a button for every task that you listed, but that would be cheating"
				02:18.8	Facilitator: "There is in fact a way to do it, there is functionality provided"
		2.1.8	Explore gradebook options	02:26.5	"Well I'm here, let's see, gradebook"
				02:29.1	Clicks the Gradebook menu bar item
				02:32.1	"Someone has got to be in here, these are add assignments, so I assume that its the list of people already"
		2.1.9	Explore resource options	02:38.8	Selects the Resources menu bar item "Resources"
				02:50.6	Selects the Revise link "Revise"
				02:54.3	Scrolls down and back up the page
				03:02.5	Mouses over the menu bar items
		2.1.10	Explore site info options	03:05.6	Selects the Site Info menu bar item
				03:09.7	Mouses over the Edit Site Information, Edit Tools, and Manage Groups links "Edit site information, edit tools, manage groups"
				03:13.8	"Add participants, that's gotta be people"
2.2	Add the participant				
		2.2.1	Determine if username is pid	03:17.1	Clicks the Add Participants link

				03:20.0	Clicks on the Username(s) text box
				03:21.6	"So, is username pid"
				03:23.9	Facilitator: "Do you think it is correct, can you distinguish between what the two boxes are asking for"
		2.2.2	Determine difference between the text boxes	03:37.7	"Yea, it seems like the first box is people internal to the course and external to the course"
				03:42.7	Clicks on the Guest(s) Email Address text box
				03:45.4	"So I assume that it goes in the first box"
				03:50.5	"But I don't know if username is the same as pid in this case"
				03:53.0	Types psd_student_1 in the Username box
				04:01.4	"Continue"
		2.2.3	Select how to assign roles	04:06.9	"Assign each person a role individually"
				04:12.4	Scrolls up the page
				04:13.3	Clicks the Assign each participant a role individually radio button
				04:14.0	"Ok, we"ll see what kind of roles I can assign"
				04:14.4	Clicks the continue button
		2.2.4	Assign the student role	04:16.5	"I want him to be a student"
				04:18.6	Clicks on the Please select a role drop down menu and selects the Student item
				04:19.4	Clicks on the continue button
		2.2.5	Select whether to send an email	04:20.7	"An email can be automatically, probably don"t send an email"
				04:24.9	Clicks on the continue button
		2.2.6	Finish the addition of the student	04:28.7	"Ok, finish"
				04:29.3	Clicks on the finish button

Appendix D.3 Modified SUPEX Output for Representative User 1, Task 2

ID	Subtask	ID	Step	Time-stamp	Content
3.1	Find the page for removing students				
		3.1.1	Explore gradebook options	00:13.4	Clicks on the Gradebook menu bar item "Gradebook"
		3.1.2	Explore roster options in the gradebook	00:14.7	Clicks on the roster link "Roster"
				00:17.1	"Kill this guy"
				00:20.7	Laughs
				00:20.8	Scrolls down the page
				00:24.3	"This is not the place that I need to be"
		3.1.3	Explore worksite setup options	00:25.1	Clicks on the My Workspace tab
				00:27.4	Clicks on the Worksite Setup menu bar item "Worksite setup"
				00:29.9	"Um, no"
				00:33.3	"I just knew where this was"
				00:34.8	Moves the mouse pointer around the screen
		3.1.4	Explain the difference between the workspace and worksites to the facilitator	00:37.6	Facilitator: "Have you determined the difference between your workspace and the philosophy of software design 1"
				00:43.4	"Yea, this is like, um, giving me, allowing me to create access to this area"
				00:50.6	Facilitator: "This area is"
				00:52.4	"Which is eh, which is uh, which is uh, this one"
		3.1.5 -> 3.1.2	Further explore roster options	00:54.2	Clicks on the Roster link
				00:58.0	Clicks on the Overview link
				00:59.5	"This is the gradebook"
				01:00.0	Clicks on the Roster link "Roster"

				01:05.5	"Now at one point I was at a place where I could do stuff to these people"
		3.1.6	Explore resource options	01:10.3	Clicks on the Resources menu bar item
		3.1.7	Explore site info options	01:16.8	Clicks on the Site Info menu bar item
		3.1.8	Explore class roster options in site info	01:21.1	Clicks on the Edit Class Roster link "Edit class roster"
				01:23.3	Clicks on the Cancel button "Cancel"
				01:25.0	Scrolls down the page
3.2	Remove the student				
		3.2.1	Select the student	01:29.6	"Go here and remove this guy simply, there we go, what is his name, John Dewey"
				01:35.1	Clicks the checkbox to the right of John Deweys entry
		3.2.2	Edit the student's options	01:38.9	Clicks on the drop down menu labeled Student
				01:39.0	"I don't need to change something here, no"
				01:41.7	Clicks the drop down menu labeled Active and changes the value to Inactive
				01:41.7	"Make him inactive as well"
		3.2.3	Remove the student	01:45.7	Clicks on the Update Participants button "Update participants"
		3.2.4	Confirm the removal of the student	01:46.9	"Ok, make sure that he is not there any more. He"s not there"

Appendix E Instance Coder Materials

Appendix E.1 Instance Coder Consent Form

Informed Consent for Participant of Investigative Project

Virginia Polytechnic Institute and State University
Department of Computer Science

Title of Project: Addressing Usability Engineering Process Effectiveness with Tool Support

Role of Participant: Instance Coder

Investigators:
Dr. Rex Hartson, Professor, Computer Science, Virginia Tech
Jonathan Howarth, Graduate Student, Computer Science, Virginia Tech

I. The Purpose of this Research

You are invited to participate in a research study of usability engineering tools. Specifically, you will be creating a master list of usability problem instances in a video of representative users using Scholar and applying it to lists produced by other participants in the study. There is one other participant in this study performing the same task as you.

II. Procedures

Your participation in this study involves two parts. In the first part, you will watch a video of representative users performing tasks with Scholar and apply a process to develop a list of usability problem instances. You can do this in a place of your choosing. You have a week to perform this step. You will then compare your list with the list of the other instance coder in a meeting arranged by Jonathan Howarth and reconcile any differences. The list that results will be the master list of usability problem instances in the video. In the second part, you will compare the master list with lists produced by other participants in the study and note any differences. You have two weeks to complete this part. Jonathan Howarth is not evaluating you or your performance in any way; you are helping him to evaluate the reports. All information that you help him attain will remain anonymous. He may ask you questions during either of the two parts. Your total time commitment is expected to be 25 hours.

III. Risks

There are no more than minimal risks associated with this study.

IV. Benefits of this Project

Your participation in this study will provide information that may be used to improve usability engineering tools. No promise or guarantee of benefits has been made to encourage you to participate. If you would like to receive a synopsis or summary of this research when it is completed, please notify Jonathan Howarth.

V. Extent of Anonymity and Confidentially

The results of this study will be kept strictly confidential. At no time will the results of the study be released to anyone other than individuals working on the project without your written consent. It is possible, however, that the Institutional Review Board (IRB) may view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research. The information you provide will have your name removed and only a participant number will identify you during analyses and any written reports of the research. The only individual that will have access to your name and participant number is Jonathan Howarth. He will destroy any identifying information within three years of completion of the study.

VI. Compensation

You will not receive compensation for your participation in the study.

VII. Freedom to Withdraw

You are free to withdraw from this study at any time for any reason without penalty. You may also choose not to complete any part of the study, such as individual questions on a questionnaire, without penalty.

IX. Participant's Responsibilities

I voluntarily agree to participate in this study. I have the following responsibilities:

- To notify Jonathan Howarth at any time about a desire to discontinue participation.
- After completion of this study, I will not discuss my experiences with any other individual for a period of two months. This will ensure that everyone will begin the study with the same level of knowledge and expectations.

X. Participant's Permission

I have read and understand this informed consent form and the conditions of this study. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty.

____ ______________________

Signature Date

Should I have any questions about this research or its conduct, I may contact:

Dr. Rex Hartson, Investigator, hartson@vt.edu, (540)231-4857

Jonathan Howarth, Investigator, jhowarth@vt.edu, (540)961-5231

In addition, if you have detailed questions regarding your rights as a participant in University research, you may contact the following individual:

Dr. David Moore, Institutional Review Board Chair, moored@vt.edu, (540) 231-4991

Appendix E.2 Instance Coder Instructions

Overview

During this study, we will ask you to do the following:

- Task 1 - Meet as a group to learn about the process that you will be using to create a master list of usability problem instances and practice identifying usability problem instances
- Task 2 - Watch a video of users performing tasks (about 14 minutes) and create lists of usability problem instances
- Task 3 - Meet as a group to compare your lists of usability problem instances and reconcile them to create a master list
- Task 4 – Compare lists of usability problem instances produced by the evaluators to the master list
- Task 5 – Confirm any reconciliations and additions to the master list

Task 1

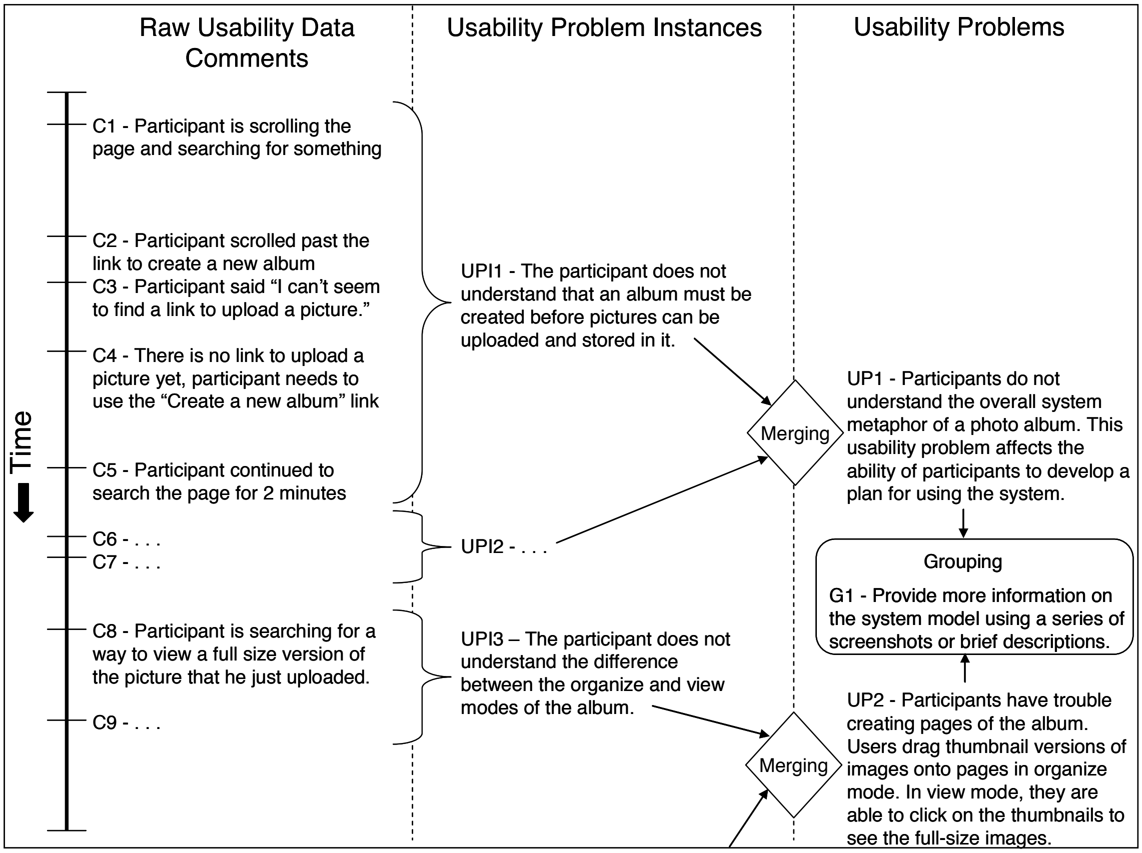
This task is intended to familiarize you with the process that we will be using to establish a master list of usability problem instances. During this task, you will be asked to do the following:

- Familiarize yourself with the levels of usability problem data
- Practice identifying and documenting usability problem instances

This task is expected to last two hours.

1.1 Levels of Usability Problem Data

A usability problem describes the effect that an interaction design flaw has on the user. Usability problems are documented with usability problem descriptions and represent analyzed usability problem information. The same usability problem may be experienced by multiple participants or multiple times by one participant. Each occurrence of a usability problem as encountered by a participant and observed by the evaluator is a usability problem instance. Usability problem instances are determined by analyzing the raw usability problem data produced by the facilitator during the collection stage. The following figure shows example usability data for a photo application.


Raw Usability Data Comments
Usability Problem Instances
Usability Problems
Time
C1 - Participant is scrolling the page and searching for something
C2 - Participant scrolled past the link to create a new album
C3 - Participant said "I can't seem to find a link to upload a picture."
C4 - There is no link to upload a picture yet, participant needs to use the "Create a new album" link
C5 - Participant continued to search the page for 2 minutes
C6 - . . .
C7 - . . .
C8 - Participant is searching for a way to view a full size version of the picture that he just uploaded.
C9 - . . .
UPI1 - The participant does not understand that an album must be created before pictures can be uploaded and stored in it.
UPI2 - . . .
UPI3 – The participant does not understand the difference between the organize and view modes of the album.
Merging
Merging
UP1 - Participants do not understand the overall system metaphor of a photo album. This usability problem affects the ability of participants to develop a plan for using the system.
Grouping
G1 - Provide more information on the system model using a series of screenshots or brief descriptions.
UP2 - Participants have trouble creating pages of the album. Users drag thumbnail versions of images onto pages in organize mode. In view mode, they are able to click on the thumbnails to see the full-size images.

Levels of usability problem data for a photo album application

1.2 Practice

During this practice session, you will be asked to watch a video of a correct way to perform a task using the Internet Movie Database (IMDB). You will then be asked to watch a video of a user actually performing the task and to create a list of usability problem instances.

There is no time limit for this practice exercise. All files referenced are located in the "Practice" folder on the CD that we gave you.

1.2.1 Video of a Correct Way

Please watch the "IMDB Correct Practice Video.wmv" video. This video will show you the correct way to accomplish the task of finding movies where two individuals are credited alongside one another in the Internet Movie Database (IMDB).

1.2.2 Video of the Actual Way and Creation of a List of Usability Problem Instances

Please watch the “IMDB Actual Practice Video.wmv” video. This video shows an actual user trying to perform the task of finding movies where two individuals are credited alongside one another in the IMDB. You may rewind, pause, and fast forward the video as much as you like.

Create a list of usability problem instances in a Word document. Each instance should contain a name, a timestamp, a description, and a severity rating. For the severity rating, please assign one of the following values:

- Minor – A minor problem will result in the participant being misdirected or hesitating for a few seconds. The participant may express mild frustration.
- Severe – A severe problem will result in the participant being misdirected or stalled for more than a few seconds. The participant may express extreme frustration.

Keep the following in mind as you work:

- Only create usability problem instances for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance unless the user in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the users featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance for these terms.
- Be specific and provide detail in your usability problem instances. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance.

1.2.3 Comparison with a Reference List of Usability Problem Instances and Reconciliation as a Group

When you are finished, please open “IMDB Reference List Practice.doc” and compare your usability problem instances with those in it. There are two lists in the document: Instances of Usability Problems Experienced by the User and Usability Problems Not Experienced by the User. As per the directions above, your list of usability problem instances should not contain those in the second list.

We will discuss the usability problem instances in your list and in the reference list and solve any issues.

Task 2

This task is intended to produce lists of usability problem instances for videos of users performing tasks with Scholar, a course management application. During this task, you will be asked to do the following:

- Watch some videos to become familiar with Scholar and the steps for adding and removing students from courses.
- Create lists of usability problem instances for two videos of users adding students to a course in Scholar and one video of a user removing a student from a course.
- Match individual usability problem instances to steps.

This task is expected to take three to five hours. All files referenced are located in the “Study” folder on the CD that we gave you. Please email the investigator your file(s) when you are finished.

2.1 Familiarization with Scholar and Videos of Correct Ways to Perform Tasks in Scholar

To familiarize yourself with Scholar and the correct way for adding and removing students, please watch the following videos:

- “Scholar Introduction.wmv”
- “Scholar Correct Add Student.wmv”
    - The text for the task reads: A student emailed you to ask your permission to force add the course. Add him to the course. His pid is psd_student_1.
- “Scholar Correct Remove Student.wmv”
    - The text for the task reads: On the first day of class, you realized that John Dewey has not taken the necessary prerequisites and is not eligible for the course. Remove him from the course.

You can refer back to these videos at any time during the study and rewind, pause, and fast forward them as much as you like.

2.2 Videos of the Actual Users and Creation of Lists of Usability Problem Instances

Please watch and create lists of usability problem instances in a Word document for the following three videos:

- “s27 Task 2.wmv” (add a student)
- “s67 Task 2.wmv” (add a student)
- “s27 Task 7.wmv” (remove a student)

You can refer back to these videos at any time during the study and rewind, pause, and fast forward them as much as you like.

Each instance should contain a name, a timestamp, a description, and a severity rating. For the severity rating, please assign one of the following values:

- Minor – A minor problem will result in the participant being misdirected or hesitating for a few seconds. The participant may express mild frustration.
- Severe – A severe problem will result in the participant being misdirected or stalled for more than a few seconds. The participant may express extreme frustration.

Keep the following in mind as you work:

- Only create usability problem instances for usability problems that the participant experiences in the video. Even if you see something in the video that could result in a user experiencing a usability problem, do not create a usability problem instance unless the user in the video actually does experience a problem as a result of it. For example, the terms “edit” and “revise” are used inconsistently and interchangeably in Scholar. A practitioner would normally note this as a usability problem, but the users featured in this study do not experience a usability problem as a result of the terms and so you should not create a usability problem instance for these terms.
- Be specific and provide detail in your usability problem instances. Someone who has not seen the task run video should be able to understand the usability problem from the text in your usability problem instance.

2.3 Matching Individual Usability Problem Instance to Steps

After you have created your lists of usability problem instances, please match them to individual steps defined in the following files:

- “s27 Task 2 SUPEX.xls” for the “s27 Task 2.wmv” video
- “s27 Task 7 SUPEX.xls” for the “s27 Task 7.wmv” video
- “s67 Task 2 SUPEX.xls” for the “s67 Task 2.wmv” video

For example, if you identify a usability problem instance in the “s27 Task 2.wmv” video that maps to step 1.1.5 in the “s27 Task 2 SUPEX.xls” file, append [1.1.5] to the end of the name of the usability problem instance.

Task 3

The purpose of this task is to produce a master usability problem instance list. You will be asked to do the following:

- Integrate and reconcile the lists of usability problem instances that you produced in the previous task

This task is expected to take two hours.

3.1 Preparation Performed by the Investigator

The investigator will assign unique ids to the usability problem instances that you emailed to him in the previous task. The investigator will also print the lists of usability problem instances with the assigned ids and bring them to the group meeting.

3.2 Integrating Lists of Usability Problem Instances

Please decide who will present usability problem instances from his or her list and who will match with those on his or her list.

You can use the following (listed in no particular order) to determine if two usability problem instances are the same:

- Description
- Step
- Timestamp

For each usability problem instance in your list, you will need to decide the following:

- If it matches with a usability problem instance in the other instance coder's list
- If it does not match with a usability problem instance in the other instance coder's list and which of the following it represents:
    - A real usability problem instance that has been omitted from the other instance coder's list
    - A false positive or usability problem instance that does not exist and should not have been included in your list

Additionally, for each usability problem instance that you choose to include in the master list, you'll need to agree on the severity rating.

The investigator will take notes and record your decisions while you discuss the usability problem instances in your lists.

3.3 Additional Work Performed by the Investigator

Based on your decisions, the investigator will create the master usability problem instance list. The investigator will email the list to you for approval.

Task 4

The purpose of this task is to compare lists of usability problem instances produced by evaluators to the master list. You will be asked to do the following:

- Review each evaluator's list of usability problem instances and assign each usability problem instance a code

This task is expected to take eight to ten hours. All files referenced are located in the “Study” folder on the CD that we gave you. Please email the investigator your file when you are finished.

4.1 Comparisons

Please open the “Comparisons.xls” spreadsheet. The Introduction worksheet explains how to code usability problem instances and in what order you should review the lists of usability problem instances produced by evaluators. The lists of instances produced by the evaluators are in the “Instances” folder. The master list of usability problems is titled “masterList.doc”.

4.2 Concerning Unfinished Evaluator Documents

All evaluators had 1.5 hours to watch the videos of the participants performing tasks and create a list of usability problem instances. Some evaluators did not finish. The following text appears at the top of the documents of those who did not finish, but still were able to document all their usability problem instances to a reasonable degree: “* The evaluator ran out of time and did not completely finish documenting the usability problem instances in the report. *”. Others were not able to document all their instances to a reasonable degree; the undocumented usability problem instances were removed from their reports. The following text appears at the top of their reports: “* The evaluator ran out of time and did not completely finish documenting the usability problem instances in the report. Some instances were removed. *”

Task 5

The purpose of this task is to reconcile disagreements in task 4 and to finalize the master list. You will be asked to do the following:

- Reconcile
- Confirm additions to the master list of usability problem instances

This task is expected to take you 15 minutes.

5.1 Preparation Performed by the Investigator

The investigator will review the “Comparisons.xls” spreadsheets that you emailed him in the previous task and use the following process to reconcile the lists:

1. If both instance coders agree, the reconciled value is the agreed upon value.
2. If both instance coders assign values that represent existing problems in the master problem list, but these values do not agree, the investigator will reconcile them and decide upon the final value. This value will be approved by instance coders in task 5.
3. If one instance coder marks an evaluator’s usability problem instance as an N, indicating that it does not represent a usability problem instance

experienced by the participant, then the reconciled value is an N. A single N value is sufficient to dismiss an evaluator's usability problem instance.

4. If both instance coders assign a value of A to a usability problem instance, the investigator will add it to the master list of usability problem instances. All additions will be approved by instance coders in task 5. If only one instance coder assigns a value of A, the reconciled value is the other instance coder's value. Both instance coders must agree to add a usability problem instance.

5.2 Confirmation of Reconciliations and Additions

The investigator will email you with a list of reconciliations and additions to update the master usability problem instance list. Please review this email. If you would like to suggest any changes, please email those to the investigator. Once all your changes have been addressed, please email the investigator that you approve the reconciliations and additions.

Appendix E.3 Instance Coder Practice Reference List

Instances of Usability Problems Experienced by the User

No joint search
Time: 0:1:47.88
The participant tried to use the search operator "and", but search operators are not supported. Because it is a search, the participant expects some form of operators. Two possible options are to either support search operators or to provide an advanced search option that uses a form-based approach to support search operators.

Participant doesn't understand how the results relate to his search query
Time: 0:1:59.98
Because the participant entered a search term with operators, he expected a fairly short list of results. Instead he is presented with an extensive list of results that do not appear to relate to his query. One possible solution is to catch the fact that a user tried to use a search operator and provide feedback on the results page that search operators are not supported.

Not sure what the name search does
Time: 0:2:28.48
There is no explanation as to whether a name is the name of a person, a character, a movie, etc. One option is to provide a more specific term. For example, if the name search searched the real names of actors, then just use the term "Actor".

Overwhelming number of results for a name search
Time: 0:2:22.50
The name search returned almost 1000 results in an uncategorized list. Such a result is overwhelming. Instead, the system could show only the most relevant subset of the results or provide a paging mechanism to allow the user to view only a subset of the results at a time.

Option for the credited alongside search is scrolled of the screen
Time: 0:3:34.37
The option for the credited alongside search is at the bottom of Owen Wilson's IMDB page. Depending on its frequency of use, it may be appropriate to move it higher on the page. Regardless, it should still appear above the message boards, which typically mark the end of content provided by the site and the beginning of content provided by users.

Links at top of the joint search seem unrelated to the purpose of the search
Time: 0:4:11.28
The links at the top do not have a readily understandable purpose and do not

seem to relate to the results of the joint search. Without having better knowledge of the purpose of the links, I would suggest removing them from the page.

Clicked the look up joint ventures button without selecting actors
Time: 0:4:34.51
The participant clicked the Look up joint ventures button without selecting actors. Requiring that users select roles after searching on joint ventures adds an extra step that users are not expecting. One solution is to show all joint ventures and then provide a mechanism for filtering by role.

Match names with any occupation scrolled off the screen
Time: 0:4:43.43
The option for finding joint ventures regardless of role (actor, director, etc) is below the categories of roles. One solution is to move the checkbox above the roles and actor checkboxes, so that users know that it exists before they take the time to check the checkboxes for several roles.

Can't distinguish between Luke Wilson (I) and (II)
Time: 0:5:01.77
There are two entries for Luke Wilson that are differentiated by roman numerals in parentheses. The (I) and (II) distinguish between the two actors with the same name, but they are not user centered. It might be more appropriate to distinguish between them by middle name, for example.

Usability Problems Not Experienced by the User

Name search returns inconsistent number of results
Time: 0:2:42.50
On the participant's first search, the name search of the results only contained 7 entries. On the participant's second search, which was a name search, the search returned almost 1000 names. A fix is to provide some explanation or rationale for why a certain number of results are displayed in each category of the "All" search. Currently, the categories displayed after an all search each have different numbers of results; the number for each category is also different than if a specific search were done on the category.

Checkboxes are after the actors' names
Time: 0:4:34.51
The checkboxes appear to the right of the actors' names. Having checkboxes to the right violates industry standards. The checkboxes belong on the left.

Appendix E.4 Instance Coder Comparison Instructions and Spreadsheet

Introduction

In your role as an instance coder, you will use this spreadsheet to compare evaluator's lists of usability problem instances to the master list of usability problem instances. There are a number of worksheets in this spreadsheet; you can navigate among them using the tabs at the bottom of the window. Each worksheet contains a list of usability problem instance numbers that correspond to usability problem instances produced by one evaluator; to see the problems referenced by the numbers, open the Word document that has the same name as the worksheet. Please compare the lists to the master list of usability problem instances in the order that the worksheets appear in the tabs (begin with the leftmost tab and continue in order to the rightmost tab).

Notation

Below is an example to show the notation that you will you use as you compare lists of usability problem instances to the master list of usability problem instances

# of the Evaluator's UP Instance	Comparison	
1	m2	If the usability problem instance in the evaluator's list matches directly to one in the master list, put the id of the usability problem instance in the master list in the Comparison column. For example the first usability problem instance in the evaluator's list matches to the usability problem instance with id m2 in the master list. To qualify as a match, you must be sure without a doubt that the evaluator's instance matches the instance in the master list. For example, if the evaluator's description of the instance is too terse or too general, do not count it as a match
2	N	If the usability problem instance in the evaluator's list does not represent a usability problem instance experienced by the participant, put a "N" (for no) in the Comparison column.
3	A	If the usability problem instance in the evaluator's list is a usability problem instance experienced by the participant that is not included in the master list, put an "A" (for add) in the Comparison column.

Appendix F Judge Materials

Appendix F.1 Judge Consent Form

Informed Consent for Participant of Investigative Project

Virginia Polytechnic Institute and State University
Department of Computer Science

Title of Project: Addressing Usability Engineering Process Effectiveness with Tool Support

Role of Participant: Judge

Investigators:
Dr. Rex Hartson, Professor, Computer Science, Virginia Tech
Jonathan Howarth, Graduate Student, Computer Science, Virginia Tech

I. The Purpose of this Research

You are invited to participate in a research study of usability engineering tools. Specifically, you will be reviewing usability evaluation reports of Scholar and rating them with respect to a number of guidelines. There is one other participant in this study performing the same task as you.

II. Procedures

This study will be conducted in a place of your choosing. Jonathan Howarth will begin by asking you to review some usability evaluation reports. He will then have you systematically review a number of reports and fill out a form for each. You can take up to a week to review the reports. Your role in this study is that of a reviewer of the reports. Jonathan Howarth is not evaluating you or your performance in any way; you are helping him to evaluate the reports. All information that you help him attain will remain anonymous. He may ask you questions while you are reviewing the reports. The total time commitment is estimated to be five hours.

III. Risks

There are no more than minimal risks associated with this study.

IV. Benefits of this Project

Your participation in this study will provide information that may be used to improve usability engineering tools. No promise or guarantee of benefits has been made to encourage you to participate. If you would like to receive a synopsis or summary of this research when it is completed, please notify Jonathan Howarth.

V. Extent of Anonymity and Confidentially

The results of this study will be kept strictly confidential. At no time will the results of the study be released to anyone other than individuals working on the project without your written consent. It is possible, however, that the Institutional Review Board (IRB) may view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research. The information you provide will have your name removed and only a participant number will identify you during analyses and any written reports of the research. The only individual that will have access to your name and participant number is Jonathan Howarth. He will destroy any identifying information within three years of completion of the study.

VI. Compensation

You will not receive compensation for your participation in the study.

VII. Freedom to Withdraw

You are free to withdraw from this study at any time for any reason without penalty. You may also choose not to complete any part of the study, such as individual questions on a questionnaire, without penalty.

IX. Participant's Responsibilities

I voluntarily agree to participate in this study. I have the following responsibilities:

- To notify Jonathan Howarth at any time about a desire to discontinue participation.
- After completion of this study, I will not discuss my experiences with any other individual for a period of two months. This will ensure that everyone will begin the study with the same level of knowledge and expectations.

X. Participant's Permission

I have read and understand this informed consent form and the conditions of this study. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty.

______________________________________ ______________________

Signature Date

Should I have any questions about this research or its conduct, I may contact:

Dr. Rex Hartson, Investigator, hartson@vt.edu, (540)231-4857

Jonathan Howarth, Investigator, jhowarth@vt.edu, (540)961-5231

In addition, if you have detailed questions regarding your rights as a participant in University research, you may contact the following individual:

Dr. David Moore, Institutional Review Board Chair, moored@vt.edu, (540) 231-4991

Appendix F.2 Judge Instructions

Overview

During this study, we will ask you to do the following:

- Task 1 - Meet as a group to discuss the guidelines and develop a basic understanding of what each means. We'll also go through some practice examples.
- Task 2 - Evaluate the lists of usability problem instances and usability reports produced by evaluators.

Task 1

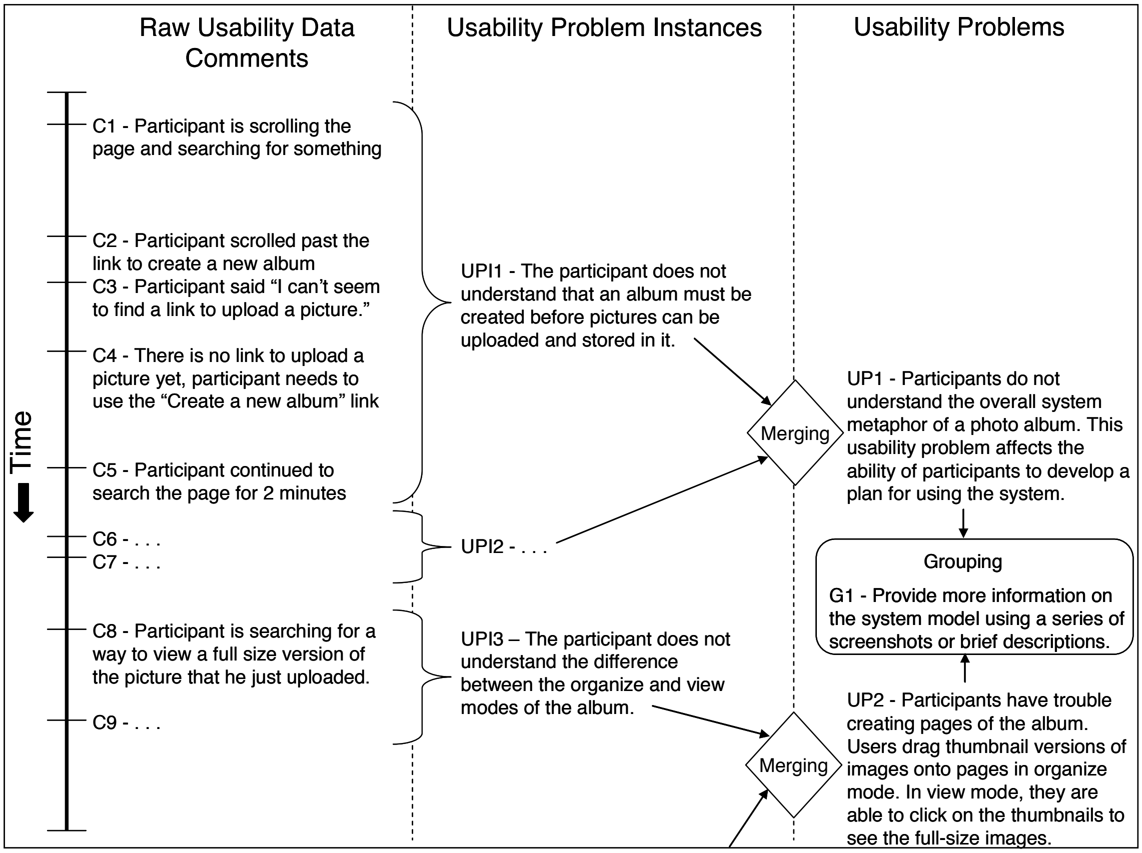
This task is intended to familiarize you with the process that you will be using to judge lists of usability problem instances. During this task, you will be asked to do the following:

- Familiarize yourself with the levels of usability problem data
- Familiarize yourself with Capra's guidelines
- Practice judging lists of usability problem instances

This task is expected to last two hours.

1.1 Levels of Usability Problem Data

A usability problem describes the effect that an interaction design flaw has on the user. Usability problems are documented with usability problem descriptions and represent analyzed usability problem information. The same usability problem may be experienced by multiple participants or multiple times by one participant. Each occurrence of a usability problem as encountered by a participant and observed by the evaluator is a usability problem instance. Usability problem instances are determined by analyzing the raw usability problem data produced by the facilitator during the collection stage. The following figure shows example usability data for a photo application.


Raw Usability Data Comments
Usability Problem Instances
Usability Problems
Time
C1 - Participant is scrolling the page and searching for something
C2 - Participant scrolled past the link to create a new album
C3 - Participant said “I can’t seem to find a link to upload a picture.”
C4 - There is no link to upload a picture yet, participant needs to use the “Create a new album” link
C5 - Participant continued to search the page for 2 minutes
C6 - . . .
C7 - . . .
C8 - Participant is searching for a way to view a full size version of the picture that he just uploaded.
C9 - . . .
UPI1 - The participant does not understand that an album must be created before pictures can be uploaded and stored in it.
UPI2 - . . .
UPI3 – The participant does not understand the difference between the organize and view modes of the album.
Merging
Merging
UP1 - Participants do not understand the overall system metaphor of a photo album. This usability problem affects the ability of participants to develop a plan for using the system.
Grouping
G1 - Provide more information on the system model using a series of screenshots or brief descriptions.
UP2 - Participants have trouble creating pages of the album. Users drag thumbnail versions of images onto pages in organize mode. In view mode, they are able to click on the thumbnails to see the full-size images.

1.2 Capra's Guidelines

For this study, you will use a modified version of the guidelines presented in [Capra, 2006].

Modified subset of Capra's guidelines for describing usability problems

1. **Be clear and precise while avoiding wordiness and jargon.**
    - Define terms that you use.
    - Be concrete, not vague.
    - Be practical, not theoretical.
    - Use descriptions that non-HCI people will appreciate.
    - Avoid so much detail that no one will want to read the description.
2. **Describe the impact and severity of the problem.**
    - Describe how it impacts the user's task.
    - Describe how often the problem will occur, and system components that are affected or involved.
3. **Support your findings with data.**
    - Include information on how many users experienced the problem and how often.
    - Include objective data, both quantitative and qualitative, such as the number of times a task was attempted or the time spent on the task.
    - Provide traceability of the problem to observed data.
4. **Describe the cause of the problem.**
    - Describe the main usability issue involved in the problem.
    - Avoid guessing about the problem cause or user's thoughts.
5. **Describe observed user actions.**
    - Include contextual information about the user and the task.
    - Include specific examples, such as the user's navigation flow through the system, user's subjective reactions, screen shots and task success/failure.
    - Mention whether the problem was user-reported or experimenter observed.
6. **Describe a solution to the problem.**
    - Provide alternatives and tradeoffs.
    - Be specific enough to be helpful without dictating a solution.
    - Supplement with usability design principles.

Capra, M. (2006). *Usability Problem Description and the Evaluator Effect in Usability Testing.* Unpublished disseration. Blacksburg, VA: Virginia Tech.

1.3 Practice

During this practice session, you will be asked to judge three sample lists of usability problem instances and then compare and discuss the results.

There is no time limit for this practice exercise. All files referenced are located in the “Practice” folder on the CD that we gave you.

Please read “Practice Sample Task Description.doc”. This document provides the background for the sample lists of usability problems that you will use in this practice exercise.

Please open “Practice Values.xls” and read the Introduction worksheet.

Please read the “Practice Sample 1.doc” and assign values for Capra’s guidelines on the worksheet titled “jts01”.

We will compare and discuss values.

Please read and assign values for “Practice Sample 2.doc” and “Practice Sample 3.doc”.

We will compare and discuss values.

Task 2

The purpose of this task is to judge lists of usability problem instances and usability reports produced by evaluators. You will be asked to do the following:

- Watch some videos to become familiar with Scholar and the steps for adding and removing students from courses.
- Assign values for Capra’s guidelines for each list of usability problem instances and for each usability report

This task is expected to take three to five hours. All files referenced are located in the “Study” folder on the CD that we gave you. Please email the investigator your file when you are finished.

2.1 Familiarization with Scholar and Videos of Correct Ways to Perform Tasks in Scholar

To familiarize yourself with Scholar and the correct way for adding and removing students, please watch the following videos:

- “Scholar Introduction.wmv”
- “Scholar Correct Add Student.wmv”
    - The text for the task reads: A student emailed you to ask your permission to force add the course. Add him to the course. His pid is psd_student_1.
- “Scholar Correct Remove Student.wmv”

  - o The text for the task reads: On the first day of class, you realized that John Dewey has not taken the necessary prerequisites and is not eligible for the course. Remove him from the course.

You can refer back to these videos at any time during the study and rewind, pause, and fast forward them as much as you like.

2.2 Videos of the Actual Users and Creation of Lists of Usability Problem Instances

Please watch the following three videos:

- “s27 Task 2.wmv” (add a student)
- “s67 Task 2.wmv” (add a student)
- “s27 Task 7.wmv” (remove a student)

You can refer back to these videos at any time during the study and rewind, pause, and fast forward them as much as you like.

2.3 Usability Problem Instances

Please open the “Instances Values.xls” spreadsheet. The Introduction worksheet explains how to assign values for Capra’s guidelines and in what order you should review the lists of usability problem instances produced by evaluators. The lists are in the “Instances” folder. Capra’s guidelines are listed in “Capras Guidelines.doc”

2.4 Usability Reports

Please open the “Reports Values.xls” spreadsheet. The Introduction worksheet explains how to assign values for Capra’s guidelines and in what order you should review the usability reports produced by evaluators. The reports are in the “Reports” folder. Capra’s guidelines are listed in “Capras Guidelines.doc”

2.5 Additional Notes

2.5.1 Concerning Unfinished Evaluator Documents

All evaluators had 1.5 hours to watch the videos of the participants performing tasks and create either a list of usability problem instances or a usability evaluation report. Some evaluators who produced lists of usability problem instances did not finish; the following text appears at the top of their documents: “* The evaluator ran out of time and did not finish the report. *”. In the body of the document, you will also see the following text before the instances that the evaluators did not finish documenting: “* The evaluator did not finish documenting the following usability problem instances. *”. The fact that some instance descriptions are not complete should not lower your ratings for the guidelines. Apply the guidelines to the completed instances and assume that the instances that are not complete would have been completed in a similar manner if the evaluator had had more time.

2.5.2 Formats

The evaluators were asked to use different formats when creating their lists of UP instances and their reports. A format does not imply the use of a specific tool or process. In other words, lists of UP instances or reports with the same format may have been produced by different tools and/or different processes.

2.5.3 Ordering of Tasks

The order in which the tasks appear in the reports is not important. The tasks will appear in one of the following two orders:

1. s27 – Add a student, s67 – Add a student, s27 – Remove a student
2. s27 – Add a student, s27 – Remove a student, s67 – Add a student

Appendix F.3 Judge Practice Sample Task Description

Practice Sample Task Description

This is an excerpt from the document that was given to the individuals who produced the lists of usability problem descriptions that you will be using in this practice exercise. This excerpt was taken from [Capra, 2006], but it was originally part of research documented in [Long et al., 2005].

Background of the Usability Evaluation

You have been asked to do a small usability evaluation of the Internet Movie Database (IMDb; imdb.com). You have had four participants do the following task:

> Task: Name all the movies that both Owen Wilson and Luke Wilson (the actor from Old School) have appeared in together.
>
> Answer: The Wendell Baker Story, Rushmore, The Royal Tenenbaums, Bottle Rocket, and Around the World in 80 Days

User profile: The IMDB has a broad range of users. The site has both occasional visitors and two types of frequent visitors – those who do only basic tasks (such as looking up an actress or movie), and those who do more complex tasks. Visitors may be general movie watchers or movie enthusiasts, independent of their level of experience with the IMDb website.

The ultimate goal of your assessment is to develop a set of improvements to the interface.

Report Goals and Audience

Your current goal is to generate a list of usability problem descriptions that summarizes the data collected during the study both from user interactions and from expert observations during the test.

The people reading the descriptions (e.g. usability practitioners, developers, marketing, product development, management, clients) may or may not have usability training. They may use this list for many purposes, such as deciding which problems to fix in the next product release, designing interface changes, or adding features to the software.

Evaluation Instructions

Please watch the movie of the usability session and comment on the IMDb user interface.

- You may watch the movie as many times as you like.
- Use the provided report template for your comments (UsabilityReport.[doclrtf])
- Your report should focus on the search feature tested in this task.
- Provide as many comments as you feel are appropriate.

For each comment that you write, please follow these guidelines.

- In the description, include as much detail as you would typically include in your own reports. If you put images in your own reports you may include them in this report.
- Report one usability problem or one positive feature per comment. Split comments that are conglomerates of several problems or positive features.

Capra, M. (2006). *Usability Problem Description and the Evaluator Effect in Usability Testing.* Unpublished disseration. Blacksburg, VA: Virginia Tech.

Long, K., Styles, L., Andre, T. S., & Malcolm, W. (2005). *Usefulness of nonverbal cues from participants in usability testing sessions.* Paper presented at the 11th International Conference on Human-Computer Interaction.

Appendix F.4 Judge Practice Samples

Practice Sample 1

Report Id: jts01

The organization / construction of the look up joint ventures page. Lots of problems with this page. The button at the top is not really necessary, since the user will have to scroll down the page to make the selections. It could be replaced with better instructions on how to use that page.

There could be default selections made - which the user could modify if they wanted. (eg. It could say “these are the movies in which X & Y have been credited together as actors. Click here to change the category in which they have been credited together”).

The Message Boards (and maybe the sponsored ads) could be moved below the search and email this page feature. This way there will be better grouping of features.

Very busy interface. Overload of features, causing users to pause and search for the feature they are trying to use.

The way the categories were divided on the look up joint ventures page. Using different colors for different categories (could alternate 2 colors) or some other distinct way of marking the categories would help reduce some confusion. Could even list the names followed by a list of the categories (eg. Owen Wilson as Actor I Writer I Director) in a selectable list (or a drop-down).

The way the search operators work. Two users expected X+Y to return results that would include the movies in which X&Y were credited together. Although not really an interface issue, the results page could try to guess what the user is trying to do and direct them to the correct feature.

Practice Sample 2

Report Id: jts02

The first participant and subsequent participants appeared to have either learned from the testing, or were familiar enough with the site to know that the search box they were looking for was at the bottom. Only one participant of the four was unsure where to go to find the search box - and for this person there was a significant delay in finding it. The placement of this search box should be evaluated closely, especially if this is a common task. If this is a common task, it is recommended that the search box be moved up within the content on the page.

After the participants entered an additional name in the “Find where [ ] Wilson is credited alongside another name” they were confused when they were presented with another page that had a button labeled “Look up joint ventures.”

On the “IMDb name search” the directions on the page were confusing to all of the participants. Three out of four participants pressed the “Look up joint ventures” button without choosing any checkboxes - one participant chose only one checkbox. If the positioning of the checkboxes was changed so that they were no longer below the button and the fold, but instead the button were below each group of checkboxes that may work better. Additionally, if the content were able to be moved further up the page to reduce the content below the fold that would improve the usability of the page.

After participants pressed the “Look up joint ventures” button, if they had not chosen enough people's names on the previous page they were given an error that stated “Need 2 or more names.” This statement does not give the user enough direction as to what they should do next. Changing this statement to say something such as “Select 2 or more names from the list to find joint ventures. Return to look up page.”

Most of the participants were not familiar with Luke Wilson and so were unsure if he was Luke Wilson (I) or Luke Wilson (II). Because of this, three out of four participants reviewed the Like Wilson (I) page to make sure they were selecting the correct actor. One user commented “Looks like him.” It is suggested that the site consider adding thumbnail images of the people on the site where they are listed so that users can scroll through the pictures and names to more quickly identify their goal.

All participants used the Search box in the upper left-hand corner. Two participants used the drop down to narrow their search and one participant used more advanced search techniques such as quotation marks and the symbol for inclusivity (+). The implementation of Search on this site is effective and useful to the participants.

It appeared that the cache was not cleared between testers because when participants typed in the names of the actors previous entries were shown.

The last participant chose the Writer's by accident and when she went back to select actors the writers stayed chosen. She did not notice this right away and had to be prompted to change it by the moderator.

One participant viewed the keywords which are organized by the number of occurrences. While this does seem to be an interesting bit of trivia, scanning the column of words and numbers is very time consuming. If the list were alphabetized or better yet, sortable by either # of instances or alphabetical it would be much easier to use.

Practice Sample 3

Report Id: jts03

Participant was not aware that they needed to reselect owen Wilson from the list of actors since the previous screen asked which actor to search in conjunction with Owen Wilson. Solve by having the website remember the user's previous selection when entering the “joint” search.

User clicked on hyperlink to select actor instead of checking the checkbox. Solve by left aligning all the checkboxes in a column and including instructional text asking user's to check the box below to select an actor.

Multiple data entries confuse user. Owen Wilson I vs. Owen Wilson II needs to be joined.

Provide a “back” button on the error page when no matches are found.

Perform Joint searches in the search tool by using the plus sign.

Search tool does not allow joint searches with logic operators.

Search functionality is not grouped together. User does not know where to look to do a combined search. Incorporate joint search functionality + UI into the search tool box.

Appendix F.5 Judge Rating Instructions and Spreadsheet

Introduction

In your role as a judge, you will use this spreadsheet to assign values for Capra's guidelines to evaluator's lists of usability problem instances. There are a number of worksheets in this spreadsheet; you can navigate among them using the tabs at the bottom of the window. Each worksheet contains a table that you can use to assign values for Capra's guidelines. The title of each worksheet is an evaluator's id number; to see the evaluator's list of usability problem instances, open the Word document that has the same name as the worksheet. Please assign values to the lists in the order that the worksheets appear in the tabs (begin with the leftmost tab and continue in order to the rightmost tab).

Assigning Values

Below is an example of the table that you will use to assign values for Capra's guidelines. The guidelines are listed in order vertically on the left. Values are listed horizontally at the top. Please place an "x" in the correct column for each guideline as demonstrated below.

	Strongly Disagree	Disagree	Somewhat Disagree	Somewhat Agree	Agree	Strongly Agree
Be clear and precise					x	
Describe the impact			x			
Support with data				x		
Describe the cause			x			
Describe observed actions		x				
Describe a solution						x

Appendix G Developer Materials

Appendix G.1 Developer Consent Form

Informed Consent for Participant of Investigative Project

Virginia Polytechnic Institute and State University
Department of Computer Science

Title of Project: Addressing Usability Engineering Process Effectiveness with Tool Support

Role of Participant: Developer

Investigators:
Dr. Rex Hartson, Professor, Computer Science, Virginia Tech
Jonathan Howarth, Graduate Student, Computer Science, Virginia Tech

I. The Purpose of this Research

You are invited to participate in a research study of usability engineering tools. Specifically, you will be reviewing usability evaluation reports of Scholar and expressing your opinions of them. There are three other participants in this study performing the same task as you.

II. Procedures

This study will be conducted in a place of your choosing. Jonathan Howarth will begin by asking you to review some usability evaluation reports. He will then have you systematically review a number of reports and fill out a form for each. You can take up to a week to review the reports. Your role in this study is that of a reviewer of the reports. Jonathan Howarth is not evaluating you or your performance in any way; you are helping him to evaluate the reports. All information that you help him attain will remain anonymous. He may ask you questions while you are reviewing the reports. The total time commitment is estimated to be three hours.

III. Risks

There are no more than minimal risks associated with this study.

IV. Benefits of this Project

Your participation in this study will provide information that may be used to improve usability engineering tools. No promise or guarantee of benefits has been made to encourage you to participate. If you would like to receive a synopsis or summary of this research when it is completed, please notify Jonathan Howarth.

V. Extent of Anonymity and Confidentially

The results of this study will be kept strictly confidential. At no time will the results of the study be released to anyone other than individuals working on the project without your written consent. It is possible, however, that the Institutional Review Board (IRB) may view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research. The information you provide will have your name removed and only a participant number will identify you during analyses and any written reports of the research. The only individual that will have access to your name and participant number is Jonathan Howarth. He will destroy any identifying information within three years of completion of the study.

VI. Compensation

In exchange for your participation and the participation of the other developers, Jonathan Howarth will perform a usability evaluation of Scholar.

VII. Freedom to Withdraw

You are free to withdraw from this study at any time for any reason without penalty. You may also choose not to complete any part of the study, such as individual questions on a questionnaire, without penalty.

IX. Participant's Responsibilities

I voluntarily agree to participate in this study. I have the following responsibilities:

- To notify Jonathan Howarth at any time about a desire to discontinue participation.
- After completion of this study, I will not discuss my experiences with any other individual for a period of two months. This will ensure that everyone will begin the study with the same level of knowledge and expectations.

X. Participant's Permission

I have read and understand this informed consent form and the conditions of this study. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty.

____________________________________ ____________________

Signature Date

Should I have any questions about this research or its conduct, I may contact:

Dr. Rex Hartson, Investigator, hartson@vt.edu, (540)231-4857

Jonathan Howarth, Investigator, jhowarth@vt.edu, (540)961-5231

In addition, if you have detailed questions regarding your rights as a participant in University research, you may contact the following individual:

Dr. David Moore, Institutional Review Board Chair, moored@vt.edu, (540) 231-4991

Appendix G.2 Developer Instructions

Overview

The purpose of this task is to assign values that reflect quality to usability reports produced by evaluators in an experiment. You will be asked to do the following:

- Watch the videos that evaluators watched before they created the usability evaluation reports
- For each usability report, assign quality assessment values in a spreadsheet titled "Reports Values.xls"

This task is expected to take three to five hours. All files referenced are located in the "Study" folder on the CD that we gave you. Please email the investigator your spreadsheet file when you are finished.

1. Videos Viewed by Evaluators to Familiarize Themselves with Scholar

Evaluators watched the following videos in the following order to familiarize themselves with Scholar and the correct way for adding and removing students:

- "Scholar Introduction.wmv"
- "Scholar Correct Add Student.wmv"
    - The text for the task reads: A student emailed you to ask your permission to force add the course. Add him to the course. His pid is psd_student_1.
- "Scholar Correct Remove Student.wmv"
    - The text for the task reads: On the first day of class, you realized that John Dewey has not taken the necessary prerequisites and is not eligible for the course. Remove him from the course.

Please watch the videos in the order specified above. You can refer back to these videos at any time during the study and rewind, pause, and fast forward them as much as you like.

2. Videos Viewed by Evaluators of Actual Users

Evaluators watch the following videos in the following order and then created usability evaluation reports based on the usability problem experienced by the users in these videos:

- "s27 Task 2.wmv" (add a student)
- "s67 Task 2.wmv" (add a student)
- "s27 Task 7.wmv" (remove a student)

Please watch the videos in the order specific above. You can refer back to these videos at any time during the study and rewind, pause, and fast forward them as much as you like.

3. Usability Reports

Please open the “Reports Values.xls” spreadsheet. The Introduction worksheet explains how to assign values and in what order you should review the usability reports produced by evaluators. The reports are in the “Reports” folder.

4. Additional Notes

4.1 Evaluators’ Familiarity with Scholar

Please keep in mind that the evaluators in the study only watched the videos listed above. They had not used Scholar before they participated in the study. So, for example, you should not expect the evaluators to comment on the usability of the wiki tool because it was not addressed in any of the videos.

4.2 Formats

The evaluators were asked to use different formats when creating their lists of UP instances and their reports. A format does not imply the use of a specific tool or process. In other words, lists of UP instances or reports with the same format may have been produced by different tools and/or different processes.

4.3 Ordering of Tasks

The order in which the tasks appear in the reports is not important. The tasks will appear in one of the following two orders:

3. s27 – Add a student, s67 – Add a student, s27 – Remove a student
4. s27 – Add a student, s27 – Remove a student, s67 – Add a student

Appendix G.3 Developer Rating Instructions and Spreadsheet

Introduction

In your role as a developer, you will use this spreadsheet to assign quality values to usability reports produced by evaluators in an experiment. There are a number of worksheets in this spreadsheet; you can navigate among them using the tabs at the bottom of the window. Each worksheet contains a table that you can use to assign values. The title of each worksheet is an evaluator's id number; to see the evaluator's report (in the "Reports" folder), open the Word document that has the same name as the worksheet. Please assign values to the lists in the order that the worksheets appear in the tabs (begin with the leftmost tab and continue in order to the rightmost tab).

Assigning Values

Below is an example of the table that you will use to assign quality values. Items are listed in order vertically on the left. Values are listed horizontally at the top. Please place an "x" in the correct column for each item as demonstrated below.

	Strongly Disagree	Disagree	Somewhat Disagree	Somewhat Agree	Agree	Strongly Agree
1. This usability evaluation report describes usability problems in a clear and precise manner and avoids jargon.				x		
2. This usability evaluation report describes how the usability problems impact the users.			x			
3. This usability evaluation report supports its claims with references to the problems of users in the video.			x			
4. This usability evaluation report describes causes of usability problems.						x
5. This usability evaluation report describes what the users were doing when they encountered usability problems.		x				
6. This usability evaluation report describes solutions to the usability problems that it documents.				x		
Additionally, there is an item, which asks you to rate the overall quality of the report. Please enter a quality rating value from 1 to 10 to the right of the item as demonstrated below.						
7. On a scale of 1 to 10, with 1 being the least useful and 10 being the most useful in terms of fixing usability problems in the target application, this usability evaluation report is a ___.	4					

Curriculum Vitae

Jonathan R. Howarth

Education

Virginia Tech Spring, 2007
Doctor of Philosophy, Computer Science and Applications

- GPA 4.0/4.0

Virginia Tech Spring, 2004
Master of Science, Computer Science and Applications

- GPA 4.0/4.0

Furman University Spring, 2002
Bachelor of Science, Computer Science
Bachelor of Science, German Literature

- GPA 3.98/4.0

Work Experience

Microsoft Summer, 2006
Program Manager, Lifecycle Team, Core Operating Systems Division

- Documented an existing build verification system
- Established criteria and a process for integrating new content into the build verification system
- Defined, focused, and drove the implementation of build verification tests for Windows Longhorn Server components

Air Force Office of Scientific Research (AFOSR) Grant Spring, 2004 to Fall, 2006
Research Assistant for Dr. Rex Hartson, Virginia Tech

- Worked with Pearson Knowledge Technologies to develop tools to support usability engineering processes
- Authored a 50,000+ LOC C# application that interacts with networked SQL Server 2000 databases and supports multiple concurrent users

National Science Foundation (NSF) Grant Summer, 2004
Research Assistant for Dr. Manuel Pérez-Quiñones, Virginia Tech

- Developed tools to help non-programmers create web applications
- Helped develop a web-based PHP application with a MySQL database

National Technology Alliance Grant Summer, 2003
Research Assistant for Dr. John Carroll and Dr. Mary Beth Rosson, Virginia Tech

- Worked on a team organized through Rossetex Technology Ventures Group to assess the National Imagery and Mapping Agency's collaboration capabilities and develop a roadmap for improvement

Virginia Tech Fall, 2002 to Fall, 2003 and Fall, 2006 to Spring, 2007
Teaching Assistant

- Graded and prepared material for two professionalism courses, one usability engineering course, one software engineering course, and one programming course

Liberty Insurance Services Summer, 2001

Intern

- Worked on an IBM mainframe to update JCL jobs, procedures, and libraries for use in Endevor, a source management system
- Used Brio to coordinate retrieval from SQL, Informix, and DB2 databases

Associations

Professional Memberships

- Usability Professional's Association (UPA) 2007
- Association of Computing Machinery (ACM) 2007
- ACM Special Interest Group on Computer-Human Interaction (SIGCHI) 2007

Honor Society Memberships

- Phi Eta Sigma (National Academic Honor Society)
- Upsilon Pi Epsilon (Computer Science Honor Society)

Activities

McBryde 102 Usability Lab Summer, 2004 to Fall, 2006

Administrator

- Set up and currently administer a usability lab based on TechSmith's Morae that is used by graduate students to run research studies

Computer Science Graduate Council Fall, 2003 to Spring, 2006

Secretary, Webmaster

- Represent the interests of graduate computer science students
- Established a new travel funding policy to help students pay for expenses to present published work at conferences
- Organized volunteers for a graduate student recruitment weekend

Publications

Selected Papers

- **J.R. Howarth**, P.S. Pyla, B. Yost, Y. Haciahmetoglu, D. Young, R. Ball, S. Lambros, P. Layne (2007). Designing a Conference for Women Entering Academe in the Sciences and Engineering. Accepted for publication in the Advancing Women in Leadership Journal.
- **J.R. Howarth**, (2006). Identifying Immediate Intention during Usability Evaluation. Proceedings of the ACM Southeast Conference, pages 274-279.
- P.S. Pyla, **J.R. Howarth**, C. Catanzaro, C. North, (2006). Vizability: A Tool for Usability Engineering Process Improvement through the Visualization of Usability Problem Data. Proceedings of the ACM Southeast Conference, pages 620-625.
- J. Rode, Y. Bhardwaj, M.B. Rosson, M. Perez-Quiñones, **J.R. Howarth** (2005). As Easy as 'Click': End-User Web Engineering. 5th International Conference on Web Engineering.
- **J.R. Howarth** (2004). An Infrastructure to Support Usability Problem Data Analysis. Unpublished Master's Thesis, Virginia Tech.