SUPPORTING HEURISTIC EVALUATION FOR THE WEB

A Dissertation

by

ANA ERENDIRA FLORES MENDOZA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2009

Major Subject: Computer Science

SUPPORTING HEURISTIC EVALUATION FOR THE WEB

A Dissertation

by

ANA ERENDIRA FLORES MENDOZA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	William Lively
Committee Members,	John J. Leggett
	Marshall Scott Poole
	Frank M. Shipman, III
Head of Department,	Valerie E. Taylor

August 2009

Major Subject: Computer Science

ABSTRACT

Supporting Heuristic Evaluation for the Web. (August 2009)

Ana Erendira Flores Mendoza, B.S., Universidad Autonoma Metropolitana;

M.S., New Mexico State University

Chair of Advisory Committee: Dr. William Lively

Web developers are confronted with evaluating the usability of Web interfaces. Automatic Web usability evaluation tools are available, but they are limited in the types of problems they can handle. Tool support for manual usability evaluation is needed. Accordingly, this research focuses on developing a tool for supporting manual processes in Heuristic Evaluation inspection.

The research was conveyed in three phases. First, an observational study was conducted in order to characterize the inspection process in Heuristic Evaluation. The videos of evaluators applying a Heuristic Evaluation on a non-interactive, paper-based Web interface were analyzed to dissect the inspection process. Second, based on the study, a tool for annotating Web interfaces when applying Heuristic Evaluations was developed. Finally, a survey is conducted to evaluate the tool and learn the role of annotations in inspection. Recommendations for improving the use of annotations in problem reporting are outlined. Overall, users were satisfied with the tool.

The goal of this research, designing and developing an inspection tool, is achieved.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

First of all, I want to thank my family for their endless support throughout this project. I could not have made it with out them.

"Domo domo domo arigato" to my friend Koji Ouchi for his help, support, and countless research discussions that helped shape this research.

My gratitude to my committee members for their support, but specially to Dr. Lively who exposed me to additional Software Engineering principles and practices, and for letting me pursue this research idea. To Dr. Shipman who helped me decipher the theory of "Reflection-in-Action [Schön 1983]" in the context of evaluation, reviewed the design of the first study in this dissertation, and pointed me in the right direction in qualitative data analysis. Thank you to Dr. Poole for his comments on groupware considerations for the design of this tool and help.

I would like to thank Michael Moore because he was always willing to answer my Human-Computer Interaction questions. I don't want to leave out those people who participated in my studies. Thank you for participating. Finally, my gratitude to all those people that I am not mentioning here individually, but that in one way or another contributed to this project.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

1   INTRODUCTION

As part of the Web development process, Web developers are confronted with evaluating the usability of Web interfaces (i.e. Web sites and applications). Typically, a combination of manual methods and automatic tools are used for an effective Web site evaluation –e.g. manual inspection is needed to supplement automatic validation tool results [Rowan et al. 2000]. However, Web projects are highly affected by their fast paced life cycles, leaving little room for full evaluations. Other major factors contributing to this situation are low budgeting assigned for testing and availability of usability experts.

Web developers need effective and cheap approaches to Web usability evaluation. Available automatic Web usability evaluation tools such as LIFT online and LIFT onsite [UsableNet 2002] and WebXACT [WatchFire 2007] have proven to be useful in finding syntactic problems. These include problems of consistency, verification of broken links, if pages contain links to the home page, alternative description of images (with use of the ALT tag in HTML), among others [Brajnik 2000]. Other problems of semantic and pragmatic nature are left out by automatic evaluation tools [Farenc et al. 1996], and need to be handled. Farenc and collaborators [Farenc et al. 1996] explored the limitations of automatic usability evaluation tools. In analyzing 230 rules for their ERGOVAL automatic usability evaluation tool for Windows systems they found that a maximum of 78% of the rules could be automated "whatever the

_____

This dissertation follows the style of *ACM Transactions on Information Systems.*

implemented methods are." The other 22% require input from humans to provide information and resolve semantic and pragmatic conflicts.

Usability problems that are not handled by automatic evaluation tools can be handled with semi-automatic and manual approaches. In semi-automatic approaches, the identification of usability problems start by the analysis of source files and completed with human intervention to provide information, make decisions or confirm problems. There are three manual methods that are typically used to find usability problems in user interfaces [Preece et al. 2002]: a) *usability testing* where testers observe users performing tasks and report usability problems based on their observations, b) with *questionnaires and interviews* users are asked about their experience in using a system, missing features, and overall satisfaction, among other matters, c) in *inspection methods* experts examine user interfaces and report usability problems based on their judgment and expertise.

There is an inspection method that appears frequently in the literature and which is widely used in academy and industry: Heuristic Evaluation. Heuristic Evaluation is an inspection method proposed by Nielsen and Molich [1990]. It follows the "discount" philosophy, in which simplified versions of traditional methods are employed (e.g. discount usability testing not requiring elaborate laboratory setups). It consists of having a small number of evaluators independently examine a user interface in search for usability problems. Evaluators, then, collaborate to aggregate all usability problems. During interface inspection evaluators use a set of usability principles as guide, known as "heuristics," to focus on common problem areas in user interfaces. An example of

such heuristics is "Help users recognize, diagnose, and recover from errors [Nielsen 2005b]." Interface features that violate the heuristics are reported as usability problems.

There have been just a couple of tools developed for assisting evaluators in Heuristic Evaluations. Problem aggregation has been supported [Cox 1998]. There was no intent for automating the aggregation process but rather supporting evaluators in manual processes in problem aggregation. These include identifying unique problems, discarding duplicates, and merging descriptions using the affinity diagrams [Snyder 2003]. There has been some effort in semi-automating problem identification in Heuristic Evaluation, but it is a formal, application-dependent approach. Loer and Harrison [2000] developed a system for querying a model checker for searching potential usability problems in user interfaces.

The need for supporting manual problem identification has led to the exploration of ways to support manual processes in Heuristic Evaluation inspection.

1.1   The Problem

The first idea of a tool for Heuristic Evaluation looked like a combination of a logging tool to keep track of usability problem, and a system that guides evaluators throughout the entire process from entering usability problems to generating problem reports. However, this was not enough. Other ways to support Heuristic Evaluation in inspection needed to be proposed. This was the challenge.

Cox [1998] studied the usability problem aggregation process in Heuristic Evaluation in depth and developed groupware based on his findings. Similarly, the Heuristic Evaluation inspection process was studied in depth and a tool was

development based on findings. Once there was a better understanding of the inspection process, the process was characterized, software tool requirements were identified, and a tool for inspection based on those requirements was developed.

The goal, objectives, and methodology of the research are stated in the following sections.

1.2   Research Goal and Objectives

The goal of this research is to develop a tool for supporting Heuristic Evaluation inspection.

Specific objectives to achieve this goal are:

1.   Characterize the inspection process in Heuristic Evaluation.

2.   Identify tool requirements.

3.   Build the tool.

4.   Evaluate the tool.

1.3   Methodology

The first step was to characterize the inspection process in Heuristic Evaluation to understand it better and come up with different ways to support it. A user study in the laboratory (Study 1) was conducted to understand how evaluators apply Heuristic Evaluation on Web interfaces. The output of this step is a rough characterization of the process and tool requirements.

Tool requirements were identified from the literature, Study 1 findings, and experience. Evaluators in Study 1 were found spending time in observing, annotating,

and navigating the interface, as well as elaborating usability problems. Tools for inspection are proposed based on these activities.

The research effort was focused on developing a tool for a specific activity in inspection: interface annotation. A tool called "HEAssistant" was developed for annotating Web interfaces when applying Heuristic Evaluation. Study 2 was designed to learn both the uses of annotations in inspection and the overall tool satisfaction.

1.4   Dissertation Overview

The remaining of this dissertation consists of the following sections.

In Section 2, related work is covered.

Section 3 describes Study 1 (inspection process dissection and characterization) in detail.

Section 4 includes tool requirements for Heuristic Evaluation support.

Section 5 describes the development of the tool, HEAssistant.

Section 6 describes Study 2; the evaluation of HEAssistant – Annotator.

In Section 7, this dissertation ends with conclusions.

2   RELATED WORK

This section discusses Heuristic Evaluation in detail. It supplements other major Heuristic Evaluation surveys [Cox 1998; Dykstra 1993; Woolrych 2001]. It focuses is on Heuristic Evaluation process and tool support, however.

The reader will find software requirements highlighted throughout the section. These are later referenced in Section 4 (Figure 2.1) when discussing software requirements for a Heuristic Evaluation suite.
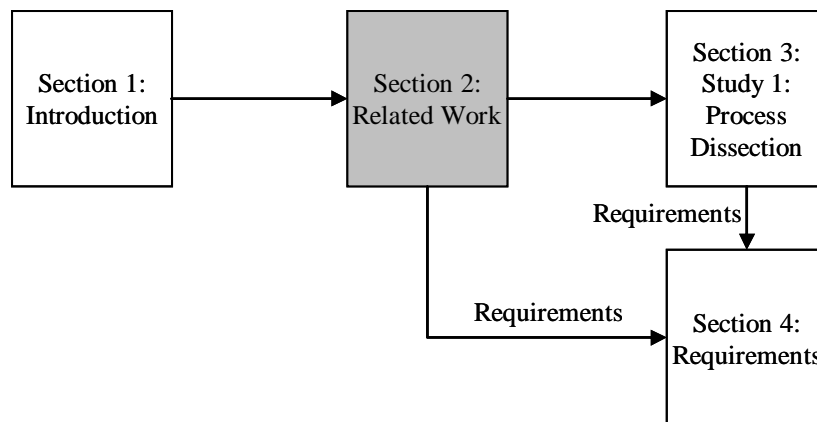
Figure 2.1. Section 2 Dissertation Context

2.1   Heuristic Evaluation

Heuristic Evaluation is an inspection method proposed by Nielsen and Molich [1990]. It is a simple method used to discover usability problems in user interfaces. It consists of having a small set of evaluators individually examine a user interface and judge for compliance with recognized usability principles called "heuristics". The lists of potential usability problems are aggregated in a single usability report. Members of the

development team are presented with the report to agree on the usability problem fixes and priorities. Figure 2.2 depicts the overall Heuristic Evaluation process.

Inspecting independently

Aggregating usability problems

Reporting to developers
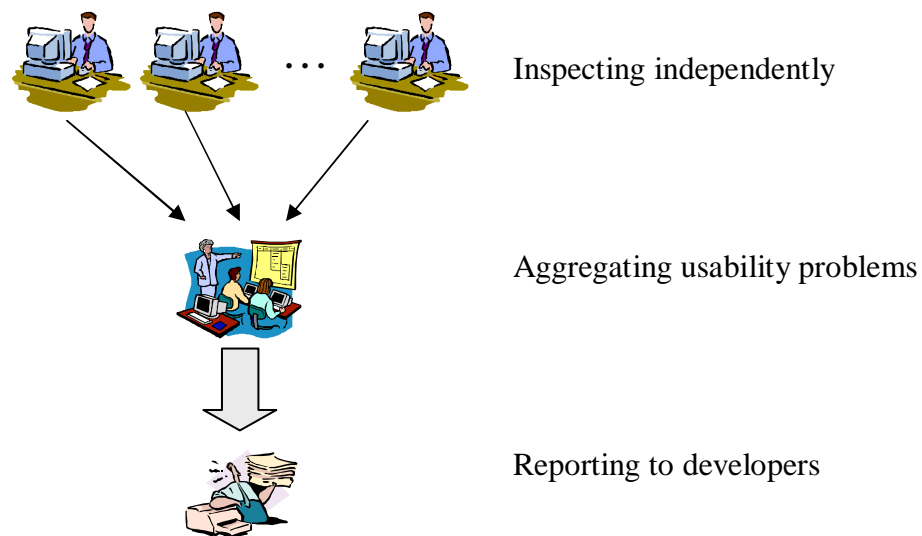
Figure 2.2. Heuristic Evaluation Overview

Nielsen makes recommendations to conduct a Heuristic Evaluation [Nielsen 2005a, 1994a]. A typical Heuristic Evaluation session lasts 2 hours. The evaluation can start with 2 passes of the user interface. A pass to get a general idea of the user interface design and overall interaction. Evaluators focus on particular parts in a second pass. Heuristics are meant to be used to help identify usability problems. With heuristics in mind evaluators carefully examine an interface and report interface features that were noticed to have violated them.

The output of a Heuristic Evaluation is a list of potential usability problems. Lists generated by all evaluators are aggregated. Evaluators meet and identify duplicates, combine problem descriptions, suggest solutions to problems and possibly rate their severity so they can be prioritized. Nielsen recommends using a 0-4 severity rating scale [Nielsen 1995b] (Table 2.1).

Table 2.1. Nielsen's Severity Rating Scale Borrowed from [Nielsen 1995b]

"0 = I don't agree that this is a usability problem at all" "1 = Cosmetic problem only: need not be fixed unless extra time is available on project" "2 = Minor usability problem: fixing this should be given low priority" "3 = Major usability problem: important to fix, so should be given high priority" "4 = Usability catastrophe: imperative to fix this before product can be released"

Several Heuristic Evaluation dimensions can be identified from the description above: the *heuristics* that are used to guide the inspection, *evaluators* performing the inspection, the *user interface* that is being evaluated, and the *process* that is followed. These are discussed immediately below.

2.1.1   Heuristic Evaluation Dimensions

2.1.1.1   Heuristics

Heuristics are general usability principles that "seem to describe common properties of usable interfaces [Nielsen 2005a]." Nielsen and Molich [1990] initially proposed nine heuristics, which were defined based on their experience of common problem areas in interfaces and consideration of guidelines. The results of a factor analysis of 249 usability problems [Nielsen 1994b] lead to 10 heuristics (Table 2.2). These are commonly used to evaluate interfaces in general. Instone [1997], for example,

explained Nielsen's 10 heuristics for the Web, emphasizing more on navigational aspects.

Table 2.2. Nielsen's Ten Usability Heuristics [Nielsen 1994b, 2005b]

1	Visibility of system status
2	Match between system and the real world
3	User control and freedom
4	Consistency and standards
5	Error prevention
6	Recognition rather than recall
7	Flexibility and efficiency of use
8	Aesthetic and minimalist design
9	Help users recognize, diagnose, and recover from errors
10	Help and documentation

Some alternatives have been proposed for specific domains to provide evaluators with domain knowledge they can use in evaluations. For instance, Dykstra [1993] developed calendar-specific heuristics based on results of user testing different commercial calendar systems. It was found that evaluators performed better when using calendar-specific heuristics. More usability problems were found by evaluators and more were severe than those performing a standard Heuristic Evaluation. Notice, however, that Dykstra's proposed heuristics had sub-headings. Dykstra's 9 heuristics had an average of 6.6 sub-headings describing a high-level heuristic, including a heuristic with 19 sub-headings. This may appear to be more like a Guideline Review with 60 guidelines than a Heuristic Evaluation with 9 high-level heuristics.

Nielsen recommends keeping the list short (about 10) for easy remembering [Nielsen and Molich 1990] (p. 249), although some may be added if they are domain-specific [Nielsen 2005a]. Muller et al. [1998] reformatted the list and added four more heuristics for his participatory approach to Heuristic Evaluation. In their approach they call for the participation of "work-domain experts" (users) to evaluate the targeted interface and added heuristics about human goals and experience.

The role of heuristics is not quite established. Heuristics are meant to help evaluators identify usability problems [Nielsen 2005a]. However, it is not clear that heuristics support the discovery and analysis of usability problems [Cockton and Woolrych 2001; Cockton et al. 2003]. In usability problem analysis, heuristics as analysis resource have not proven to be effective in *eliminating* false alarms and *confirming* actual usability problems [Cockton and Woolrych 2001].

Evaluators should not only report likes and dislikes, but they should explain problems with reference to violated heuristics or other usability principles or guidelines [Nielsen 2005a]. Cockton and Woolrych's [Cockton and Woolrych 2001] extended usability problem format (introduced in [Woolrych 2001]), for example, require evaluators to "hypothesise likely difficulties in context, rather than to just focus on problem features." The extended format encouraged evaluators to be more "reflective and less likely to propose problems with little justification [Cockton and Woolrych 2001] (p.175)." In fact, in an updated version of the form [Cockton et al. 2003] an entry for providing evidence of heuristic non-conformance was added, encouraging evaluators to reflect on their choose for violated heuristics.

Solutions to fix problems can be suggested based on violated heuristics [Nielsen 2005a] or some other taxonomy such as the User Action Framework [Andre et al. 2000] for classifying usability problems based on Norman's seven-stage theory of action [Norman 2002] (pp. 45-53).

Table 2.3 describes a tool requirement relevant to this discussion.

Table 2.3. Tool Requirement 2.1

Tool requirement 2.1:
Specify the heuristic checklist to be used.

Evaluators should be allowed to select the heuristic checklist to be used in a Heuristic Evaluation session. This being general, domain-specific, for the Web or for WIMP interfaces.

2.1.1.2   The Evaluator

Typically 5 [Nielsen 1992; Bevan et al. 2003] to 8 [Nielsen and Landauer 1993] evaluators are used in Heuristic Evaluation (although the number is still in debate [Bevan et al. 2003]).

Novice evaluators seem to perform poorly in Heuristic Evaluation [Nielsen 1992; Jeffries et al. 1991; Desurvire et al. 1992]. Evaluator performance is attributed in part to inexperience with usability and application domain arenas. Nielsen [1992] classifies evaluators as "novice," "regular specialists" (those with usability expertise), and "double specialists" (those with both usability and application domain expertise). In his study regular specialists found 75% of the problems when aggregating individual problem lists. To achieve the same success rate, it was required fourteen novice evaluators.

Users can become part of the evaluation force. Muller et al. [1998] incorporated users to take into account user's work-domain expertise in evaluations.

Table 2.4 summarizes evaluator expertise.

Table 2.4. Evaluator's Expertise

Expertise	Evaluator
Work-domain [Muller et al. 1998]	User
Software	Software engineer
Usability	Regular specialist [Nielsen 1992]
Usability and application domain	Double-expert [Nielsen 1992]

To balance the evaluator effect, modifications to Heuristic Evaluation method have been proposed. For instance, domain-specific checklists [Dykstra 1993] and tasks scenarios [Nielsen 1995a] (pp. 74-75) are aimed for non-experts in the application domain.

2.1.1.3   User Interfaces

The user interface format (paper vs. computer based) and interactivity (simulated or supported, see Table 2.5) may influence the way user interfaces are evaluated. Nielsen [1990] found that evaluating paper and computer mockups may influence the types of usability problems that are found. The author of this dissertation argues that "physical" characteristics of user interfaces have an effect on how they can be used and evaluated. When evaluating interactive interfaces, for example, evaluators interact with the interface, entering information, going from one screen to another, trying functionality, and so on. This at the same time enables evaluators to *experience* problems directly and, hence, providing a way for identifying problems.

Table 2.5. User Interface Types and Formats

Type	Interactivity	Format	Examples
Static	None	Paper	Screenshots and sketches with no interactivity
Interactive	Simulated	Paper	Paper prototypes [Snyder 2003]
		Computer	Wizard of Oz software-based prototypes [Kelley 1983]
	Supported	Computer	"Live" systems: computer prototypes and fully developed systems

Another aspect of user interfaces that may affect how interfaces are evaluated is its complexity. Slavkovic and Cross [1999] performed some initial studies on more elaborated and complex interfaces than those in the initial work of Heuristic Evaluation [Nielsen and Molich 1990]. Their results indicated that novice evaluators tend to focus on certain parts of the (Palm Pilot) user interface.

2.1.1.4   Usability Problem Formats

Evaluator's performance may be impacted by usability problem formats used to capture problem details in evaluation sessions. Cockton and collaborators [Cockton et al. 2003] designed an extended form and found unexpected improvement on evaluator's performance compared with a previous study [Woolrych 2001; Cockton and Woolrych 2001]. Results showed a 19% reduction on the number of false alarms and a 26% increase on appropriateness of heuristic application when using the extended form.

Heuristic Evaluation is known to produce not only a large number of problems [Jeffries et al. 1991; Bailey et al. 1992; Tan et al. 2009], but also a large number of false alarms [Bailey et al. 1992]. False alarms are identified problems that are not actual problems in the interface. A major risk of having a large number of false alarms is

making changes to an interface design based on them. Hence, we want to keep false alarms to a minimum.

Table 2.6 compares basic and extended usability problem formats. Notice that the extended format from [Cockton et al. 2003] goes beyond than reporting what was observed or experienced. It encourages evaluators to post-analyze the problem with entries such as likely/actual difficulties, evidence of heuristic non-conformance, and rationale for eliminating/confirming the problem.

Table 2.6. Various Usability Problem Formats

Type	Formats	Usability problem attributes
Basic	1. A simple format	• Problem description • Violated heuristic(s)
	2. Lavery and collaborators' form [Lavery et al. 1996]	• Problem description and justification • How was the problem found • Violated heuristic
Extended	3. Woolrych's form [Woolrych 2001; Cockton and Woolrych 2001]	• Problem description • Likely difficulties • Specific context • Assumed causes • Violated heuristic
	4. A form based on Cockton's and collaborator's form description [Cockton et al. 2003]	**Part 1. Problem** • Problem description • Likely/actual difficulties • Specific contexts • Assumed causes **Part 2. Discovery resources and methods** • Problem discovery description • Problem discovery specification: a) System scanning b) System searching c) Goal playing d) Method following • Confirmation rationale for probable problems

Table 2.6 continued

		Part 3. Heuristic(s) • Violated heuristic(s) • Evidence of conformance of heuristics **Part 4. Problem elimination** Rationale for eliminating the problem

Tables 2.7 and 2.8 describe tool requirements relevant to this discussion.

Table 2.7. Tool Requirement 2.2

> **Tool requirement 2.2:**
> Define new usability problem formats for describing usability problems
>
> Evaluators should be allowed to define new usability problem formats that can be used in Heuristic Evaluations for describing usability problems. Formats can be defined based on available formats.

Table 2.8. Tool Requirement 2.3

> **Tool requirement 2.3:**
> Specify the usability problem format for describing usability problems in a HEA project.
>
> Evaluators should be allowed to select the usability problem format for describing usability problems identified in a Heuristic Evaluation project.

2.2   Heuristic Evaluation Process

The Heuristic Evaluation process can be separated in three major phases: An inspection phase, in which evaluators independently evaluate the user interface; a preparation phase where evaluators independently prepare their list of identified problems for aggregation; and an aggregation phase, in which evaluators together

collaborate to generate a single report of usability problems. Figure 2.3 shows Heuristic

Evaluation phases and activities.
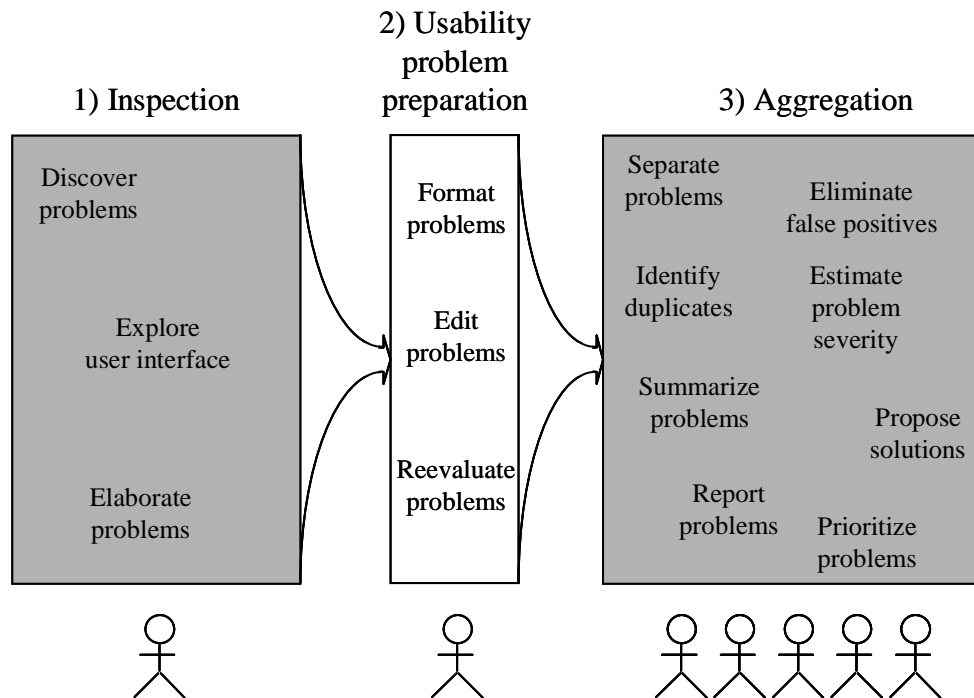
Figure 2.3. Heuristic Evaluation Phases

Table 2.9 describes a tool requirement relevant to this discussion.

Table 2.9. Tool Requirement 2.4

Tool requirement 2.4:
Fully support Heuristic Evaluation by providing capabilities for inspection, usability problem preparation, and usability problem aggregation.

There are 3 phases in Heuristic Evaluation: inspection, usability problem preparation, and problem aggregation. Fully support Heuristic Evaluation by assisting evaluators in all 3 phases.

2.2.1   Inspection Phase

Several activities can be depicted in this phase. Evaluators are involved in exploring the interface, identifying usability problems, and elaborating problems.

Exploring the Interface. Nielsen [2005a] recommends exploring the interface at least twice. A first pass is to get a general idea of the interface. A second pass is to analyze individual interface elements in context.

Exploration is dependent of the interface format. The format defines affordances (i.e. characteristics objects have that determine how they can be used [Norman 2002]) that allow particular ways of exploration. For example, several paper screenshots can be compared at once by positioning them side by side. Computer mockups [Nielsen 1990], on the other hand, allow exploring the interface via interaction and experiencing situations (e.g. feeling entrapped and not being able to exit to the "main system" [Nielsen 1990]).

Problem search influence how interfaces are explored. Cockton et al. [2003] introduced four (4) discovery methods: a) System Scanning: it consists in examining the interface without following any particular approach; b) System Searching: it involves some kind of strategy such as focusing in certain interface elements; c) Goal Playing: it consists in setting up goal and trying to achieve it; and d) Method Following: is similar to Goal Playing, but a step-by-step procedure is established and executed. These can be used in deciding how to approach problem search while illustrating different ways of exploration. Work needs to be done to look deeper into exploration patterns in terms of discovery methods.

Table 2.10 describes a tool requirement relevant to this discussion.

Table 2.10. Tool Requirement 2.5

Tool requirement 2.5:
Provide discovery resources [Cockton et al. 2004]

      Heuristic Evaluation as an inspection method that does not give much direction on how to go about the inspection. It is also characterized for not being task-based, as opposed to Cognitive Walkthrough [Wharton et al. 1994] in which evaluators have usage scenarios to guide the inspection. Goal playing scenarios [Cockton et al. 2003] and task scenarios [Nielsen 1995a] can be used to better guide evaluators when applying Heuristic Evaluation and provide task knowledge they can use.

Identifying Usability Problems. There are other factors than interface format and search strategies that may induce evaluators to notice potential problems. Inspection guidelines [Mack and Montaniz 1994; Zhang et al. 1999], for example, are intended to "stimulate inspectors to notice things about the software that might lead, on further reflection, to identifying a potential problem [Mack and Montaniz 1994]." Inspection guidelines give details of how to proceed, what to focus on, if post-meetings are needed, among others [Zhang et al. 1999]. Heuristic Evaluation is an informal method and few guidelines are given to evaluators.

Elaborating Usability Problems. "Once a potential problem is suspected, the inspector must develop the specifics of the problem description. [Mack and Montaniz 1994]". Evaluators can draw on different sources to elaborate problems [Mack and Montaniz 1994]: a) experiencing a problem directly, b) remembering having a similar problem, c) remembering others having a similar problem, d) simulating usage scenarios or exploring the interface further.

2.2.2   Usability Problem Preparation Phase

Evaluators may need to format, edit, and reevaluate usability problems before aggregation. Cox [1998] talks about formatting problems to facilitate aggregation of problem lists.

2.2.3   Aggregation Phase

Cox elaborated more on what problem aggregation is. It involves not only arranging, selecting, or categorizing identified raw problems [Cox 1998] (p. 3), but other activities. He renames problem aggregation as "results synthesis":

"Results synthesis is the process of transforming the entire collection of raw problem descriptions into a coherent, complete, and concise statement of the problems in the evaluated interface along as well as recommended actions to address the problems identified." [Cox 1998] (p. 139)

2.3   Tool Support

2.3.1   Inspection Method Tool Support

Little work has been done in supporting inspection methods in general. Ivory and Hearst [2001] performed and extensive survey of 75 approaches for evaluating WIMP (Windows, Icons, Menus, and Pointers) and 57 approaches for evaluating Web interfaces (29 of them applicable to both Web and WIMP). These include automated and non-automated approaches. Among the 9 inspection methods considered, only 2 have been automated in different aspects of the method. One of them, Cognitive Walkthrough, has been supported in usability data *capture* (1 approach surveyed). The other method,

Guideline Review, has been supported in user interface *analysis* (8 approaches) and *critiquing* (11 approaches).

Ivory and Hearst [2001] describe the capture, analysis, and critiquing automation types. Systems automating usability data capture automatically record necessary data to find usability problems in user interfaces. Analysis systems automatically identify potential usability problems in user interfaces. The highest level of automation is obtained with critiquing systems. Critiquing systems not only identify potential usability problems, but also make recommendations to improve user interfaces. Automated inspection method systems surveyed in [Ivory and Hearst 2001] follow.

Capture automation. There was an early effort to automate the capture process in a Cognitive Walkthrough. The "Automated Walkthrough" [Rieman et al. 1990, 1991] was a HyperCard system designed to reduce tedious paperwork and guide the evaluator through the question-oriented process. However, this initial effort still needed adjustments to reduce the tediousness and repetitiveness of work [Rieman et al. 1990].

Analysis automation. Eight (8) approaches to automate user interface analysis in Guideline Review were surveyed. Two of them use quantitative measures to aid in identifying potential usability problems in WIMP interfaces (e.g., semi-Automated Interface Designer and Evaluator (AIDE) [Sears 1995]). SHERLOCK [Mahajan and Shneiderman 1997] "analyzes terminology and consistency" conditions in user interface elements. In 3 other approaches they "analyze the structure of Web pages" (e.g. the Hypertext Authoring Tool (HyperAT) [Theng and Thimbleby 1998]). The Web Static Analyzer Tool (WebSAT) [Scholtz et al. 1998] uses guidelines to analyze Web sites. A

different approach is seen in Design Advisor [Faraday 2000], which "analyzes the scanning path of a Web page" based on eye tracking study results.

As part of her doctoral work, Ivory [2001] developed a tool called WebTango [Ivory and Hearst 2002]. It is a metric-based tool for evaluating Web sites.

Critiquing automation. Eleven (11) systems illustrate the critiquing approach to evaluate user interfaces with Guideline Review. Critiquing systems not only analyze the user interface in search for usability problems, but recommends possible solutions to problems and perhaps help correct them. Four approaches are used for evaluating WIMP interfaces (e.g. Knowledge-based Review of user Interface (KRI/AG) [Löwgren and Nordqvist 1992], and Ergoval [Farenc et al. 1996; Farenc and Palanque 1999]). SYNOP [Balbo 1995] is another critiquing system for WIMP, which also helps modifying the user interface to correct the identified problems. Two other systems "check HTML syntax" (e.g. Dr. Watson [Addy and Associates 2009]). In addition, 4 critiquing systems for the Web were surveyed (e.g. LIFT online [UsableNet 2002] and Bobby [CAST 2004]).

No support for Heuristic Evaluation was reported in [Ivory and Hearst 2001], but the next section discusses current Heuristic Evaluation support tools found elsewhere.

2.3.2   Heuristic Evaluation Support

A formal approach to supporting Heuristic Evaluation was identified. Loer and Harrison [2000] approach formally analyzes behavioral aspects of user interfaces based on encoded heuristics. The behavior of the system and its environment are specified using Ofan models [Degani 1996]. These consist of representations of the interface

(control elements, functionality, and output elements), environment properties, and user tasks. Evaluators query a model-checker in search for potential usability problems. The disadvantage of this approach is that heuristics should be specified on an application basis. In addition, using formal notations can be time-consuming and difficult to use.

Other researchers propose using defect tracking tools such as IBM Rational ClearQuest [IBM 2008] to enter usability problems directly to defect databases. There is a major problem with these types of tools, however. They lack support for inspection-specific tasks such as problem identification.

Cox [1998] on the other hand, focuses on supporting one task of Heuristic Evaluation: aggregate problems. He views it as a collaborative effort of not just merging sets of problems and eliminating duplicates, but of obtaining a consensus of what the usability problems are, and sharing their rationales among evaluators. Cox renames this task as "results synthesis." Groupware to support results synthesis is designed based on his observational studies.

3 STUDY 1: INSPECTION PROCESS DISSECTION

The need for finding ways to support inspection led to the execution of the study described in this section, Study 1. By dissecting the inspection process in Heuristic Evaluation and gaining a better understanding of it, a set of tools for inspection are proposed.

High-level tool requirements are derived and listed in the section. These are incorporated later in Section 4 (Figure 3.1).
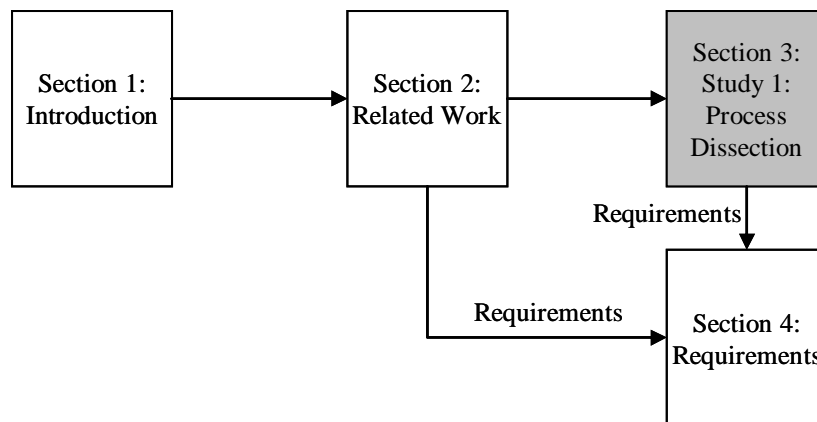
Figure 3.1. Section 3 Dissertation Context

3.1 Study Design

This is an in-depth observational study of Heuristic Evaluation applied to the Web. Evaluators are observed performing a Heuristic Evaluation on a paper-based, non-interactive Web interface in the laboratory. Recorded videos are analyzed to identify and

measure events. Main activities evaluators exhibit in action are derived and tool features are proposed to support those activities.

3.1.1 Subjects

A group of 7 people participated in the study: 4 Computer Science graduate students, 2 people with Human-Computer Interaction background – specifically, people who have taken 3 related courses, and a Web developer with 2 years of experience.

3.1.2 The Static Web Interface

The term "static" is used to emphasize that the interface is a paper-based Web interface with no simulated interactivity. This is different from paper prototyping [Snyder 2003] where a person plays "computer" and simulates interactivity by presenting screens based on user's actions.

The static Web interface consists of a set of printed screenshots and a storyboard created for navigational purposes. Six (6) Web pages from the Zen Cart [2006] Web site were selected, and printed in full (i.e. from top to bottom), color, and with a comparable width as they would appear on screen. Zen Cart is a customizable shopping cart package for e-commerce Web sites. It is an open source Web site that comes as an online store of hardware, software and DVD Movies. Zen Cart release version 1.3.6 was used.

The static Web interface is considered to be of low-fidelity. The fidelity of a prototype is defined by Virzi [1989] as "a measure of how authentic or realistic a prototype appears to the user when it is compared to the actual service." The static Web interface is far from being seen as the actual system as it is. It consists of only six (6) paper Web pages and a storyboard with no interactivity.

The static Web interface is formed by the following Web Pages (WP) and StoryBoard (SB). These are found in Appendix A.

- WP1. Home page

- WP2. "DVD Movies" category page

- WP3. "Speed 2: Cruise Control" (DVD movie) product page

- WP3.1. "Larger image" page: It appears when a link below the product image is clicked in WP3.

- WP3.2. "Shopping Cart" page: This page is shown after adding "Speed 2: Cruise Control" movie to the shopping cart in WP3.

- WP3.3. "Sign in" page: It appears when wanting to write a product review in WP3.

- SB. Storyboard: It depicts the sequence in which Web pages are presented.

3.1.3 Procedures

The study consisted of two phases (Figure 3.2). Phase 1 was a training phase. Evaluators got familiar with Heuristic Evaluation and practice applying the method on a given Web site. Training took place online and unsupervised. Phase 2 was the core part of the study. User testing was conducted. Evaluators met at the laboratory. They were asked to perform a Heuristic Evaluation on the static Web interface for 20 minutes. After the evaluation session they were interviewed about their approaches to finding problems for 35 minutes. The interview session was recorded.

```
┌─────────────────────┐        ┌─────────────────────┐
│      Phase 1:       │        │      Phase 2:       │
│       Online        │───────▶│    User Testing     │
│ Heuristic Evaluation│        │                     │
│      Training       │        │                     │
└─────────────────────┘        └─────────────────────┘
```
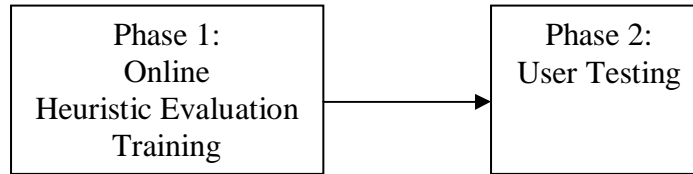
Figure 3.2. Study 1 Part 1 Overview

A Web site was developed to guide evaluators through Phase 1. Evaluators were asked to complete a background questionnaire, download training materials, and practice applying the method. They practiced a Heuristic Evaluation on the Gutenberg Web site [Project Gutenberg 2009] for 40 minutes. A Heuristic Evaluation booklet was among the training materials. It describes the method, how to conduct it, and a usability problem form to report problems. Nielsen's [Nielsen 2005b] ten usability heuristics and Wood's [Wood 2004] explanation of the heuristics for Web interfaces were provided. Appendix A includes the background questionnaire, Heuristic Evaluation booklet, and instructions given to evaluators in the exercise.

Evaluators spent 20 minutes in applying Heuristic Evaluation on the given Web interface in Phase 2. The evaluation session was videotaped. A video camera (the "Back" video camera) was located on the back and side of evaluators to capture evaluators' actions. These include writing on notebook, looking at Web pages, and moving materials around. Figure 3.3 shows the study setting.

We video recorded the writing area separately to look into writing events closer if necessary. The video camera covered only the evaluator's hand and a notebook (see

Figure 3.4). They were asked to keep the notebook within the marked area to capture

writing events.
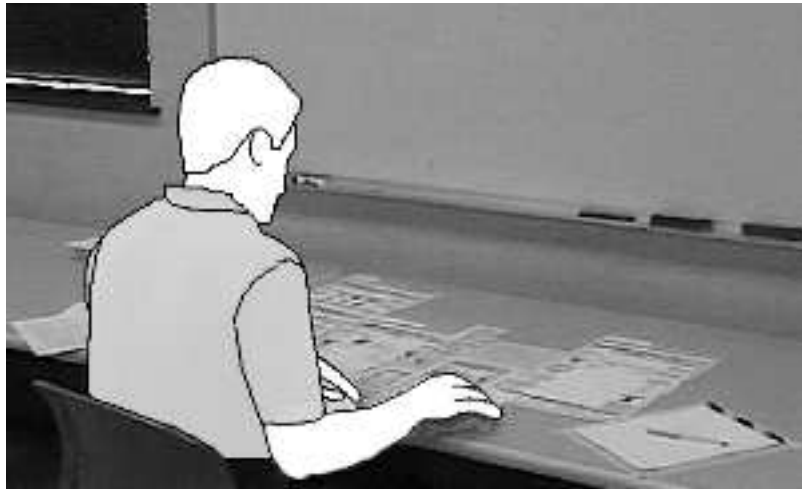
Figure 3.3. Study Setting
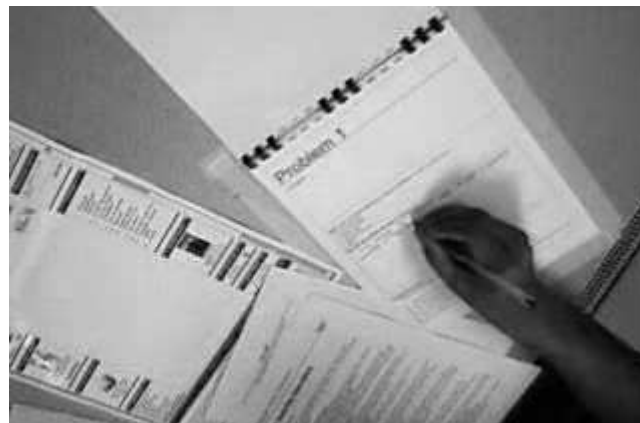
Figure 3.4. Writing Setting

The notebook was prepared to help evaluators describe problems and learn their rationales. It was made using the same format used in training. The format was a short version of Cockton et al.'s [Cockton et al. 2003] format. The problem attributes include a brief problem description, discovery method (i.e. System Scanning, System Searching, Goal Playing, and Method Following [Cockton et al. 2003]) specification and description, specific steps taken in finding the problem, and reference to violated heuristic(s).

Figure 3.5 shows the initial material arrangement. Supplementary materials were found on the left. These consisted of the same Heuristic Evaluation booklet that was used in training, and Nielsen's [Nielsen 2005b] heuristic checklist. The interface materials were found in the middle area. The printed Web pages were arranged based on the storyboard. The notebook was far on the right to facilitate distinguishing when evaluators started writing. It required them to slide the chair (with wheels) to reach the writing area.

In the evaluation session, evaluators were instructed to focus on the "Speed 2: Cruise Control" product page. However, they could visit other Web pages if needed and report problems there. The Web site was described as being under construction. There was no maximum number of usability problems to report. The facilitator was not present in the same room, but was available for questions. Instructions were explained to evaluators and given in written form (see Appendix A).
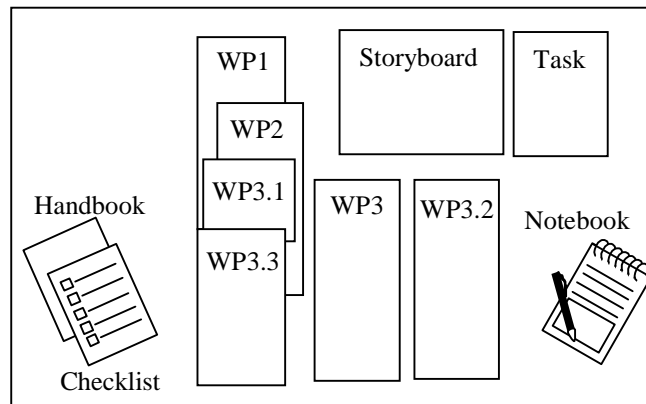
Figure 3.5. Initial Material Arrangement.

Supplementary materials: Heuristic Evaluation booklet, Nielsen's [Nielsen 2005b] heuristic checklist; Interface materials: Home page (WP1), Category page (WP2), Product page (WP3), "Larger image" page (WP3.1), "Shopping Cart" (WP3.2), "Sign-in" page (WP3.3), Storyboard; Notebook; Task description

3.1.4   Time Analysis

A time analysis was performed to look deeper into the Heuristic Evaluation process. This was through an in-depth video analysis where a set of activities were identified and measured. These activities give indications of the type of features tools may have to support evaluators in inspection.

A significant amount of effort was invested in post analyzing Back videotapes. Videos were reviewed at least twice. Timings were extracted to learn when events occurred. Events were coded, and total timings were calculated. At the end, a timeline of events was produced, so going back to certain points in videos was possible when necessary.

The steps in the time analysis are described below:

1. Describe and record actions. Evaluator actions were briefly described aloud while watching Back videos. A Sony ICD-P320 IC recorder was used to audio record sessions. Some actions were missed when the analyst could not keep up with actions being watched. This was fine as the goal was to generate a preliminary list of actions.

2. Transcribe preliminary list of actions. Audio recordings were listened and actions were written on paper.

3. Confirm and measure events. Back videos were watched at least a second time to confirm events and extract their approximate start and end timestamps. Videos were paused and rewind frequently. The list of preliminary actions was used to "look ahead" in videos and note groups of actions that could be simplified as single events.

4. Code events. Data coding [Kuniavsky 2003] involved classifying events based on selected categories. A set of categories were proposed initially, but refined later in the analysis when new actions challenged category definitions. Documenting categories were of help to be consistent throughout the analysis. However, this was not done until mid-analysis when too many details were difficult to remember.

5. Calculate total timings. Total timings were calculated for each event category. The categories represent the different activities observed in the process.

3.2   Characterized Static Inspection Process

After analyzing all videos, a rough characterization of the Static inspection process was obtained. Figure 3.6 shows the data aggregated from all 7 evaluators. Several activities were found in the Static process: a) observe the interface, b) quickly visit a Web page, c) elaborate problems and revisit materials, d) navigate the interface, and e) annotate the interface. These are described in the following sections.
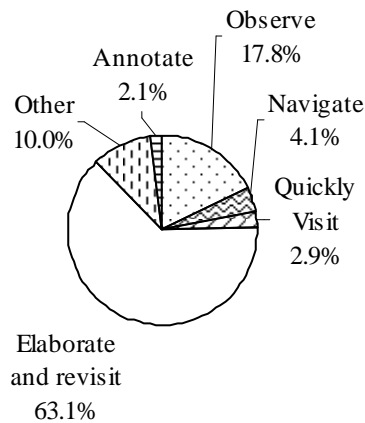
Figure 3.6. A Rough Characterization of the Static Inspection Process

3.2.1   Observing and Quickly Visiting the Interface

An "observe" event is defined as the time spent in carefully examining a screenshot before starting to write. It was found that evaluators visited the interface before and after starting to write on the notebook. We distinguish between these two cases because focus of evaluators changes when writing starts. It is argued that once evaluators start writing, focus changes from searching for usability problems and

gathering details about them to elaborating problems. At the action level evaluators diverge from observing, quickly visiting, navigating, and annotating interface to start writing.

There was a three-second threshold established to distinguish between "observe" and "quickly visit" events. When evaluators look a screenshot for less than 3 seconds the event was considered a "quickly visit" event. A one- or two-second event was too short to be a careful interface examination.

There were limitations to the study due to the fact we relied solely on the Back video camera to determine events. An eye-tracking technique would have aided resolving some conflicts that arose in video analysis. Conflicts appeared when evaluators slightly turned their head, and there were several materials that were potential targets of focus. There were cases where two or more materials were overlapping or too close to each other. This made it difficult or impossible to determine the target of focus and, hence, the type of event. A conservative position was taken when resolving conflicts.

Experience in evaluating static interfaces has been that evaluators spend a significant amount of time observing screenshots when searching for usability problems. Hence, the criterion to define events was to avoid overestimating such a time. Conflicts occurred when deciding whether the evaluator had turned his or her head enough to consider it a new event. The initiation of a new event was considered over continuing one. Other conflicts occurred when the target of focus was unclear. Non-interface materials were chosen over interface materials. In this way total observation time was a lower bound of the "actual" observation time.

A significant amount of time was spent in observing the interface. About 17% of the aggregated time was time was spent in this activity, which is the second highest time observed. This is an indication of the importance observation has in Static evaluation.

Table 3.1 highlights a tool requirement relevant to this discussion.

Table 3.1. Tool Requirement 3.1

| **Tool requirement 3.1:** |
| Provide observation tools to improve evaluator's observation skills. |

3.2.2 *Elaborating (Problems) and Revisiting (Interface and Materials)*

Evaluators spent a significant amount of time in elaborating problems and revisiting interface and supplementary materials (63.1% of the aggregated time). These activities are tightly coupled. They appear interchangeably until evaluators initiate inspection again.

In elaborating problems through usability problem forms evaluators may visit materials to recall or gather information relevant to problems. For instance, the usability problem format used in this study required evaluators to describe steps involved in discovering problems. In such cases, evaluators might visit interface materials to check involved interface elements (e.g. visited links) or details (e.g. page layout). When referencing heuristics, evaluators might visit the heuristic checklist to recall heuristics. There is more work needed to investigate reasons for revisiting materials when elaborating problems.

Table 3.2 describes a tool requirement relevant to this discussion.

Table 3.2. Tool Requirement 3.2

Tool requirement 3.2: Provide tools to capture usability problems   Once usability problems are identified they are elaborated. This can be done immediately after or after finding several other problems.

3.2.3 *Navigating the Interface*

Navigating the interface involves passing from an interface material (i.e. a Web page or storyboard) to another, rearranging them, and visiting the storyboard. Time spent in the storyboard is by definition navigation time. The storyboard has page hierarchy and linkage information that can be used to navigate through printed Web pages.

The interface format affects how evaluators move through dialogues. When an interactive interface is being evaluated, or used, the order in which dialogues are viewed is determined by the system based on evaluator's actions. In a Static interface, this depends on the evaluator him/herself.

In case evaluators try to navigate through a scenario of usage it will be limited on the dialogues that are available for evaluation. These are selected and instantiated before an evaluation session. In this study only six Web pages were available. Evaluators could navigate from the home page to the category page to the product page. From the product page there was only one level down the page hierarchy, limiting how far they could go.

In the study, 4.1% of the aggregated time was spent on navigating the interface. This is relatively low compared to the time spent in observing (i.e. 17.8%). This gives us an idea of the dynamics (or lack of) of Static evaluation. Notice, however, evaluators were instructed to focus on evaluating the product page. Therefore, a different time distribution could be obtained in a more free-form inspection.

3.2.4 Annotating the Interface

Two evaluators annotated the product page. There were 15 annotations, including 4 single markings and the rest were compound formed of several makings. Note that not all evaluators were instructed about the possibility of writing on Web pages. This was an author's slip. Therefore, more annotation instances could have occurred.

Evaluators made use of lines, shapes, text, arrows, and question marks to add different kinds of annotations to the printed Web page. Interface elements or areas were connected through arrows or simple lines. Areas were marked and labeled. Questions were attached to marked areas (e.g. "Is this important", "why", "Is this location best"). Table 3.3 shows selected annotations made by evaluators.

Table 3.3. Annotation Examples
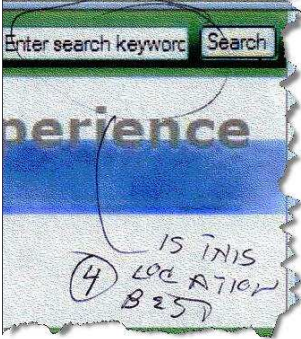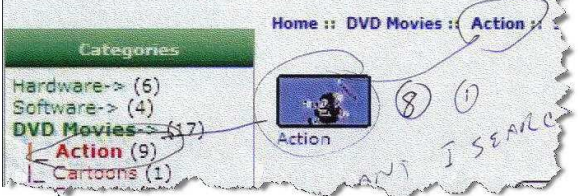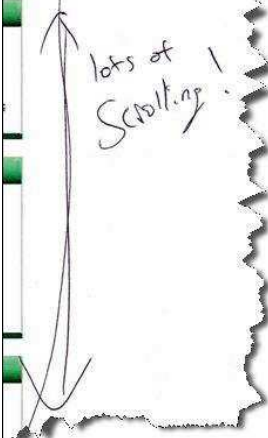
Annotation	Single/Compound	Markings
	Compound	Circle, comment/label, number, connector
	Single	Question marks
	Single	Label
	Compound	Connectors, circles, numbers
	Compound	Arrows, comment/label

Table 3.4 highlights a tool requirement relevant to this discussion.

Table 3.4. Tool Requirement 3.3

| **Tool requirement 3.3:** |
| Provide annotation tools to support evaluators in inspection |

3.3   Implications for Tool Support

To fully support Heuristic Evaluation, we need to develop tools for inspection, usability problem preparation, and problem aggregation. Tools for problem aggregation have been proposed elsewhere [Cox 1998], and can be adopted and enhanced. Tool features for problem preparation can also be proposed based on [Cox 1998] and results from this research.

The inspection process when evaluating Static Web interfaces has been dissected in this study. Identifying the types of activities evaluators perform in inspection give us insights of the types of tools that can be developed to support inspection. Tools for inspection are proposed below.

3.3.1   Annotation Tools

Annotation has been studied in contexts such as in reading [Marshall and Bursh 2004; Marshall 1997] and video viewing [Chaudhary 2008]. It is proposed in this research for inspection. It is of interest to learn the role of annotations in problem discovery and the types of problems that are identified when annotation is used.

Table 3.5-3.9 list requirements for designing tools for annotation of Web interfaces in inspection. These are derived from annotation examples in Table 3.3 and from experience.

Table 3.5. Tool Requirement 3.4.

Tool requirement 3.4:
Provide ways to highlight/mark interface elements and areas

Evaluators should be able to mark interface elements/areas in the interface, not necessarily by circling items.

Table 3.6. Tool Requirement 3.5.

Tool requirement 3.5:
Provide ways to attach comments to highlighted items or annotations.

Table 3.7. Tool Requirement 3.6.

Tool requirement 3.6:
Provide a way to apply heuristics to interface items or annotations.

Table 3.8. Tool Requirement 3.7.

Tool requirement 3.7:
Annotation type set

Evaluators should be able to annotate with different types of annotations. These may include:
- Question mark(s)
- Notes
- Circles
- Connectors
- Ranges (e.g. $\leftarrow \rightarrow$)
- Labels

Table 3.9. Tool Requirement 3.8.

Tool requirement 3.8:
Provide ways to connect/associate interface items and annotations

3.3.2 *Observation Tools*

Two tools are proposed to improve evaluator's observation skills: Magnifiers and Window views.

3.3.2.1 Magnifiers

Magnifiers follow a similar approach as Fisheye views [Furnas 1999], which model fisheye lenses to visualize large amounts of information in a small area. Magnifiers, however, model magnifying glasses (Figure 3.7). They are intended to not only magnify the focused area, but also to provide contextualized information about interface items. Feedback should go beyond visual aspects. Web page metrics such as those in [Ivory 2001] can be used to provide relevant Web statistics. The goal is to expose interface aspects that are not easily visible to the evaluator eye, which can lead to the identification of usability problems in user interfaces.

Figure 3.7. Magnifiers for Inspection

3.3.2.2 Window Views

The Window view (Figure 3.8) will provide similar feedback as the Magnifiers, but the appearance will be different. The focus area is a see-through area (no magnification). The negative space is shaded in a translucent tone. In this way the evaluator can see the rest of the interface behind and at the same time highlight the focus area.
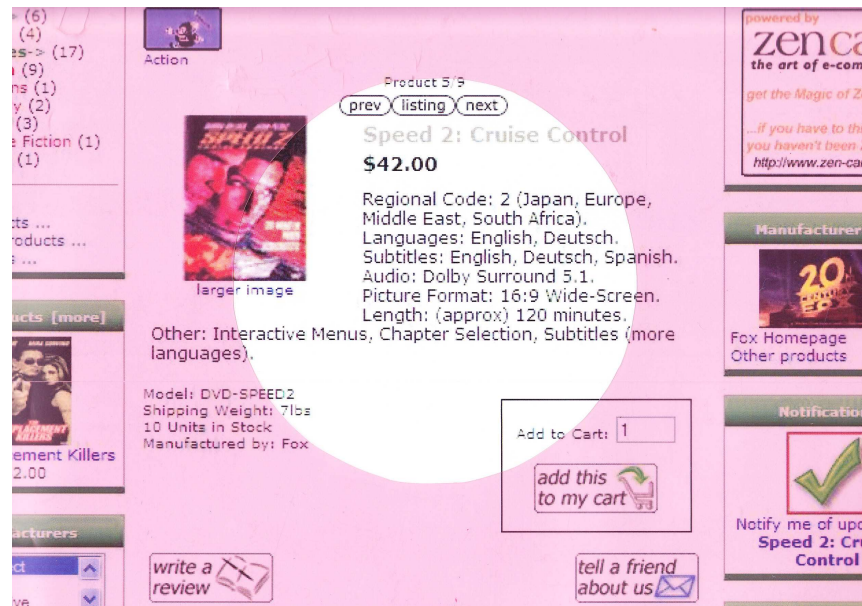
Figure 3.8 Window View for Inspection

3.3.3 Elaboration Tools

Requirements for developing problem elaboration tools are listed in Section 4.

3.3.4 Navigation Tools

Browsers already support navigation capabilities; however, if annotation, observation, and usability problem elaboration tools are being proposed, one may wonder if there is the need for navigation capabilities especially designed for inspection.

3.4 Conclusions

The Heuristic Evaluation inspection process has been dissected at a certain degree through an in-depth observational study. Evaluators were observed in video performing a Heuristic Evaluation on a non-interactive, paper-based Web interface.

Evaluators were found elaborating usability problems, observing, navigating, and annotating the interface. Tools are proposed based on these activities.

Observation and annotation tools are the most prominent tools that are proposed. Magnifiers and "Window views" are proposed to improve evaluator's observation skills.

Software requirements for developing Heuristic Evaluation tools are highlighted in the section and are incorporated in Section 4.

4   REQUIREMENTS

This section includes software requirements for developing a Heuristic Evaluation suite (HEAssistant Suite) consisting of a collection of tools supporting Heuristic Evaluation.

Requirements highlighted in Section 2 are included, as well as those derived from Study 1 findings in Section 3 (Figure 4.1).
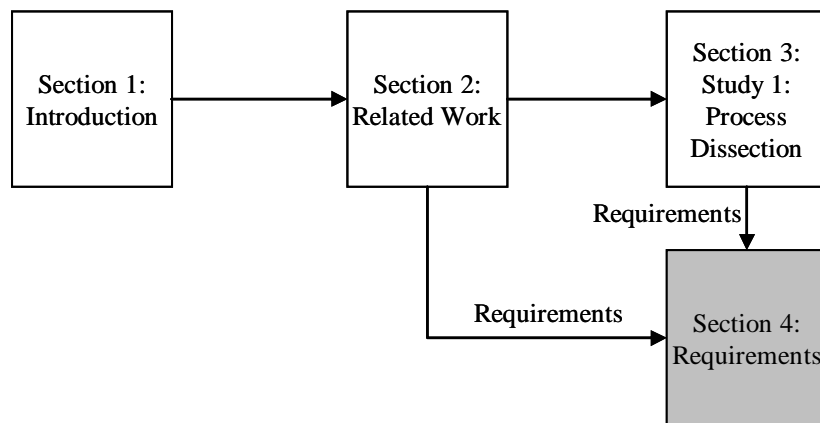
Figure 4.1. Section 4 Dissertation Context

4.1   Software Requirements

4.1.1   Heuristic Evaluation Phases

☐ *Requirement 2.4. Fully support Heuristic Evaluation by providing capabilities for inspection, usability problem preparation, and usability problem aggregation.* See Section 2 for the description.

4.1.2   Inspection

☐   *Requirement 4.1. Provide capabilities for inspection.*

☐   *Requirement 4.2. Provide ways to specify "to do" items per Web page or in general.* When a session is interrupted, "to do" item can be made so that they can be taken care later.

☐   *Requirement 3.1. Provide observation tools to improve evaluator's observation skills*

4.1.2.1   Discovery Resources

☐   *Requirement 2.5. Provide Discovery Resources [Cockton et al. 2004].* Assist evaluators in problem discovery by providing task resources such as goal playing scenarios [Cockton et al. 2003] or, similarly, task scenarios [Nielsen 1995a]. See Section 2.

4.1.2.2   Annotation of User Interfaces

☐   *Requirement 3.3. Provide annotation tools to support evaluators in inspection.*

☐   *Requirement 3.4. Provide ways to highlight/mark interface elements and areas.* Visit Section 3 for the description.

☐   *Requirement 3.5. Provide ways to attach comments to highlighted items or annotations.*

☐   *Requirement 3.6. Provide a way to apply heuristics to interface items or annotations.*

☐ *Requirement 3.7. Annotation type set.* Section 3 includes the description of the requirement.

☐ *Requirement 3.8. Provide ways to connect/associate interface items and annotations.*

4.1.2.3 Elaboration of Usability Problems

☐ *Requirement 3.2. Provide tools to capture usability problems*. See Section 2 for the description.

☐ *Requirement 4.3. First enter a brief usability problem description then elaborate it.* Allow evaluators to enter brief problem descriptions when they are mostly focused on searching usability problems. Problems can be elaborated afterwards.

☐ *Requirement 4.4. Provide capabilities to embed images in usability problem descriptions.* Evaluators should have easy ways to describe problems; embedding images in description are needed.

☐ *Requirement 4.5. Indicate related interface elements mentioned in usability descriptions.* Describing interface elements in descriptions can be tedious (e.g. "the second menu option on the upper right corner"). Provide easy ways to indicate related elements in problem descriptions. This can be done with drag-and-drop operations or element selection in the interface.

☐ *Requirement 4.6. Encourage the use of "use-impact-related" terminology in text.* Provide features that will encourage evaluators to describe

probable user difficulties when encountering problems. For example, highlighting the word "user" each time evaluators type the word and a pull down menu is accessible with user reaction phrases. The phrases can be embedded when they are selected. Phrases will look like: "gets frustrated, unable to complete a task, difficulty in finding information, other …".

4.1.3   Usability Problem Preparation

☐ *Requirement 4.7. Provide capabilities for usability problem preparation.*

☐ *Requirement 4.8. Usability problem attributes needed for an effective problem aggregation.* "Raw" problems [Cox 1998] (p. 83) need to have a description, violated heuristic, and author identifier for an effective aggregation session.

☐ *Requirement 4.9. Handle usability problem versioning.* We write differently when writing for ourselves than for others (e.g. when making personal annotations on documents public authors modify annotations to make them "intelligible to others" [Marshall and Brush 2004]). Evaluators may want to polish their descriptions before they are released, but still have access to the original descriptions.

☐ *Requirement 4.10. Confirm and eliminate usability problems.* Allow evaluators to mark usability problems as confirmed or eliminated [Cockton et al. 2003].

☐ *Requirement 4.11. Provide analysis tools to determine if a usability problem is confirmed or eliminated.* Cockton and collaborators [Cockton et al. 2003] extended usability problem format can be considered as a problem analysis tool. The format prompts evaluators to go beyond than just reporting the problem. Evaluators reflect on the implications of the problem, provide evidence of heuristic non-conformance, and so on. Provide analysis tools to help evaluators determine if a problem needs to be eliminated or kept in the usability problem set.

4.1.4   Usability Problem Aggregation

☐ *Requirement 4.12. Provide capabilities for usability problem aggregation.* Requirements for problem aggregation have been identified elsewhere [Cox 1998]. Please visit Cox's work.

4.1.5   Heuristic Checklist

☐ *Requirement 4.13. Define new heuristic checklists.* Capabilities to create new heuristic checklists should be available. Checklists can be created based on other checklists. Heuristics can be added and removed. Once a checklist is completed it can be incorporated in the Suite's checklist collection.

☐ *Requirement 2.1. Specify the heuristic checklist to be used.* See Section 2 for its description.

☐ *Requirement 4.13. To edit checklist definitions.* Capabilities for changing an unreleased checklist' attributes should be in place. Checklist attributes

include a title, author(s), publication date, reference/bibliography, domain (calendar, E-commerce, communities, medical systems, etc.) and platform it applies (Web and WIMP), as well as the version.

□ *Requirement 4.14. To manage heuristic checklists.* There is the need to delete, add, duplicate, and restore previously deleted checklists. The system has core heuristics that can't be deleted. Only user-defined checklists can be removed from the system's checklist collection. User-defined checklists should be saved as part of HEAssistant projects.

□ *Requirement 4.15. Printable heuristic checklists.* Evaluators may choose to have the checklist handy during inspection or when elaborating usability problems. Provide a printable version of the checklist so it can be printed in different formats (e.g. short and long versions).

□ *Requirement 4.16. To import and export heuristic checklists.* Checklists can be created and shared evaluators. Evaluator should be able to import and export heuristic checklists. Checklist can be stored in some standard format such as in XML.

4.1.6 *Heuristics*

□ *Requirement 4.17. Edit heuristic definitions.* An unreleased heuristic definition can be modified. Otherwise, a new checklist version should be created to incorporate the changes.

□ *Requirement 4.18. Heuristics attributes.* Heuristics should have at a minimum a title, brief description, and help documentation.

☐ *Requirement 4.19. Provide different heuristic formats in printed checklists or in help documentation.* Heuristics can be presented in different formats. Nielsen [Nielsen 2005b] presents a title then a brief description below it. Dykstra [1993] presents a title and a list of sub-heuristics below it.

☐ *Requirement 4.20. Provide usability problem examples violating heuristics in the Help documentation.* Heuristics are general and difficult to apply for novice evaluators. Provide examples that illustrate in which ways usability problems violate heuristics.

4.1.7 Usability Problem Formats

☐ *Requirement 2.2. Define new usability problem formats for describing usability problems.* Visit Section 2 for the description of the requirement.

☐ *Requirement 4.21. Usability problem format attributes.* The very basic format should have entries for a brief description of the problem and be able to specify one or more heuristics violated by problems, a title, author(s), publication date, reference/bibliography, and version.

☐ *Requirement 2.3. Specify the usability problem format for describing usability problems in a HEA Project.* Check Section 2 for the description.

☐ *Requirement 4.22. Edit usability problem format definitions.* Capabilities for updating unreleased usability problem formats should be available.

☐ *Requirement 4.23. Manage usability problem formats.* There is the need to delete, add, duplicate, and restore previously deleted formats. The

system has core format that can't be deleted. Only user-defined formats can be removed from the system's format collection. User-defined formats should be saved as part of HEA Projects.

☐ *Requirement 4.24. Printable usability problem formats.* Evaluators may choose to describe problems in paper. Provide printable versions of the formats.

☐ *Requirement 4.25. Provide help documentation to describe how to use usability problem formats.* For each entry in the format provide a description and examples how to enter the required information.

4.1.8 *Compatibility*

☐ *Requirement 4.26. Export a HEA project to GroupSystems* [GroupSystems 2009]. GroupSystems is meeting groupware supporting "collaboration for innovation, decision-making, leadership", and new product development. It can be used to aggregate usability problems [Poole 2004]. Provide capabilities to export HEA projects to GroupSystems in case evaluators choose to aggregate problems outside the HEA Suite.

4.1.9 *HEA Projects*

☐ *Requirement 4.27. Setup a HEA Project.* Project administrators should be able to set HEA Projects shared by the evaluation team. Projects ill include a title, description, evaluation team information, usability problem form and heuristic checklist to be used, goals of the Heuristic

Evaluation, tasks assigned to evaluators, and general instructions to provide more guidance to evaluators.

☐ *Requirement 4.28. Assign task to evaluators.* Evaluators can be assigned to concentrate on a part of the interface, on some functionality or, similarly to the Perspective-Based Usability Inspection method [Zhang et al. 1999] –where evaluators focus on identifying usability problems of a certain type such as novice use, expert use, and error handling, on identifying usability problems relevant to certain heuristics.

4.1.10 Reports

☐ *Requirement 4.29. Generate Affinity diagram materials.* Evaluators may choose to aggregate usability problems manually using Affinity diagrams [Snyder 2003]. Capabilities for printing materials from usability problems and project information should be in place.

☐ *Requirement 4.30. Facilitate generating usability reports.* Capabilities for generating usability reports should be available. Report templates can be designed based on the CUE usability reports [Molich et al. 2004; Dumas et al. 04; Molich and Dumas 2007, 2008]. Comparative Usability Evaluation reports were created by professionals applying different methodologies including Heuristic Evaluation and usability testing.

4.1.11 Search Capabilities

☐ *Requirement 4.31. Provide search capabilities to easily retrieve usability problems.* Heuristic Evaluation is known to generate a large number of

usability problems. Evaluators need search tools to retrieve usability problems.

5   THE TOOL: HEASSISTANT

In the previous section software requirements for developing Heuristic Evaluation tools were identified. Software requirements for annotation tools for inspection are among those requirements. A tool designed based on annotation requirements was developed, called "HEAssistant–Annotator" (it will be referred only as "HEAssistant" in this dissertation). This section describes HEAssistant v0.1.

5.1   HEAssistant v0.1

HEAssistant v0.1 is an annotator of Web pages. It is the first tool in a series of tools for supporting Heuristic Evaluation for the Web. The annotator was designed to be used in the inspection stage of Heuristic Evaluation when evaluators carefully examine Web pages in search for violations to heuristics.

The annotator was built as a Mozilla Firefox [Mozilla 2009a] add-on/extension. It was built on top of another add-on called ScrapBook [Gomita 2009]. This add-on handles a single annotation type (Sticky Notes), which is not enough for the purposes if this research.

HEAssistant v0.1 is a multi-item package [Mozilla 2009b]. ScrapBook v1.3.3.9 and HEAssistant v0.1 are automatically installed when installing the package. By handling ScrapBook separately we can easily update the package with a newer version. HEAssistant v0.1 is compatible with Mozilla Firefox 2 and 3.

5.2   HEAssistant User Interface Design

5.2.1   Views

Two views were implemented to make annotation capabilities more accessible when inspecting Web pages. The idea was to be able to start annotation whenever it was needed without having to open menus and making selections. These views are: Normal View and Notes View.

In the Normal View evaluators perform usual browsing. They navigate and interact with a Web interface to learn it, get first impressions, experience usability problems, and exercise the interface. After the HEAssistant v0.1 package is installed the ScrapBook main menu will appear in Firefox's top menu, and HEAssistant Normal View tool bar will appear on the bottom (Figure 5.1). In the Notes View (Figure 5.2) evaluators annotate Web pages using any of the annotation types in the annotation type set. The annotation type set is described in the following section.
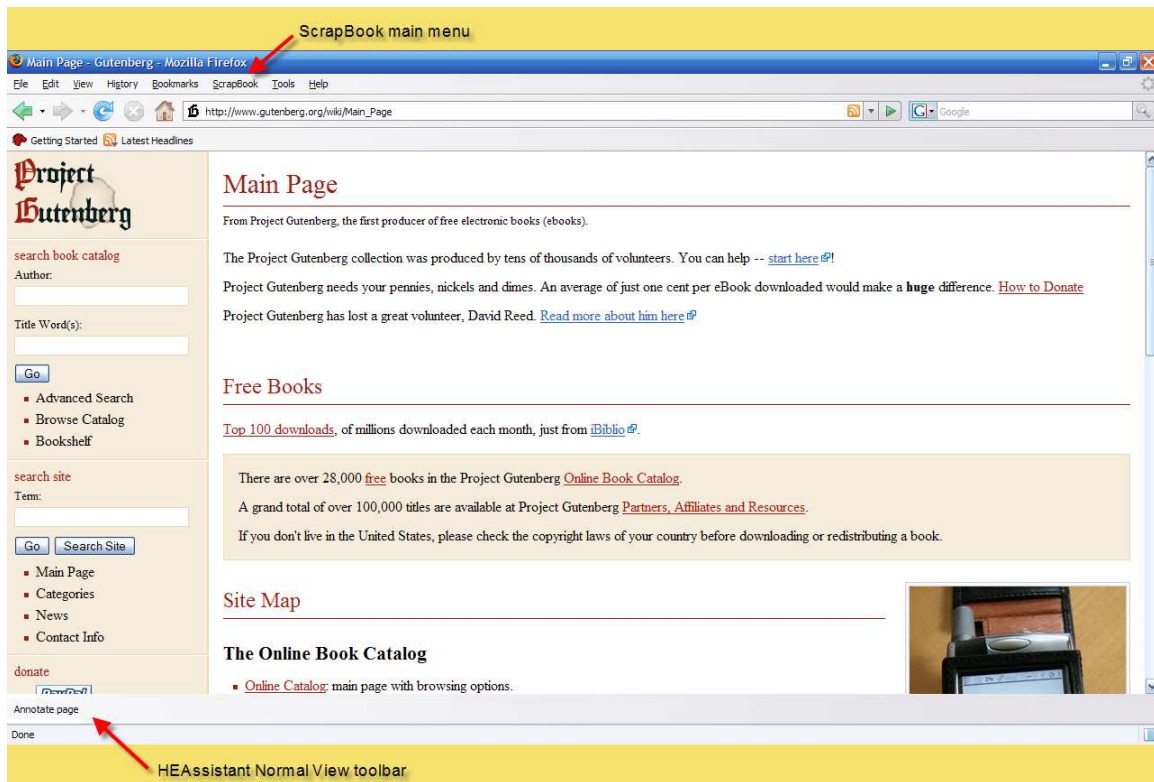
Figure 5.1. HEAssistant Normal View
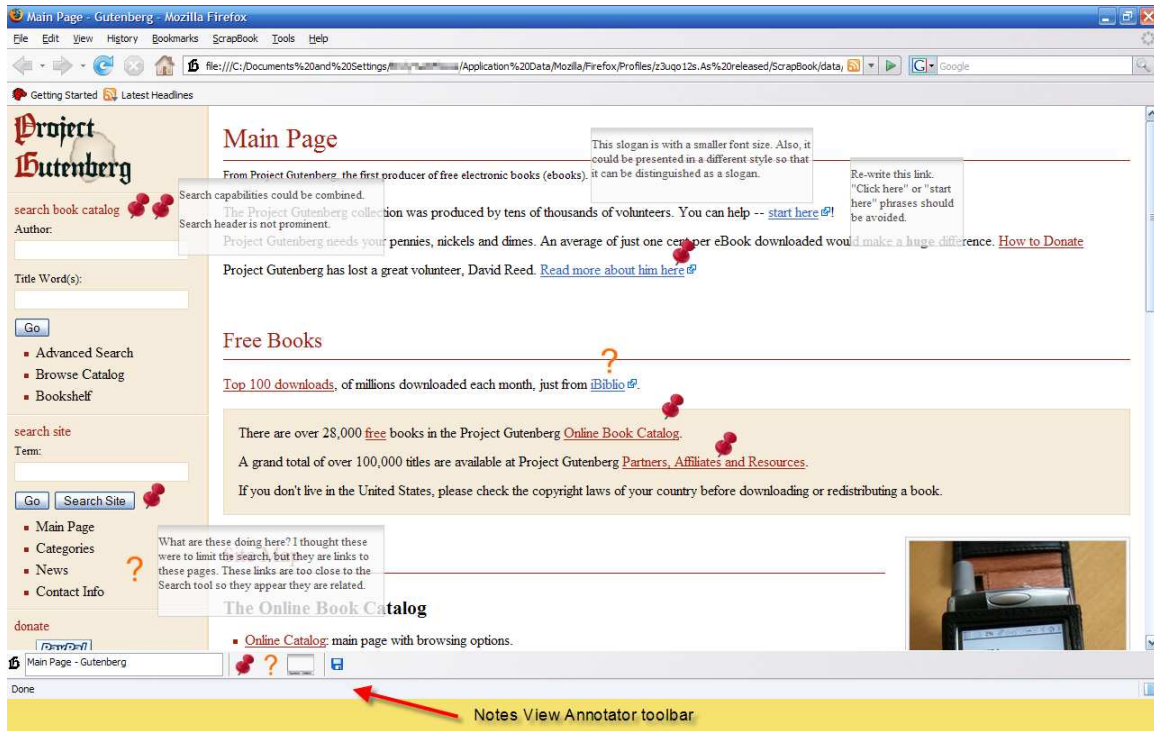
Figure 5.2. HEAssistant Notes View

Three internal steps take place when evaluators stop browsing and decide to annotate the current Web page. After pressing the "Annotate page" button in the Normal View toolbar a copy of the current Web page is saved in Firefox's current profile directory, the copy is automatically loaded in the Notes View, and the Annotator toolbar appears at the bottom of Firefox browser. Evaluators can, then, start annotating the Web page.

Firefox profiles [Mozilla 2009c] are a collection of personal information such as histories, extensions installed by the user, cookies, and user files. As a developer,

creating several profiles resulted very useful. Profiles were used to test different HEAssistant versions.

Leaving the Notes View to go back to the Normal View can be done through the usual browsing capabilities of the browser such as clicking on links, entering a new URL in the browser's address textbox, and visiting bookmarked Web pages. To view a previously annotated Web page, the ScrapBook's main menu (Figure 5.3) is used.

Figure 5.3. View an Annotated Web Page through ScrapBook's Main Menu

5.2.2   Annotation Type Set

HEAssistant v0.1 has capabilities to add different types of annotations to Web pages: Notes, Push Pins, and Question Marks. Figure 5.4 shows the Annotator toolbar in the Notes View.
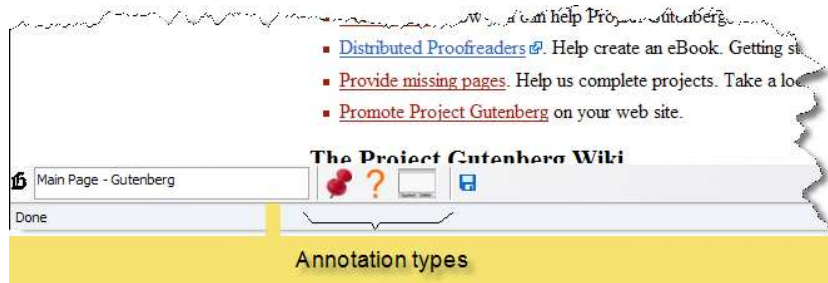
Figure 5.4. Notes View Annotator Toolbar

Notes. Notes are considered to be the very essential annotation type. In this version of HEAssistant we support simple text notes.

Push Pins. An annotation type to quickly mark interface elements or areas was needed. A "X" was considered at first, but it was discarded because it might be confused with a deleting functionality. A push pin representation was chosen instead. It was proposed from a map metaphor where detectives mark crime scenes on maps with push pins.

Push Pins were designed so that they can be added quickly, and without selecting an option each time one was to be added. For this, the corresponding button in the Annotator toolbar was left selected until no more Push Pins were needed or another annotation type was selected. The rationale behind this was to reduce the time the inspection process is interrupted by having to select options.

Question Marks. It has been observed that there is an occasional use of questions in usability problems forms and annotated paper-based Static interfaces. In Study 1, there were 2 evaluators who chose to annotate a given Static Web page. One of them made 3 annotations containing questions (e.g. "WHERE AM I?..." and "WHY"), and

another annotation being a double question mark ("??"). The second evaluator annotated a question. The use of Question Marks needed to be studied further, and, hence, it was included in the annotation type set.

5.3   HEAssistant Implementation

I would like to acknowledge Koji Ouchi for his contribution to this project. He was involved in HEAssistant's final development stage.

5.3.1   HEAssistant Architecture

A typical three-layered software architecture [Bass et al. 2003] was implemented to build the HEAssistant add-on. The *Presentation* layer was developed with the XML User Interface Language (XUL) [Mozilla 2009d]. XUL is used to define user interfaces of Mozilla-based applications such as Firefox. The *Business Logic* layer was developed in JavaScript. Annotations were directly saved in annotated Web pages. These Web pages form part of the *Data* layer.

Much was learned from the ScrapBook project. It was used to better understand how to develop Firefox extensions and as a basis for HEAssistant development. A difference between HEAssistant and ScrapBook is that in the HEAssistant project an Object-Oriented approach was followed to extend the annotation type set more straight forward. ScrapBook users can add comments to captured Web pages with "Sticky Notes." The first challenge was to identify and extract code related to "Sticky Notes" and encapsulate it in a HEAssistant class called heaNote. Additional member functions were included in heaNote to implement the needed Note behavior. Once the heaNote

class was working, the heaPushPin and heaQuestion classes were implemented to support the Push Pin and Question Mark behaviors, respectively.

Figure 5.5 shows the software architecture of the HEAssistant package. The ScrapBook extension was kept separately to integrate newer versions in the future.
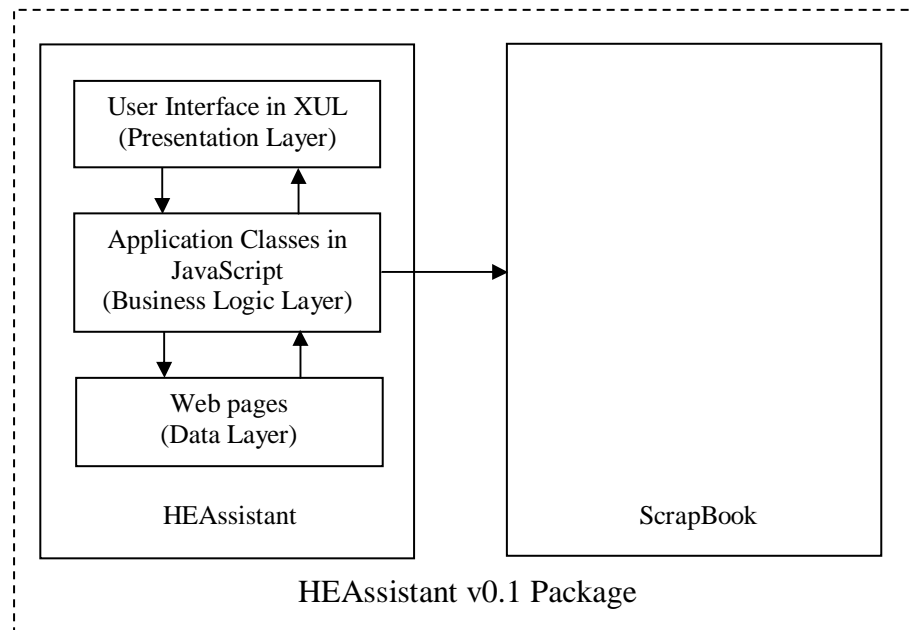
Figure 5.5. HEAssistant v0.1 Software Architecture

5.3.2 GRASP Patterns

General Responsibility Assignment Software Patterns (GRASP) [Larman 2004; Freeman et al. 2004] were used as a guide to determine classes and their main role in the design. Classes can be viewed as entities responsible for fulfilling tasks or subtasks. GRASP patterns are design principles followed by expert designers to assign responsibilities to classes [Valtech 2000]. The "Expert" GRASP pattern, for instance, is

about assigning a responsibility to a class that has the necessary information to complete the responsibility [Valtech 2000]. Whenever HEAssistant users add Notes to a Web page objects that appear like a notes are added. The heaNote class is "the expert" that has the necessary information to produce a code segment that renders as a Note in the Web page. GRASP patterns were used throughout HEAssistant.

6 STUDY 2: TOOL EVALUATION

In the previous section HEAssistant was described. HEAssistant is a Web page annotator for Heuristic Evaluation. This section describes Study 2 where HEAssistant is evaluated and annotation usage is studied.

6.1 Study Design

The study consisted in an online survey where participants were asked to download the tool, try it, and answer a questionnaire about their background and experience using the tool. Training was involved for those participants not familiar with Heuristic Evaluation. All procedures were done online. A Web site for the study was developed to guide participants through the process and download the tool and training materials.

6.1.1 Participants

Data from 22 participants was analyzed in the study. Web developers and software engineers with at least a year of experience were recruited. Students and professionals were invited to participate in the study. Undergraduate and graduate students from both "Department of Computer Science and Engineering" and "Department of Information & Operations Management," Texas A&M University, were recruited. Professional Web developers and software engineers from inside and outside the Texas A&M University community formed part of the recruitment pool (Table 6.1).

Table 6.1. Study 2 Participants

	Professionals	Students	**Total**
Software Engineers	6	5	11
Web Developers	7	4	11
Total	13	9	**22**

Six participants were familiar with Heuristic Evaluation before participating in the study (Table 6.2).

Table 6.2. Participants with Heuristic Evaluation Background

	Professionals	Students	**Total**
Software Engineers	0	2	2
Web Developers	2	2	3
Total	2	4	**6**

6.1.2 *The Interactive Web Interface*

Participants performed a Heuristic Evaluation on an interactive Web page using the tool. The Texas A&M University Transportation Services Web site [Texas A&M University Transportation Services 2009] was chosen for the study. Participants were asked to focus on the home page of the site, but they were allowed to visit other Web pages if necessary. The Web site is interactive and information-based. It has limited functionality to be considered a Web application. Appendix C includes the Web site home page, and other pages one level down the hierarchy (City and University Traffic Construction, Parking Rules and Regulations, Pay Citation, and Search Results).

6.1.3   Procedures

Participants went through several steps in the study. All procedures were done online and on their own schedule. They could work on the different steps at different times. Let me summarize the procedures:

0. Heuristic Evaluation training (80 minutes)

    a.   Read training materials. (40 minutes)

    b.   Practice performing a Heuristic Evaluation with no tool. (40 minutes)

1. Install the tool (5 minutes)

2. Try the tool (45 minutes)

    a.   Complete a ten-minute tool tutorial.

    b.   Use the tool while performing Heuristic Evaluation on a Web page.

3. Complete a post-questionnaire (35 minutes)

6.1.3.1   Heuristic Evaluation Training

Training was available for those people not familiar with Heuristic Evaluation. It consisted in reading training materials and practicing applying Heuristic Evaluation on a given Web site. Among the training materials were a Heuristic Evaluation Booklet, Nielsen's ten usability heuristics [Nielsen 2005b], and Wood's explanation of the heuristics but for the Web [Wood 2004]. The booklet describes Heuristic Evaluation, how to conduct it, and a usability problem form to be used to describe usability problems in the Web site. The usability problem form only had two entries: problem description and which heuristics were being violated by the usability problem. The booklet can be found in Appendix C.

Forty minutes were assigned for reading all documents. After learning Heuristic Evaluation, participants spent an additional 40 minutes in an exercise. They were asked to apply a Heuristic Evaluation on the "Project Gutenberg [Project Gutenberg 2009]" Web site. Appendix C includes instructions given to participants in the exercise. Usability problems that were found in the exercise were not submitted.

6.1.3.2   Tool Installation

HEAssistant v0.1 was developed as a Mozilla Firefox extension/add-on. Evaluators downloaded HEAssistant from the Study Web site and install it in Mozilla Firefox 2 or 3.

6.1.3.3   Tool Evaluation

Participants followed a ten-minute tool tutorial before performing the main task. The tutorial covered how to annotate Web pages, saving changes, and retrieving annotated Web pages. They were asked to reserve time to complete the questionnaire right after performing the main task. This was for the purpose of capturing their opinions right after using HEAssistant when they have the freshest recollections of their experiences using the tool.

The task given to participants consisted in using HEAssistant while performing a Heuristic Evaluation on a given Web page. Thirty-five minutes were assigned for the task. Participants were instructed to evaluate the home page of the Texas A&M University Transportation Services Web site [Texas A&M University Transportation Services 2009]. They were asked to focus only on the home page, but they were allowed to visit other pages if necessary. The same usability problem form used in training was

used to report usability problems they found in the Web site. Since the study was designed to learn how annotations were used during a Heuristic Evaluation, participants were encouraged to use annotations extensively whenever it was appropriate. Instructions given to participants can be found in Appendix C.

6.1.3.4 Questionnaire Completion

Participants were asked to complete a questionnaire about their background and experience using HEAssistant v0.0 right after trying it. The questionnaire was available in text and Microsoft Word formats. People downloaded the questionnaire, answered it, and submitted it via e-mail. Thirty-five minutes were assigned for completing it. There were 2 versions of the questionnaire (see Appendix C), one for Web developers and another for Software Engineers. The only differences were questions 2-3 regarding their background.

The following deliverables were requested to participants after the completion of the study:

1. Annotated Web-page(s) generated while using HEAssistant v0.1.

2. Usability problems identified when evaluating the given Web page.

3. Questionnaire responses.

Participants completing the study were compensated with a small fee (gift card) for their participation in the study.

6.2   Results

Annotations can help us remember things in inspection. The question is what kind of things. This section investigates the types of annotations, usability problems, and uses of annotations by evaluators to answer this question

6.2.1   Note Characterization

It was of interest to learn how evaluators used Notes in inspection by learning the kinds of comments evaluators entered in Notes.

The content of 45 Notes created by 10 evaluators was analyzed. These are a subset of the Notes that were annotated. Evaluators had difficulties in saving the content of Notes with HEAssistant, and these are the cases in which the content was saved successfully. An evaluator saved the Notes content in a separate file and submitted the file. The analysis includes these Notes. See Appendix B for the description of HEAssistant's saving usability problem.

6.2.1.1   Notes Commonly Used to Describe Problems

There were several Note types that were identified (Figure 6.1), but the majority of Notes were found to be informal usability problem descriptions. Notes of other types rarely occurred but they give us some idea of possible Note uses. There was an explicit reminder, a reference to another Note, and a coded Note which only the author would know its purpose. Table 6.3 shows Note examples of all identified types.
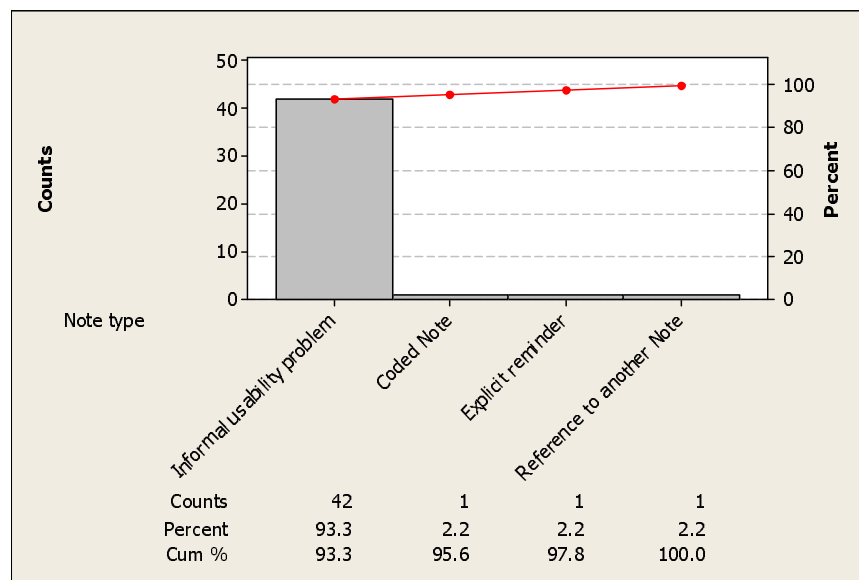
Figure 6.1. Note Types

(n=45)

Table 6.3. Note Examples

Note type	Example
Informal usability problem description	"Hyperlinks are not identical (some have underlines and some do not) and the color is not distinguishable"
Coded Note	"Link for break transit service ([semester] as well as spring [break])"
Explicit reminder	"Make note of this"
Reference to another Note	"9. (Aggieland Saturday links) Same comment as #6"

6.2.1.2   Problem Indicators

Problem descriptions in Notes had several elements that indicate they pertain to usability problems. Problem descriptions were found to have 1 or 2 of the following content elements: observations, recommendations, questions, and unexpected results (Figure 6.2).
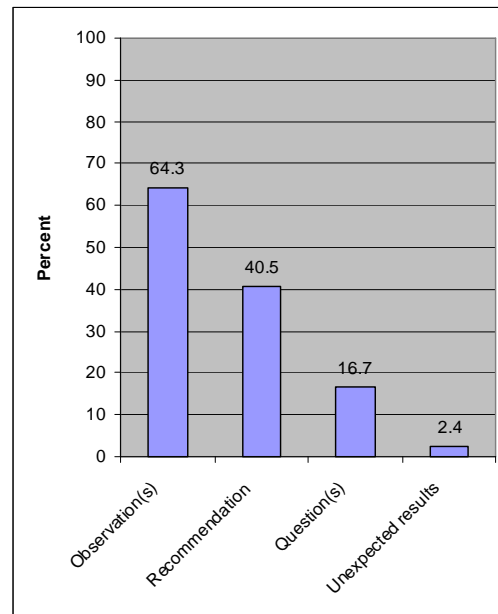
Figure 6.2. Problem Content Elements

(n=42)

It was found that 64.3% of Notes include observations of "bad" interface features. Forty percent (40.5%) of Notes include recommendations. Notice that a recommendation on its own implies a solution to fix a (explicit or implicit) usability problem. Questions also appeared in Notes. Some questions show possible confusion/unclearness or doubt for having a feature. A Note was found to include unexpected results obtained from using an interface feature. Table 6.4 includes Note element examples.

Table 6.4. Notes with Content Elements

Element	Note
Observation	"Wasted space"
	"too many menus, too confusing!"
	"3. There seems to be no logical grouping of the 6 items displayed in this area."
Recommendation	"This seems like a nice functional section. If you're a visitor for Aggieland Saturday, you might need to know this. But *this should be linked to Campus Maps*."
	"Background for Latest Announcements *should use a color with better contrast*."
	"If it's linking a pdf file and not a web page, *it would be better to add a Acrobat PDF icon*."
Question	"Who needs an account? What's the advantage?"
	"The order that these tabs are following looks haphazard (maroon-green-maroon again?)"
	"Which directory does this colored tab belongs to? Other seem to have coherent colors with the leftmost vertical directories."
Expectation	"The left tabs look like they will open in the current window like a drawer. But instead opens a new page. Not a big issue."

6.2.2 *Usability Problem Characterization*

Sixty-eight (68) usability problem reports submitted by 15 evaluators were analyzed. A problem report was excluded from the analysis because the problem description was trimmed and could not be analyzed.

It was observed that some usability problems were more complex than others. A proportion of problems were found to be "Compound". Compound problems relate to two or more problem matters, and can be split in to sub-problems. Other problems relate only to a matter, called "Single." Table 6.5 provides Single and Compound problems examples.

Table 6.5. Single and Compound Problems.
Compound sub-problems in are enclosed in angle brackets (<>)

Single/Compound	Example
Single	"All of the links on this page add a background behind the link that when hovered over, covers up other text that makes it hard to read."
Single	"Arbitrary ordering of items in the center of the page. The "Special event" items are separated from each other."
Compound (3 sub-problems)	"<The navigation is split into two places (left side and top). This is confusing because as a user I don't know where to begin> and <it is difficult to know where I am because, while there is a header when navigating to a subpage, the navigational elements do not indicate anything has changed.> <Further, I'm not sure if clicking on the Parking element on the left will give me anything more than what I see on the right side links which seem to be color coordinated to be 'Parking' links.">
Compound (2 sub-problems)	"<The site feels overwhelmingly busy at first visit.> <I have no idea what kinds of things I can do here. Can I pay a ticket or buy a pass? I'm not sure where to start looking.>"

The majority of problems were found to be Single (73.5%), which is good. We want to keep the number of Compound problems low. Nielsen [Nielsen 2005a] recommends listing usability problems separately (even if they relate to the same interface element) so that they can be handled individually. If Compound problems are treated as Single problems, there is a risk of not analyzing and fixing all sub-problems.

Figure 6.3 shows the proportion of Single and Compound problems found in the analyzed problems.
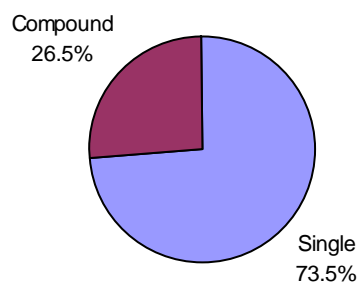
Figure 6.3. Proportion of Single and Compound Problems

(evaluators=15, n=68)

6.2.3   Annotation Usage

6.2.3.1   Overall Usage

It was found that Notes were the annotations of preference, Push Pins come second, and Question Marks third (Figure 6.4). Table 6.6 includes annotations made by evaluators.
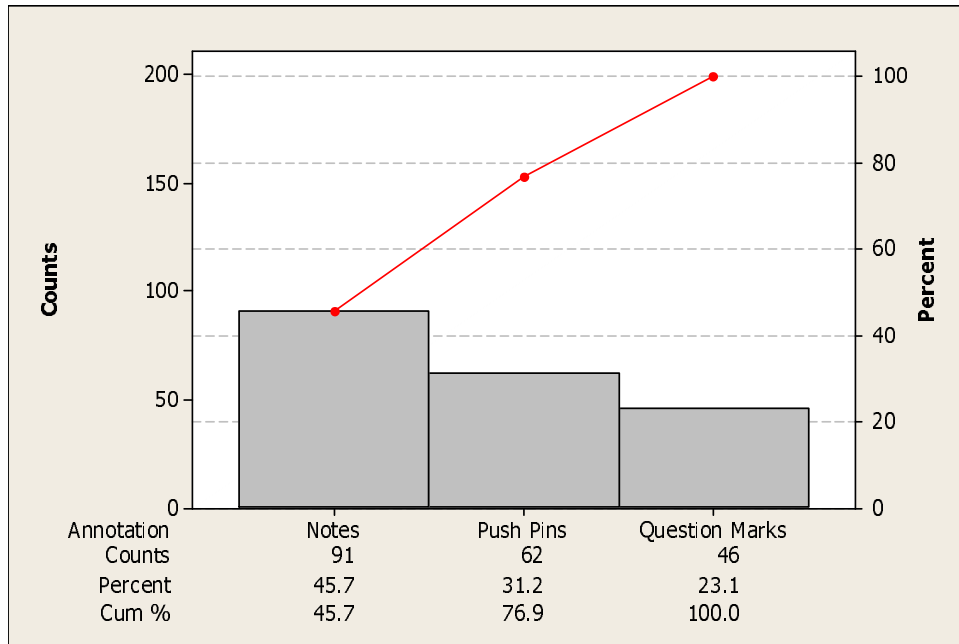
Figure 6.4. Annotation Overall Usage

(evaluators=21, n=199)

Table 6.6. Annotations by Evaluators

#	Notes	Push Pins	Question Marks	Total
1	9	11	1	21
2	12	2	6	20
3	2	4	0	6
4	2	3	4	9
5	8	1	1	10
6	4	0	12	16
7	3	3	1	7
8	3	2	1	6
9	1	2	1	4
10	2	2	0	4
11	5	4	1	10
12	5	4	2	11
13	1	2	2	5
14	5	1	0	6
15	3	2	0	5
16	13	0	0	13
17	0	2	7	9
18	2	8	2	12
19	2	2	1	5
20	8	3	1	12
21	1	4	3	8
Total	91	62	46	199

The annotated Web pages submitted by 21 evaluators were analyzed. There was a case that was excluded from the analysis. The evaluator reported to have deleted all Notes after having difficulties in saving Note contents (see Appendix B for the description of the problem) then added one summarizing all. Hence, the number of annotations would be different than the one obtained. All evaluators reported annotated home pages of the Web site; however, an evaluator submitted 4 additional annotated Web pages. Results presented in this section include annotations from these other Web pages.

A significant number of annotations made by evaluators were Notes (45.7%). This might be due to Notes' capability for adding text. In fact, some evaluators expressed the need for adding text to Push Pins and Question Marks:

> …Notes could appear as default along with question marks, user would find it useful to add notes immediately to ask questions.

> …push pins and question marks should have notes attached to them.

6.2.3.2 Note Usage

A high number of evaluators responded they used Notes to add brief descriptions of usability problems when annotating Web pages in inspection. Using Notes to a) specify heuristics being violated by usability problems and b) to add questions were less popular Note uses among evaluators.

Responses to Question 5 of the questionnaire (see Appendix C for the questionnaire) were analyzed. Question 5 prompted evaluators to indicate in which ways they had used Notes. They were presented with 3 options (in that order): a) to specify heuristics being violated, b) to add questions, and c) to add brief descriptions of usability problems. There was space for evaluators to describe other ways of using Notes. However, no one provided other Notes uses. Table 6.7 summarizes the responses considered in Question 5, 6, and 7 analyses.

Table 6.7. Questionnaire Response Summary.
Questions 5, 6, and 7 are relevant to the use of Notes, Push Pins, and Question Marks, respectively

Condition		Response counts		
		Question (relevant type)		
		5 (Notes)	**6** (Push Pins)	**7** (Question Marks)
Not applicable		0	1	3
Annotations of type were reported	"I don't know" responses	0	2	1
	Responses related to uses	22	19	18
Total		**22**	**22**	**22**

6.2.3.2.1 Notes are Commonly Used to Describe Problems

The majority of evaluators (72.7%) indicated they used Notes to add brief descriptions of usability problems (Figure 6.5). This agrees with what was obtained when analyzing Note contents in section 6.3.1. The majority of Notes in content analysis were found to be informal descriptions of usability problems. *Using Notes to describe usability problems briefly is a good indicator that Notes serve as reminders of usability problems.*

	c) To add brief descriptions of usability problems	a) To specify heuristics being violated	b) To add questions
□ Undetermined (%)	0	0	0
■ Not Checked (%)	27.3	59.1	63.6
■ Checked (%)	72.7	40.9	36.4
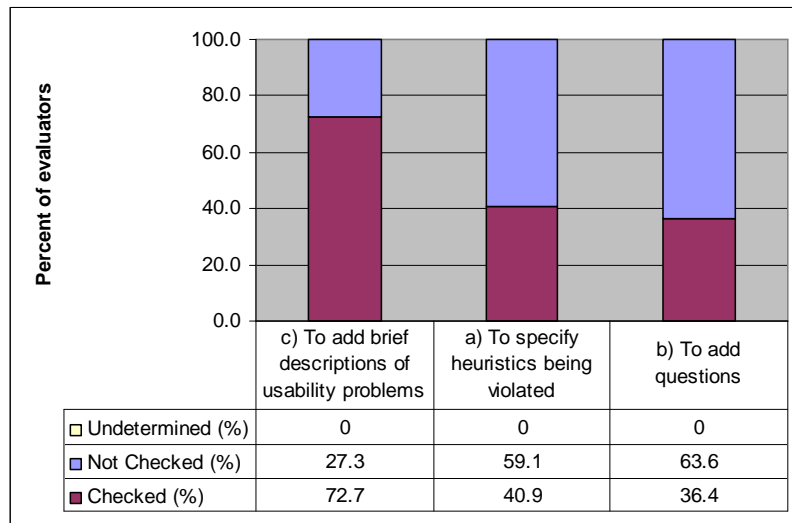
Figure 6.5. Note Uses

"Checked" answers indicate Notes were used in a certain way. "Undetermined" answers indicate "I don't know" responses (n=22)

6.2.3.3   Push Pin Usage

A large number of evaluators responded they used Push Pins to mark problem interface elements or areas when annotating Web pages in inspection. Using multiple Push Pins to mark problem interface elements or areas with several problems was a less popular Push Pin use among evaluators.

Responses to Question 6 of the questionnaire were analyzed (Table 6.7). Question 6 was designed to learn Push Pins uses. Evaluators were asked if they used Push Pins in the following ways (in that order): a) to mark problem interface elements or areas, and b) to mark problem interface elements or areas with several problems using multiple Push Pins. Figure 6.6 shows the results of the analysis.
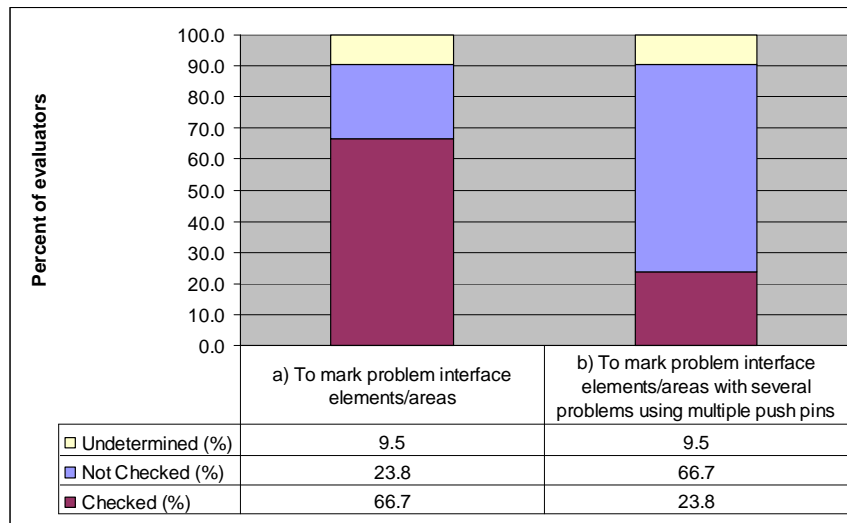
Figure 6.6. Push Pin Uses.

"Checked" answers indicate Push Pins were used in a certain way. "Undetermined" answers indicate "I don't know" responses (n=21)

6.2.3.3.1 Push Pins are Used to Pinpoint Problem Features

A significant number of evaluators (66.7%) responded to have used Push Pins to mark problem interface elements or areas. Using Push Pins to mark problem features might be due in part to their capability to easily pinpoint the *location* of usability problems as an evaluator points out:

> Explaining *where* in the web page is problematic and *how* is not easy to do with plain text. Pinpointing and adding annotations directly on the web page seems to be very efficient…

6.2.3.3.2 Other Push Pins Uses

Evaluators mentioned other ways of using Push Pins in Question 6 and comments. Table 6.8 summarizes these uses.

Table 6.8. Other Push Pin Uses Mentioned by Evaluators

Use	Evaluator's comment
To pinpoint relevant problem features mentioned in Notes	"To support a note", "…pushpin helps to pinpoint the area the note applied to."
To mark visited places	"I marked areas that I would normally to go to."
To highlight something important	"For the purpose of highlighting or something that I feel is important"

6.2.3.4   Question Mark Usage

Sixty-three percent (63.2%) of evaluators responded to have used Question Marks when drawing questions about marked problem interface elements or areas. Another fifty-seven percent (57.9%) of evaluators responded to have used Question Marks to mark problem interface elements or areas which had something confusing about them.

Responses to Question 7 of the questionnaire were analyzed. Responses of 19 evaluators were considered (see Table 6.7). In Question 7 evaluators were asked if they used Question Marks in the following ways (in that order): a) to mark problem interface elements or areas when drawing questions about them, e.g. "What does this mean?", and b) to mark problem interface elements or areas which have something confusing about them. Figure 6.7 shows analysis results.
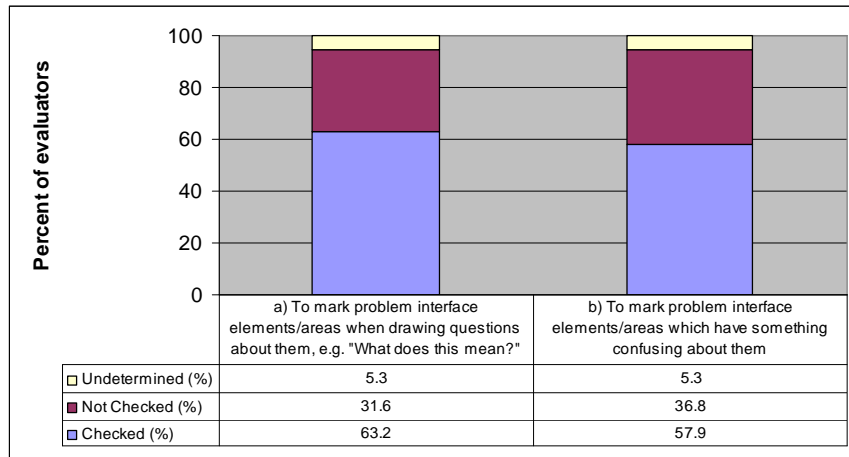
Figure 6.7. Question Mark Uses.

"Checked" answers indicate Question Marks were used in a certain way.
"Undetermined" answers indicate "I don't know" responses (n=19)

6.2.3.4.1 Question Marks are Used to Mark Problem Interface Features

Evaluators reported to have used Question Marks to mark problem interface features either when drawing questions about them (63.2%) or when features being confusing (57.9%). The use of Question Marks may be attributed in part to its appearance. In either case, the character "?" may be a good fit for representing questions and confusion. Question marks are usually part of interrogative sentences. In addition, it can be used to represent confusion. For instance, Google's Image Search engine [Google 2009] returns the string "question mark" as a related search term for the term "confusion" (Figure 6.8).
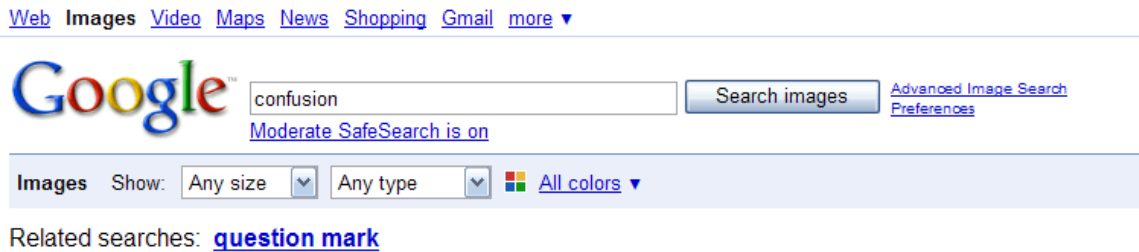
Figure 6.8. Google Image Search Related Searches for Term "Confusion"

6.2.3.5   Annotation Characteristics

In this study we used 3 types of annotations: Notes, Push Pins, and Question Marks. Each of these has affordances (i.e. characteristics objects have that determine how they can be used [Norman 2002]) that hint evaluators how to use them. Notes have text areas that allow evaluators to add text to them. Push Pins have pointed tips to pinpoint interface elements or areas. Question Marks' appearance emphasizes a feature that is confusing.

Each annotation type allows evaluators to record some aspect of usability problems explicitly and implicitly. Evaluators *explicitly* record brief *problem descriptions* in Notes, and *implicitly* record *problem locations*. Push Pins and Question Marks allow evaluators to *implicitly* record *problem locations*. Evaluators pinpoint problem features with Push Pins and mark confusing features with Question Marks.

Table 6.9 summarizes annotation affordances and capabilities, as well as problem aspects that are recorded when using annotations.

Table 6.9. Annotation Characteristics

Annotation	Relevant affordances	Capabilities	Recorded usability problem attribute	
			Explicit	**Implicit**
Note	Text area, appearance	To add text	Content	Location
Push Pin	Pointed tip, appearance	To pinpoint a problem element/area		Location
Question Mark	Appearance	To highlight a confusing element/area		Location

6.2.3.6   Annotation Uses Summary

Table 6.10 summarizes annotation uses so far identified.

Table 6.10. Compilation of Annotation Uses

Annotation	Uses
Note	To specify heuristics being violated
	To add questions
	To add brief descriptions of usability problems
	To provide rationale for placing Push Pins in marked places
Push Pins	To mark problem interface elements/areas
	To mark problem interface elements/areas with several problems using multiple push pins
	To pinpoint relevant problem features mentioned in Notes
	To mark visited places
	To highlight something important
Question Marks	To mark problem interface elements/areas when drawing questions about them, e.g. "What does this mean?"
	To mark problem interface elements/areas which have something confusing about them

6.3   Discussion

6.3.1   Improving Annotation Capabilities

HEAssistant v0.1 allows evaluators to annotate Web pages in inspection. Evaluators can add Notes, Push Pins, and Question Marks. We have seen how they can be used to mark where usability problems are, what those problems are, and highlight confusing interface features. The question is to determine which other annotation types and features can be implemented to improve HEAssistant's annotation capabilities.

We can improve annotation in different ways (these are proposed based on evaluators comments):

  □ *Highlighting capabilities*. Some evaluators mentioned the need for other ways to highlight problem elements or areas. Pinpointing interface elements may be effective with Push Pins but better ways may be needed for highlighting interface areas. For example, by using markers and boxes:

> There could be an option to have marker option to highlight only certain areas for which [I] am adding the notes/question[s]

> … I think it would be good if a tool to select a box-shaped area with color is added because sometimes the usability problems are shown in the "area" not in the specific "point".

  □ *Highlighting positive aspects*. It is recommended that usability reports not only include usability problems found in an interface, but also positive aspects of it [Dumas et al. 2004]. Reports may be received better by

development teams if they are not negative in their entirety. Indeed, an evaluator mentioned the need for an exclamation mark (!) to highlight positive aspects of user interfaces:

> How about [an] exclamation mark to highlight the good aspect[s] from the target webpage

- *Association capabilities*. There is the need for associating annotations. Evaluators reported the need for connecting annotations and grouping them:

> Connect related problems. Create graph of problems
>
> … Ability to group together annotations
>
> … There is no direct way of associating pins/question marks with notes

- *Changing annotation properties*. Annotation properties can be changed to express a particular matter better. For example, changing annotation's color, size, text format, and relevant violated heuristic(s).

- *Labeling annotations*. This can facilitate "referring" to them easier than relying solely on their appearance and location:

> I was expecting to see labels that I could add attached to the pushpins. This would make referring to problems much easier…
>
> … Ability to … add titles to notes.

6.3.2   Annotations as Reminders of Problems

Annotations serve as reminders of usability problems after inspection. A reminder is by definition "something that recalls the past [Collins English Dictionary 2006]." Evaluators add annotations to interfaces while inspecting them. They briefly describe problems in Notes. They mark problem features with Push Pins, and possibly add Question Marks beside confusing features. After inspecting the interface, evaluators may go back to the annotations and recall problems from them. Notes are more informative than Push Pins and Question Marks. They contain brief problem descriptions. Push Pins and Question Marks give pointers where problems were found, but evaluators have to do the rest to recall problems and specifics.

6.3.3   Annotation Supports Inspection: Keeping the Focus on Inspection

Annotation supports inspection. It allows evaluators to focus on inspection and not worry about elaborating problems while examining an interface. It is a mechanism for leaving traces of findings a long the way and continue searching for problems. When something has been noticed as a "bad" feature, evaluators can add quick Notes, Question Marks or Push Pins and worry about them later. They can always return to annotations and recall problems from them.

With no annotation support evaluators may want to write down problems before forgetting them. Forms used to document problems may be long, containing a number of items to fill in. Long forms may require evaluators to not just document problems, but to reflect on them [Cockton and Woolrych 2001]. This may take time and evaluators may interrupt inspection, having to start a new inspection cycle.

6.4   Annotating After Annotating – Recommendations for Annotation

Evaluators are recommended to go through two annotation passes to improve results. In the first pass evaluators annotate freely during inspection. Annotations are added freely so that evaluators can concentrate on the inspection and not on annotation. In the second pass, rationales for annotating non-textual elements such as Push Pins and Question Marks are added to the annotated interface. It was found that Notes have a higher use in problem reporting than non-textual annotations. This suggests that after inspection evaluators can go back to non-textual annotations and comment on these elements, making information about problems explicit, visible, and ready to be used when reporting problems. It is expected that the number of reported problems is increased by following these recommendations.

The background for the recommendations continues.

6.4.1   Annotations' Impact on Reported Problems

A multiple linear regression model was generated in order to see the response of the number of problems reported predicted by three types of annotations, Push Pins, Question Marks and Notes. The regression equation is

```
Problems = 1.02 - 0.105 Push pins - 0.016 Question marks + 0.703 Notes
```

and the ANOVA table for the model is shown in Table 6.11.

Table 6.11. Analysis of Variance Table for the "Problems" Model

Source	DF	SS	MS	F	P
Regression	3	73.724	24.575	4.19	0.037
Residual Error	10	58.633	5.863		
Total	13	132.357			

The p-value (0.037) in the table indicates that the model estimated by the regression procedure is significant. The results of the regression coefficients are shown in Table 6.12.

Table 6.12. Estimated Coefficients for the Predictors of "Problems"

Predictor	Coef	SE Coef	T	P
Constant	1.023	1.663	0.62	0.552
Push pins	-0.1047	0.2284	-0.46	0.656
Question marks	-0.0163	0.2259	-0.07	0.944
Notes	0.7031	0.2076	3.39	0.007

As it can be observed, the p-value for the estimated coefficient of "Notes" is 0.007, indicating it is significantly related to "Problems". On the other hand, the p-values for the estimated coefficients of "Push Pins" (p=0.656) and "Question Marks" (p=0.944) indicate they are not related to "Problems" at an α-level of 0.05. These conclusions agree with the correlation analysis.

The Pearson correlation coefficient for the relationship between Notes and Problems is 0.740 (p=0.002), indicating a strong correlation. In contrast, we cannot conclude that a correlation exists between Push Pins and Problems (Pearson correlation coefficient=-0.142, p=0.627) or between Question Marks and Problems (Pearson correlation coefficient=-0.136, p=0.642) at an α-level of 0.05.

Table 6.13 includes the numbers of annotations and problems used in the analyses. Fifteen (15) evaluators submitted problem reports; however, a case with an

inconsistency in the number of Notes was excluded from the analysis (see Section 6.2.3.1 for details of this case).

Table 6.13. Annotations and Problems by 14 Evaluators

#	Notes	Push Pins	Question Marks	Problems
1	9	11	1	3
2	2	4	0	2
3	2	3	4	2
4	8	1	1	7
5	4	0	12	3
6	3	3	1	2
7	3	2	1	3
8	2	2	0	1
9	5	4	1	3
10	5	4	2	5
11	5	1	0	2
12	13	0	0	13
13	2	8	2	7
14	8	3	1	6
Total	71	46	26	59

6.4.1.1   Notes Impact Reported Problems

A strong relationship between the number of Notes used and the number of problems reported was found (p=0.007 in the table of coefficients of regression (Table 6.12); Pearson correlation coefficient=0.740, p=0.002). This indicates that Notes have an important role when reporting usability problems.

The relation between Notes and problems suggests that Notes are used to report problems. In fact, in analyzing the content of a set of Notes and problems it was found that a large number of problems (80%) are associated with Notes. This suggests that

evaluators added Notes to Web pages and potentially used them to report problems in usability problem forms.

There were only 5 cases in which both Note contents were successfully saved and problem reports were submitted (HEAssistant v0.1 has a usability problem regarding the saving capability. See Appendix B for the description.). Figure 6.9 shows the proportion of problems associated with Notes by content.
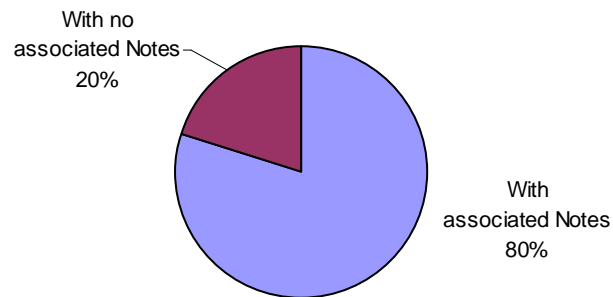
Figure 6.9. Proportion of Problems With/Without Associated Notes
(evaluators=5, problems=20)

Figures 6.10 and 6.11 show examples of Single and Compound problems with associated Notes, respectively.

Note

the three columns are a mess!

Problem

They can improve the design of the web page to make it [look] more [balanced]. [Some metaphor may be applied to the] three columns design in the middle of the page … to let [the] user feel more comfortable about what [to] focus on.
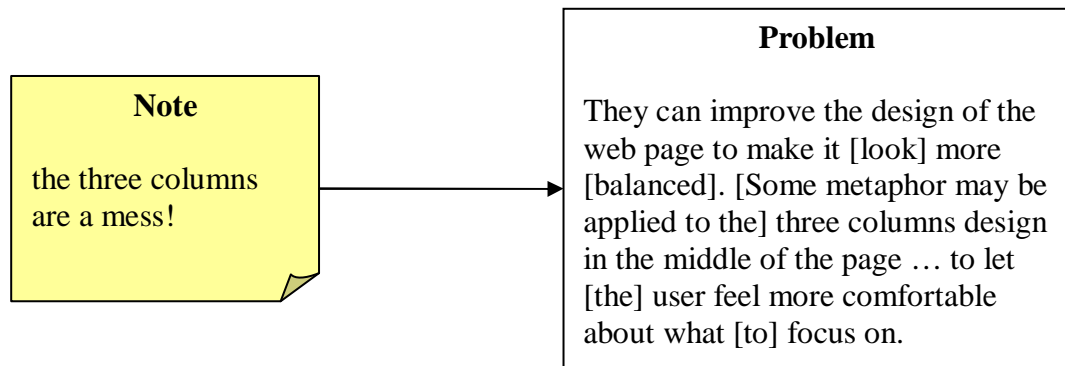
Figure 6.10. A Single Problem Associated with a Note

Problem

Overall too much empty space in the web page and too small the font is. The web page looks desultory and hard to read.

Note

Wasted space

Note

Wasted space causing fonts to be too small

Note
The order that these tabs[*] are following looks haphazard (maroon-green-maroon again?)
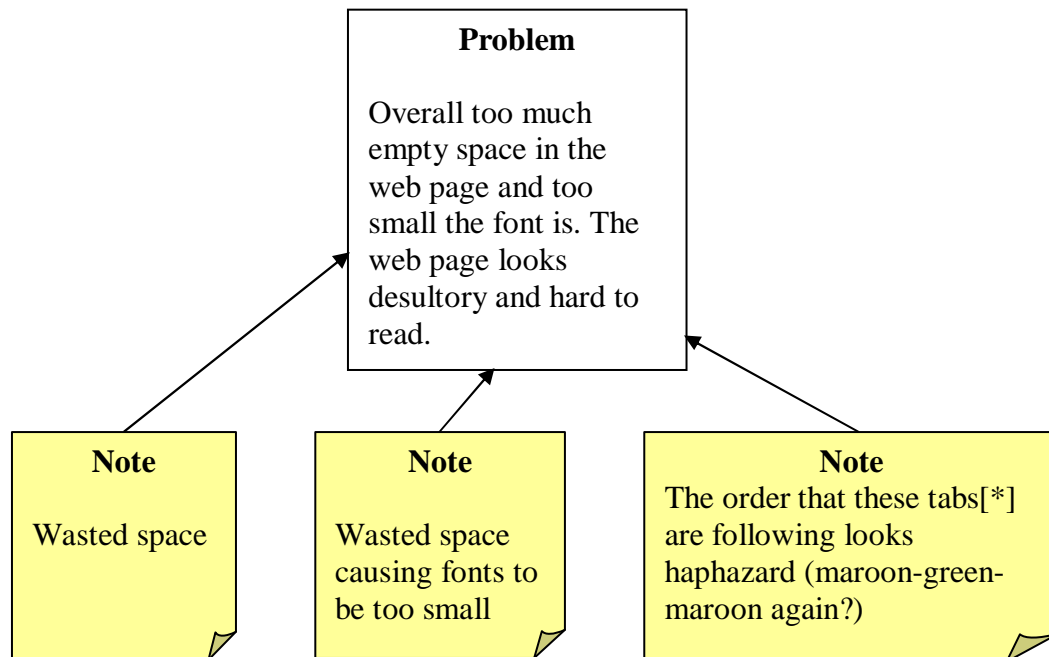
Figure 6.11. A Compound Problem Associated with 3 Notes.

*: "tabs" refer to headers of content boxes in the middle of the home page. Appendix D includes a home page instance.

6.4.1.2 Non-Textual Annotations have Little Impact on Reported Problems

No relation was found between Push Pins ($p=0.656$ in the table of coefficients of regression (Table 6.12); Pearson correlation coefficient=-0.142, $p=0.627$) or Question Marks ($p=0.944$ in the table of coefficients of regression (Table 6.12); Pearson correlation coefficient=-0.136, $p=0.642$) and reported problems. Push Pins and Question Marks support evaluators while inspecting Web interfaces. Evaluators add Push Pins and Question Marks to mark where problems are; however, it appears there is no direct use of them in the moment problems are reported. Since Notes have a great impact on problem reporting, what evaluators can do is to go back to non-textual annotations and add their rationales for adding such annotations. This may make information about problems explicit, visible, and ready to be used when reporting problems.

By adding comments about problems to non-textual annotations, it may increase the number of problems that are reported. Indeed, a significant number of Push Pins (41.3%) and Question Marks (50%) appeared isolated in annotated Web pages. This suggests that some problems might have been left unreported.

Figures 6.12 and 6.13 depict different ways Push Pins and Question Marks were found in annotated Web pages submitted by the 14 evaluators in Table 6.13, respectively.
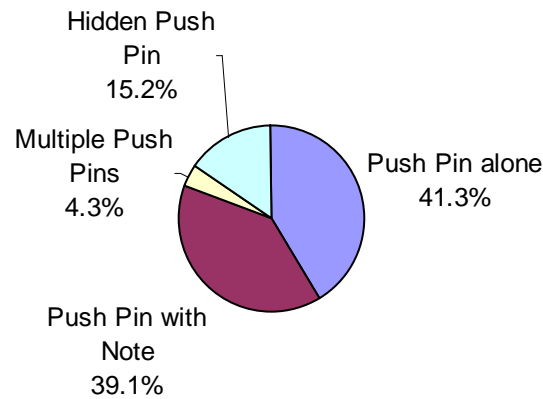
Figure 6.12. Ways Push Pins Appeared in Annotated Web Pages.

Push Pins were found isolated, accompanied by a Note or another Push Pin, and hidden by a Note (evaluators=14, n=46)
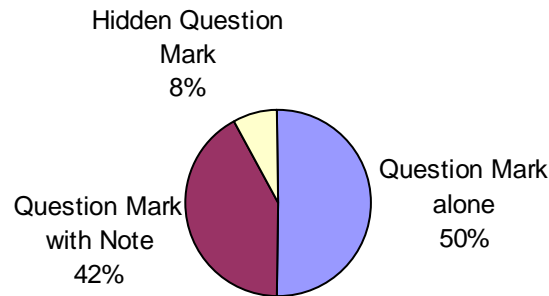
Figure 6.13. Ways Question Marks Appeared in Annotated Web Pages.

Question Marks were found isolated, accompanied by a Note, and hidden by a Note (evaluators=14, n=26)

6.4.2   Outlined Recommendations

Among the three types of annotations used in this study, Notes were found to be of most use when reporting problems. Non-textual annotations such as Push Pins and

Question Marks were found to have little impact in problem reporting. To increase the use of non-textual annotations in problem reporting and reduce the number of unreported problems, evaluators are recommended to revisit these elements and comment on their rationales for annotating them. In this way, information about related problems is made explicit and ready to be used when problems are reported.

In summary, when using annotation to support inspection evaluators are recommended to perform annotation of interfaces in two passes:

- Pass 1: Evaluators freely annotate the interface while inspecting it.

- Pass 2: After inspection, evaluators are recommended to go back to non-textual annotations and add Notes with their rationales for annotating such elements whenever it is appropriate.

6.5   Tool User Satisfaction

A considerable number of evaluators were overall satisfied with HEAssistant v0.1. Evaluators were asked if they would recommend the tool to somebody else and if they would use it again for a Heuristic Evaluation project (see questionnaire in Appendix C). Seventy-two percent (72.7%) of evaluators reported they would both recommend and use the tool.

There is more work that needs to be done to extend HEAssistant v0.1. As mentioned in the Discussion section, the annotation type set in HEAssistant needs to be extended. We need to support other activities in inspection such as interface observation and problem elaboration.

6.6   Conclusions

HEAssistant was evaluated and annotation usage in the context of inspection was studied in Study 2.

The use of annotations was studied. It was found that textual annotations ("Notes") were used to describe problems and non-textual annotations ("Push Pins" and "Question Marks") to mark problem features during inspection. However, textual annotations were found to have a higher use than non-textual annotations when reporting problems. This suggests that a number of problems are left unreported.   It is recommended that evaluators return to non-textual annotations after inspection and comment on them. This may make information about relevant problems explicit, visible, and ready to be used when reporting problems, increasing the number of problems that are reported. Recommendations are outlined in the section.

In addition, the majority of evaluators (72.7%) were found to be satisfied overall with the annotator.

7   CONCLUSIONS AND FUTURE WORK

This research was undertaken in order to develop a Heuristic Evaluation tool for inspection. A tool for inspection has been successfully developed and evaluated.

First, the Heuristic Evaluation inspection process was characterized to generate ideas for inspection tools. Second, tools for inspection were proposed. Third, software requirements for developing a Heuristic Evaluation suite were identified. Finally, a Web page annotator for inspection was developed and evaluated.

An observational study was conducted to dissect the inspection process. Seven evaluators were observed applying Heuristic Evaluation on a non-interactive, paper-based Web interface. The study shows that evaluators were involved in several activities: elaborating usability problems, observing, navigating, and annotating the Web interface.

Identified activities were used to visualize different types of tools for inspection. Tools for improving the evaluator's observational skills and annotation tools were devised. Magnifiers and Window views are proposed to highlight contextualized interface information hidden to the evaluator eye and that may lead to problem identification.

Software requirements (total 44) to build a Heuristic Evaluation suite were identified from the literature, the characterization of the inspection process, and experience. These include requirements for building tools for two phases in Heuristic Evaluation: inspection and usability problem preparation.

A Web page annotator for inspection was developed and evaluated. The annotator is a Mozilla Firefox extension with capabilities for adding textual ("Notes")

and non-textual ("Push Pins" and "Question Marks") annotations to Web pages. A survey where 22 evaluators used the annotator while applying a Heuristic Evaluation on an interactive Web interface was conducted to evaluate the tool and learn the role of annotations in inspection. Findings led to the proposal of recommendations for improving the use of annotations in problem reporting.

It was found that textual annotations are commonly used to describe problems during inspection. Seventy-two percent (72.7%) of evaluators reported to have used Notes in this way. Moreover, over ninety percent (93.3%) of a group of 45 Notes annotated by 10 evaluators were informal usability problem descriptions containing recommendations for fixing problems, unexpected results obtained from using interface features, observations of "bad" features, and questions pertaining to problem features.

Non-textual annotations are used to mark problem interface features when inspecting interfaces. A little over sixty-six percent (66.7%) of 21 evaluators responded to have used Push Pins to mark problem interface elements or areas. Among 19 evaluators, a group of evaluators (63.2%) reported to have used Question Marks for marking problem interface elements when drawing questions about them; in other cases .(57.9%), Question Marks were used when marking confusing interface features.

Usage of textual annotations in problem reporting is higher than that of non-textual annotations. The annotations and problem reports by 14 evaluators were analyzed to see if they were related. A strong relationship was found between the numbers of Notes ($p=0.007$ for the Hypothesis that the coefficient is 0 in the multiple-linear regression; Pearson correlation coefficient=0.740) and reported problems. In contrast,

there was no relationship between the numbers of Push Pins (p=0.656 for the Hypothesis that the coefficient is 0 in the multiple-linear regression; Pearson correlation coefficient=-0.142) and reported problems. Similarly, the number of Question Marks (p=0.944 for the Hypothesis that the coefficient is 0 in the multiple-linear regression; Pearson correlation coefficient=-0.136) were not related to the number of reported problems. This indicates that non-textual annotations are rarely used when problems are reported, meaning that a number of problems are left unreported. Based on these findings, a set of recommendations for using annotations and reducing the number of unreported problems were devised and outlined in this dissertation.

Finally, the majority of evaluators (72.7%) were satisfied overall with the annotator. Evaluators reported they would both recommend the tool to others and use it in future Heuristic Evaluation projects.

This research project has been successful in different ways. Three ways for supporting Heuristic Evaluation inspection have been proposed. The inspection process has been characterized. A Heuristic Evaluation tool for annotating Web pages has been developed. The use of annotation in the context of inspection has been studied. Lastly, the achieved tool satisfaction rate is an indicator that this area of research is a promising one.

7.1   Future Work

This research is in advancement in the support of Heuristic Evaluation; however, there are questions that are left unanswered. The top research questions to be explored in the near future follow:

☐ *Improve HEAssistant*. Annotation capabilities derived from HEAssistant evaluation and annotation software requirements included in Section 4 are the starting points for improving HEAssistant.

☐ *Develop more tools for inspection*. In this research Magnifiers and Window views have been proposed to support observation in inspection. However, there is work that needs to be done to explore these ideas. What is the effectiveness of using such tools? Is problem identification improved?

☐ *Develop a HEAssistant Suite*. More work is needed to build a series of tools for performing Heuristic Evaluations.

☐ *Research the impact interactivity and interface format have on inspection further*. The impact interactivity and interface form (i.e. paper vs. computer) have on inspection needs to be studied further. There is the need for determining a precise list of interface characteristics impacting inspection.

☐ *Dissect the inspection process even further*. The inspection process has been dissected into a number of activities. More work is needed to characterize mental processes behind these activities.

REFERENCES

ADDY AND ASSOCIATES, INC. 2009. *Dr. Watson*. http://watson.addy.com/.

ANDRE, T.S., BELZ, S.M., MCCREARY, F.A., AND HARTSON, H.R. 2000. Testing a Framework for Reliable Classification of Usability Problems. In *Proceedings of the Human Factors and Ergonomics Society 44th Annual Meeting*. Human Factors and Ergonomics Society, Santa Monica, CA. 1090–1094.

BAILEY, R.W., ALLAN, R.W., AND RAIELLO, P. 1992. Usability Testing vs. Heuristic Evaluation: A Head-to-Head Comparison. In *Proceedings of the Human Factors Society 36th Annual Meeting*. Human Factors Society, Santa Monica, CA. 409-413.

BALBO, S. 1995. Software Tools for Evaluating the Usability of User Interfaces. In *Proceedings of the 6th International Conference on Human-Computer Interaction*, Anzai, Y., and Ogawa, K., Eds. Elsevier, New York, NY. 337-342.

BASS, L., CLEMENTS, P., AND KAZMAN, R. 2003. *Software Architecture in Practice*, 2nd Ed. Addison-Wesley, Leading, MA.

BEVAN, N., BARNUM, C., COCKTON, G., NIELSEN, J., SPOOL, J., AND WIXON, D. 2003. The "Magic Number 5": Is it Enough for Web Testing? In *the CHI'03 Extended Abstracts on Human Factors in Computing Systems*, Cockton, G., and Korhonen, P., Eds. ACM, New York, NY. 698–699.

BRAJNIK, G. 2000. Automatic Web Usability Evaluation: What Needs to be Done? In *Proceedings of the 6th Conference on Human Factors and the Web*, Austin, Texas. http://www.tri.sbc.com/hfweb/brajnik/hfweb-brajnik.html.

CAST. 2004. *Bobby*. http://www.cast.org/products/Bobby/index.html.

CHAUDHARY, A. 2008. *Video Annotation Tools*. Master's thesis, Department of Computer Science, Texas A&M University, College Station, TX.

COCKTON, G., AND WOOLRYCH, A. 2001. Understanding Inspection Methods: Lessons from an Assessment of Heuristic Evaluation. In *People and Computers XV*, Blandford, A., Vanderdonckt, J., and Gray, P., Eds. Springer, Berlin, Germany. 171–192.

COCKTON, G., WOOLRYCH, A., HALL, L., AND HINDMARCH, M. 2003. Changing Analysts' Tunes: The Surprising Impact of a New Instrument for Usability Inspection

Method Assessment. In *People and Computers XVII*, O'Neill, E., Palanque, P., and Johnson, P., Eds. Springer, Berlin, Germany. 145–161.

COCKTON, G., WOOLRYCH, A., AND HINDMARCH, M. 2004. Reconditioned Merchandise: Extended Structured Formats in Usability Inspection. In *the CHI'04 Extended Abstracts on Human Factors in Computing Systems*, Dykstra-Erickson, E., and Tscheligi, M., Eds. ACM, New York, NY. 1433–1436.

COLLINS ENGLISH DICTIONARY. 2006. *Collins English Dictionary*, 2nd Ed. Harper Collins, Glasgow, United Kingdom.

COX, D.A. 1998. *Supporting Results Synthesis in Heuristic Evaluation*. Master's thesis, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada.

DEGANI, A. 1996. *Modeling Human-Machine Systems: On Modes, Error and Patterns of Interaction*. Ph.D. dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

DESURVIRE, H.W., KONDZIELA, J.M., AND ATWOOD, M.E. 1992. What is Gained and Lost When Using Evaluation Methods other than Empirical Testing. In *People and Computers VII*, Monk, A.F., Diaper, D., and Harrison, M.D., Eds. Cambridge University Press, New York, NY. 89–102.

DUMAS, J.S., MOLICH, M., AND JEFFRIES, R. 2004. Describing Usability Problems - Are We Sending the Right Message? *Interactions 11*, 4, 24–29.

DYKSTRA, D.J. 1993. *A Comparison of Heuristic Evaluation and Usability Testing: The Efficacy of a Domain-Specific Heuristic Checklist*. Ph.D. dissertation, Department of Industrial Engineering, Texas A&M University, College Station, TX.

FARADAY, P. 2000. Visually Critiquing Web Pages. In *Proceedings of the 6th Conference on Human Factors and the Web*, Austin, Texas. http://www.tri.sbc.com/hfweb/faraday/FARADAY.HTM.

FARENC, C., LIBERATI, V., AND BARTHET, M.F. 1996. Automatic Ergonomic Evaluation: What are the Limits? In *Proceedings of the 2nd International Workshop on Computer-Aided Design of User Interfaces, (CADUI'96)*, Vanderdonckt, J., Ed. Presses Universitaires de Namur, Namur, Belgium. 159–170.

FARENC, C., AND PALANQUE, P. 1999. A Generic Framework Based on Ergonomic Rules for Computer Aided Design of User Interface. In *Proceedings of the 3rd*

International Conference on Computer-Aided Design of User Interfaces, (CADUI'99), Vanderdonckt, J., and Puerta, A., Eds. Kluwer, Dordrecht, The Netherlands. 281–292.

FREEMAN, E., FREEMAN, E., BATES, B., AND SIERRA, K. 2004. *Head First Design Patterns*. O'Reilly Media, Inc., Sebastopol, CA.

FURNAS, G.W. 1999. The FISHEYE View: A New Look at Structured Files. In *Readings in Information Visualization: Using Vision to Think*, Card, S.K., Stuart, K., Mackinlay, J.D., and Shneiderman, B., Eds. Morgan Kaufmann, San Francisco, CA. 312–330.

GOMITA. 2009. *ScrapBook*. http://amb.vis.ne.jp/mozilla/scrapbook/.

GOOGLE. 2009. *Google's Image Search*. http://images.google.com/imghp?hl=en.

GROUPSYSTEMS. 2009. *GroupSystems*. http://www.groupsystems.com/.

IBM. 2008. *Rational ClearQuest*.
http://www.ibm.com/developerworks/rational/products/clearquest/.

INSTONE, K. 1997. *Site Usability Heuristics*. http://instone.org/heuristics/.

IVORY, M.Y. 2001. *An Empirical Foundation for Automated Web Interface Evaluation*. Ph.D. dissertation, Department of Computer Science, University of California, Berkeley, CA.

IVORY, M.Y., AND HEARST, M.A. 2001. The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys 33*, 4, 470–516.

IVORY, M.Y., AND HEARST, M.A. 2002. Improving Web Site Design. *IEEE Internet Computing 6*, 2, 56–63.

JEFFRIES, R., MILLER, J.R., WHARTON, C., AND UYEDA, K. 1991. User Interface Evaluation in the Real World: A Comparison of Four Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems, (CHI'91)*, Robertson, S.P., Olson, G.M., and Olson, J.S., Eds. ACM, New York, NY. 119–124.

KELLEY, J.F. 1983. An Empirical Methodology for Writing User-Friendly Natural Language Computer Applications. In *Proceedings of the SIGCHI Conference on*

Human Factors in Computer Systems, (CHI'83), Smith, R.N., Pew, R.W., and Janda, A., Eds. ACM, New York, NY. 193–196.

KUNIAVSKY, M. 2003. *Observing the User Experience: A Practitioner's Guide to User Research*. Morgan Kaufmann, San Francisco, CA. Chapter12.

LARMAN, C. 2004. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall, Upper Saddle River, NJ.

LAVERY, D., COCKTON, G., AND ATKINSON, M.P. 1996. *Heuristic Evaluation: Usability Evaluation Materials*. Technical report, TR-1996-15, Department of Computer Science, University of Glasgow, Glasgow, UK.

LOER, K., AND HARRISON, M. 2000. Formal Interactive Systems Analysis and Usability Inspection Methods: Two Incompatible Worlds? In *Proceedings of the Interactive Systems: Design, Specification, and Verification: 7th International Workshop, (DSV-IS 2000)*, Palanque, P.A., and Patern, F., Eds. Lecture Notes in Computer Science, vol. 1946, Springer, Berlin, Germany, 169–190.

LÖWGREN, J., AND NORDQVIST, T. 1992. Knowledge-Based Evaluation as Design Support for Graphical User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems, (CHI'92)*, Bauersfeld, P., Bennett, J., and Lynch, G., Eds. ACM, New York, NY. 181–188.

MACK, R.L., AND MONTANIZ, F. 1994. Observing, Predicting, and Analyzing Usability Problems. In *Usability Inspection Methods*, Nielsen, J., and Mack, R.L., Eds. Wiley, New York, NY. 295–339.

MAHAJAN, R., AND SHNEIDERMAN, B. 1997. Visual and Textual Consistency Checking Tools for Graphical User Interfaces. *IEEE Transactions on Software Engineering 23*, 11, 722–735.

MARSHALL, C.C. 1997. Annotation: From Paper Books to the Digital Library. In *Proceedings of the 2nd ACM International Conference on Digital Libraries, (DL'97)*. ACM, New York, NY. 131–140.

MARSHALL, C.C., AND BRUSH, A.J.B. 2004. Exploring the Relationship Between Personal and Public Annotations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, (JCDL'04)*, Chen, H., Wactlar, H.D., Chen, C., Lim, E., and Christel, M.G., Eds. ACM, New York, NY. 349–357.

MOLICH, R., AND DUMAS, J.S. 2007. Making Usability Recommendations Useful and Usable. *Journal of Usability Studies 2*, 4, 162–179.

MOLICH, R., AND DUMAS, J.S. 2008. Comparative Usability Evaluation (CUE-4). *Behaviour & Information Technology 27*, 3, 263–281.

MOLICH, R., EDE, M.R., KAASGAARD, K., AND KARYUKIN, B. 2004. Comparative Usability Evaluation. *Behaviour & Information Technology 23*, 1, 65–74.

MOZILLA. 2009a. *Firefox*. http://www.mozilla.com/en-US/firefox/.

MOZILLA. 2009b. *Multiple Item Package*. https://developer.mozilla.org/En/Multiple_Item_Packaging/.

MOZILLA. 2009c. *Profiles*. http://support.mozilla.com/en-US/kb/Profiles.

MOZILLA. 2009d. *XML User Interface Language (XUL)*. https://developer.mozilla.org/En/XUL/.

MULLER, M.J., MATHESON, L., PAGE, C., AND GALLUP, R. 1998. Methods & Tools: Participatory Heuristic Evaluation. *Interactions 5*, 5, 13–18.

NIELSEN, J. 1990. Paper versus Computer Implementations as Mockup Scenarios for Heuristic Evaluation. In *Proceedings of the 3rd IFIP International Conference on Human-Computer Interaction, (INTERACT'90)*, Diaper, D., Gilmore, D.J., Cockton, G., and Shackel, B., Eds. North-Holland, Amsterdam, The Netherlands. 315–320.

NIELSEN, J. 1992. Finding Usability Problems through Heuristic Evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems, (CHI'92)*, Bauersfeld, P., Bennett, J., and Lynch, G., Eds. ACM, New York, NY. 373–380.

NIELSEN, J. 1994a. *Usability Engineering*. Morgan Kaufmann, San Francisco, CA.

NIELSEN, J. 1994b. Enhancing the Explanatory Power of Usability Heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems, (CHI'94)*, Adelson, B., Dumais, S., and Olson, J.S., Eds. ACM, New York, NY. 152–158.

NIELSEN, J. 1995a. Scenarios in Discount Usability Engineering. In *Scenario-Based Design: Envisioning Work and Technology in System Development*, Carroll, J., Ed. Wiley, New York, NY. 59–83.

NIELSEN, J. 1995b. *Severity Ratings for Usability Problems*. http://www.useit.com/papers/heuristic/severityrating.html.

NIELSEN, J. 2005a. *How to Conduct a Heuristic Evaluation*. http://www.useit.com/papers/heuristic/heuristic_evaluation.html.

NIELSEN, J. 2005b. *Ten Usability Heuristics*. http://www.useit.com/papers/heuristic/heuristic_list.html.

NIELSEN, J., AND LANDAUER, T.K. 1993. A Mathematical Model of the Finding of Usability Problems. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computer Systems*, Ashlund, S., Mullet, K., Henderson, A., Hollnagel, E., and White, T., Eds. ACM, New York, NY. 206–213.

NIELSEN, J., AND MOLICH, R. 1990. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems, (CHI'90)*, Carrasco-Chew, J., and Whiteside, J., Eds. ACM, New York, NY. 249–256.

NORMAN, D.A. 2002. *The Design of Everyday Things*. Basic Books, New York, NY.

POOLE, M.S. 2004. A personal communication.

PREECE, J., SHARP, H., AND ROGERS, Y. 2002. *Interaction Design: Beyond Human-Computer Interaction*, 1st Ed. Wiley, New York, NY.

PROJECT GUTENBERG. 2009. *Project Gutenberg*. http://www.gutenberg.org/.

RIEMAN, J., DAVIES, S., HAIR, D.C., ESEMPLARE, M., POLSON, P., AND LEWIS, C. 1990. *An Automated Walkthrough: Description and Evaluation*. Technical report, 90-18, Institute of Cognitive Science, University of Colorado, Boulder, CO.

RIEMAN, J., DAVIES, S., HAIR, D.C., ESEMPLARE, M., POLSON, P., AND LEWIS, C. 1991. An Automated Cognitive Walkthrough. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems, (CHI'91)*, Robertson, S.P., Olson, G.M., and Olson, J.S., Eds. ACM, New York, NY. 427–428.

ROWAN, M., GREGOR, P., SLOAN, D., AND BOOTH, P. 2000. Evaluating Web Resources for Disability Access. In *Proceedings of the 4th International Conference on Assistive Technologies, (ASSETS'00)*, Tremaine, M., Cole, E., and Mynatt, E.D., Eds. ACM, New York, NY. 80–84.

SCHOLTZ, J., LASKOWSKI, S., AND DOWNEY, L. 1998. Developing Usability Tools and Techniques for Designing and Testing Web Sites. In *Proceedings of the 4th Conference on Human Factors and the Web*, Basking Ridge, New Jersey. http://www.research.att.com/conf/hfweb/proceedings/scholtz/index.html.

SCHÖN, D.A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York, NY.

SEARS, A. 1995. AIDE: A Step Toward Metric-based Interface Development Tools. In *Proceedings of the 8th ACM Symposium on User Interface Software and Technology, (UIST'95)*, Robertson, G.G., Ed. ACM, New York, NY. 101–110.

SLAVKOVIC, A., AND CROSS, K. 1999. Novice Heuristic Evaluations of a Complex Interface. In *the CHI'99 Extended Abstracts on Human Factors in Computing Systems*, Altom, M.W., and Williams, M.G., Eds. ACM, New York, NY. 304-305.

SNYDER, C. 2003. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann, San Francisco, CA.

TAN, W-S., LIU, D., AND BISHU, R. 2009. Web Evaluation: Heuristic Evaluation vs. User Testing. *International Journal of Industrial Ergonomics 39*, 4, 621-627.

TEXAS A&M UNIVERSITY TRANSPORTATION SERVICES. 2009. *Texas A&M University Transportation Services*. http://transport.tamu.edu/.

THENG, Y.L. AND THIMBLEBY, H. 1998. HyperAT: Addressing Usability Issues in Web Authoring. In *Proceedings of World Conference on the WWW and Internet & Intranet, (WebNet'98)*, Maurer, H.A., and Olson, R.G., Eds. AACE, Charlottesville, VA.

USABLENET. 2002. *LIFT Online and LIFT Onsite*. http://www.usablenet.com/.

VALTECH, INC. 2000. *Chapter 11: Designing with the GRASP Patterns*. A presentation.

VIRZI, R.A. 1989. What Can You Learn from a Low-Fidelity Prototype? In *Proceedings of the Human Factors Society 33rd Annual Meeting*. Human Factors Society, Santa Monica, CA. 224–228.

WATCHFIRE. 2007. *WebXACT*. http://webxact.watchfire.com/.

WHARTON, C., RIEMAN, J., LEWIS C., AND POLSON, P. 1994. The Cognitive Walkthrough Method: A Practitioner's Guide. In *Usability Inspection Methods*, Nielsen, J., and Mack, R.L., Eds. Wiley, New York, NY. 105–140.

WOOD, J. 2004. *Usability Heuristics Explained*.
http://iqcontent.com/publications/features/article_32/.

WOOLRYCH, A. 2001. *Assessing the Scope and Accuracy of the Usability Inspection Method Heuristic Evaluation*. M.Phil.'s thesis, School of Computing and Technology, University of Sunderland, Sunderland, UK.

ZEN CART. 2006. *Zen Cart*. http://www.zen-cart.com/.

ZHANG, Z., BASILI, V., AND SHNEIDERMAN, B. 1999. Perspective-based Usability Inspection: An Empirical Validation of Efficacy. *Empirical Software Engineering 4*, 1, 43–69.

APPENDIX A

STUDY 1 INSTRUMENTS

A.1   Phase 1 Instruments

A.1.1   Background Questionnaire

Background Questionnaire

This questionnaire is to help us understand your background and experience.

1. General

01: What age group are you in?

Please choose **only one** of the following:

☐ Under 20 years

☐ 20 - 29 years

☐ 30 - 39 years

☐ 40 - 49 years

☐ 50 - 59 years

☐ Over 59 years

02: Are you a student?

Please choose **only one** of the following:

☐ Yes

☐ No

[Only answer this question if you answered 'Yes' to question '02 ']

02a: What is your academic program?

Please choose **only one** of the following:

☐ Bachelors

☐ Masters

☐ Doctorate

☐ Postdoctorate

☐ Other [_____]

[Only answer this question if you answered 'No' to question '02 ']

02b: What is the highest level of education you have completed?

Please choose **only one** of the following:

☐ High school

☐ Bachelors

☐ Masters

☐ Doctorate

☐ Postdoctorate

☐ Other [＿＿＿＿＿＿＿]

02c: Which is your major area of study?

Please choose **only one** of the following:

☐ Computer Science

☐ Computer Engineering

☐ Management Information Systems

☐ Other [＿＿＿＿＿＿＿]

2. Computer and Web Experience

03: For how many years have you used computers?

Please choose **only one** of the following:

☐ None

☐ 1- 2 years

☐ 3 - 6 years

☐ 7 - 10 years

☐ 11 - 14 years

☐ Over 14 years

04: How many hours do you use computers every week?

Please choose **only one** of the following:

- ☐ None
- ☐ 1 - 4 hours
- ☐ 5 - 10 hours
- ☐ 11 - 20 hours
- ☐ 21 - 40 hours
- ☐ Over 40 hours

05: For how many years have you known about the Web?

Please choose **only one** of the following:

- ☐ None
- ☐ 1 - 2 years
- ☐ 3 - 5 years
- ☐ 6 - 8 years
- ☐ Over 8 years

06: How many hours do you access the Web every week?

Please choose **only one** of the following:

- ☐ None
- ☐ 1 - 4 hours
- ☐ 5 - 10 hours
- ☐ 11 - 15 hours
- ☐ 16 - 20 hours
- ☐ Over 20 hours

07: How long have you authored Web pages or develop Web applications?

Please choose **only one** of the following:

☐ None

☐ 1 year

☐ 2 - 3 years

☐ 4 - 5 years

☐ 6 - 8 years

☐ Over 8 years

[Only answer this question if you answered '1 year' or '2 - 3 years' or '4 - 5 years' or '6 - 8 years' or 'Over 8 years' to question '07 ']

08: How often do you authored Web pages or develop Web applications?

Please choose **only one** of the following:

☐ Less than monthly

☐ Monthly

☐ Weekly

☐ Daily

09: Please rate your programming expertise on each of the following Web technologies (1=None, 5=Expert)

Please choose the appropriate response for each item:

	1	2	3	4	5
HTML	☐	☐	☐	☐	☐
CSS	☐	☐	☐	☐	☐
Javascript	☐	☐	☐	☐	☐
Java Servlets	☐	☐	☐	☐	☐
CGI scripts in C++	☐	☐	☐	☐	☐
Java Applets	☐	☐	☐	☐	☐
ASP	☐	☐	☐	☐	☐
PHP	☐	☐	☐	☐	☐

3. Web Evaluation

10: Do you know Heuristic Evaluation?

> <u>Please choose **only one** of the following:</u>
>
> ☐ Yes
>
> ☐ No

[Only answer this question if you answered 'Yes' to question '10 ']

10a: How many times have you applied it in projects?

> <u>Please choose **only one** of the following:</u>
>
> ☐ 0 times
>
> ☐ 1 - 2 times
>
> ☐ 3 - 5 times
>
> ☐ More than 5 times

11: Have you used Web accessibility automated evaluation tools before? (e.g. Bobby, LIFT, A-Prompt, etc.)

> <u>Please choose **only one** of the following:</u>
>
> ☐ Yes
>
> ☐ No
>
> ☐ I don't know

---

4. Usability

Please answer the following 10 questions about usability.

Question 01

12: Which is the recommended way to link to a page about news?

> <u>Please choose **only one** of the following:</u>
>
> ☐ <u>News</u> lists all our recent articles. (Answer)
>
> ☐ Go to our <u>News page</u> to see all our recent articles.
>
> ☐ <u>Click here</u> to see all our recent articles.
>
> ☐ To check out our News page <u>click here</u>.
>
> ☐ I don't know

Question 02

13: Why should red text not be used on blue background? Because ...
(question adapted from [1])

<u>Please choose **only one** of the following:</u>

☐ Not really, it depends on the design.

☐ It is fuzzy to read. (Answer)

☐ It is not an aesthetic-pleasing color combination.

☐ It is used greatly in advertisements.

☐ I don't know

Question 03

14: What is the best type of help? (question adapted from [2])

<u>Please choose **only one** of the following:</u>

☐ Online documentation

☐ Context-sensitive help

☐ Tutorials

☐ Needing no help at all (Answer)

☐ I don't know

Question 04

15: All are recommended testing practices EXCEPT:

<u>Please choose **only one** of the following:</u>

☐ Testing through out multiple iterations

☐ Testing competitor's designs

☐ Testing with detailed screen designs

☐ Testing until you have a working prototype (Answer)

☐ I don't know

Question 05

16: All are recommended ways to refer to people in a Web site EXCEPT:

Please choose **only one** of the following:

☐ **Customer** as in: New Customer? Create an Account

☐ **Member** as in: New Member? Create an Account

☐ **Guest** as in: Hello, Guest

☐ **User** as in: New User? Create an Account (Answer)

☐ I don't know

Question 06

17: What is the MAIN purpose of a Web site's home page?

Please choose **only one** of the following:

☐ Provide navigation aids to get to contents

☐ Show a logo

☐ Convey what the site is about (Answer)

☐ Provide short-cuts to most frequently searched content

☐ I don't know

Question 07

18: Which is the best design to navigate through a collection of items?

Please choose **only one** of the following:

☐ Previous 10 items   Next 10 items (Answer)

☐ Previous 10 items , Next 10 items

☐ Previous 10 items | Next 10 items

☐ [Previous 10 items] [Next 10 items]

☐ I don't know

Question 08

19: Which is the best predictor of software usability? (question adapted from [2])

Please choose **only one** of the following:

☐ Self-evidence (Answer)

☐ Consistency

☐ Effectiveness

☐ Efficiency

☐ I don't know

Question 09

20: Which is the recommended way to write a title in a Web page?

Please choose **only one** of the following:

☐ **THIS IS A TITLE**

☐ **This is a title** (Answer)

☐ **This Is A Title**

☐ **This is a Title**

☐ **I don't know**

Question 10

21: Which is the best ally of learnability in an application? Having ...

Please choose **only one** of the following:

☐ Tutorials

☐ "Undo" capabilities (Answer)

☐ Online help

☐ Wizards

☐ I don't know

Acknowledgements

References

[1] Human Factors International, "Web Usability Quiz," URL: http://www.humanfactors.com/training/webquiz.asp#, as of 7/9/06
[2] Human Factors International, "Usability Quiz," URL: http://www.humanfactors.com/downloads/usabilityquiz.asp, as of 7/9/06.
[3] PHPSurveyor, URL: http://www.phpsurveyor.org, as of 12/1/06.
[4] LimeSurvey, URL: http://www.limesurvey.org, as of 7/14/09.

Heuristic Evaluation Booklet

This document describes Heuristic Evaluation and how to conduct it. It also describes the form for reporting usability problems.

1. Heuristic Evaluation

Heuristic Evaluation was proposed by Nielsen and Molich in 1990[1]. It is a simple method used to evaluate find usability problems in user interfaces.

It consists in "having a small set of evaluators examine the interface and judge its compliance with recognized usability principles ('the heuristics')[2]."

The term "**heuristic**" here refers to *general rules of thumb,* which describe common characteristics that well-designed user interfaces have. For instance, the heuristic: "Help users recognize, diagnose and recover from errors.[3]" More heuristics are cited in Section 5.

In summary, in Heuristic Evaluation we do the following:
- Carefully examine the interface, inspecting the different interface elements and the interface as a whole.
- Compare the observed interface characteristics against the heuristics.
- Report usability problems when finding violations to the heuristics (and other usability problems, design guidelines, etc.)

2. Overall Process

In a Heuristic Evaluation evaluators independently inspect the interface and meet to aggregate problems in a single report. In this collaborative effort, problem duplicates are discarded, solutions are recommended, and problems are prioritized. This report is then delivered to the development team to decide the best strategy to fix the usability problems.

Figure 1. Heuristic Evaluation overall process

3. A Typical Session

A typical session may last 1 or 2 hours. Nielsen recommends giving at least two passes to the interface:

> "The first pass would be intended to get a feel for the flow of the interaction and the general scope of the system. The second pass then allows the evaluator to focus on specific interface elements while knowing how they fit into the larger whole.[2]"

You can approach your inspection in different ways. Cockton et al.[4] mention 4 types of strategies one can follow:

1. *System Scanning*. This consists in browsing the interface without following any particular approach.
2. *System Searching*. This consists in following some kind of strategy such as focusing in certain interface elements such as inspecting menu options and group of navigation links.
3. *Goal Playing*. It consists in setting up a goal and trying to achieve it. For example, choosing goals such as searching for a particular piece of information, browsing a photo gallery, and buying a gift for a friend.
4. *Method Following*: This is similar to Goal Playing, but in addition to setting up a goal you define a specific step-by-step procedure to achieve such goal. For instance, in inspecting a word processor, you may setup a procedure to insert a picture into a document from a a) Clip-Art collection or b) file.

4. Usability Problem Definition

We will be reporting usability problems found in Web site(s). But, what is a usability problem? Cockton and colleagues[5] define a usability problem precisely:

> A "usability problem can be described as a feature or element of the interface that by its design, implication, or use may cause the user various degrees of difficulty in progressing or completion of a particular task."

5. Ten Usability Heuristics by Nielsen

In Heuristic Evaluation evaluators inspect an interface with heuristics in mind. The most commonly used heuristics are Jakob Nielsen's ten heuristics. Please read both of the following Web pages about the heuristics:

- Nielsen's[3] 10 usability heuristics:
  http://www.useit.com/papers/heuristic/heuristic_list.html
- John Wood's[6] explanation of the heuristics, but for the Web:
  http://iqcontent.com/publications/features/article_32/ (Note: If you have trouble opening this article from this document, copy the link and paste it into your browser.)

6. Usability Problem Form

This section describes a form we will be using to document usability problems. For each problem you will be describing the problem, how it was found, and which heuristic(s) was been violated. The Appendix includes the form.

a. Problem Description

In this part, describe the problem briefly. For example, when evaluating Hotmail´s (www.hotmail.com) e-mail Web-based application a usability problem may be described as follows:

> "Sender's name is erroneously linked. To open a message from the list, I need to click on the sender's name and not on the message's subject. The message's subject is not 'clickable'."

Figure 2. Sender's name is erroneously linked

b. Discovery Method

Specify which discovery method better describes the way you found the problem. There are four main method categories[4]: *System Scanning, System Searching, Goal Playing,* and *Method Following.* See page 3 for their description.

In our problem example:

> **System Scanning** is the discovery method that better describes the way the problem was found. I was scanning through the list of messages.

c. Involved Steps

This part is probably the most important part of this research. We are very interested in learning how problems are discovered. We would like you to describe in detail the different steps involved in discovering problems. There is no space limit here.

You can mention the tools that were used and how they were used. Tell us what you tried, the obtained system responses, your inputs, as many details as you can. If you don't know whether to put something or not, you are encouraged to do so.

In our previous example, we might describe the involved steps as follows:

> "I was just scanning through the message list. When trying to open a message, I noticed that the sender's name was a link instead of the message's subject. I would expect the Subject to be a link since it is the one describing the message."

d. Violated heuristic(s)

The problem may have violated more than one heuristic. Indicate each of them.

In our example, two heuristics have been violated:

> "Heuristic 2 (Match between system and the real world) is violated because real world conventions are not followed. The user clicks on the sender's name and gets a message. Instead, it makes more sense to click on the message subject and open the message which is described by that text."

> "Heuristic 4 (Consistency and standards) is violated because there is no consistency between the link (sender's name) and its destination (message content)."

References

[1]: Nielsen, J., and Molich, R. (1990). Heuristic Evaluation of User Interfaces, *Proc. ACM CHI'90 Conference*, pp. 249-256.

[2]: Nielsen, J. (2005). How to Conduct a Heuristic Evaluation, URL: http://www.useit.com/papers/heuristic/heuristic_evaluation.html, as of July 17, 2005.

[3]: Nielsen, J. (2005). Ten Usability Heuristics, URL: http://www.useit.com/papers/heuristic/heuristic_list.html, as of July 17, 2005.

[4]: Cockton, G., Woolrych, A., Hall, L., and Hindmarch, M. (2003). Changing Analysts' Tunes: The Surprising Impact of a New Instrument for Usability Inspection Method Assessment. *People and Computers XVII: Designing for Society-Human Computer Interaction 2003*, O'Neill, E., Palanque, P., and Johnson, P. (Editors), Springer, October, pp. 145-161.

[5]: Cockton, G., Woolrych, A., Hall, L., and Hindmarch, M. J. (2003). Important Notes for Heuristic Evaluation and Problem Report, URL: http://osiris.sunderland.ac.uk/~cs0mho/hci/hciSyllabus.html, as of December 1, 2003.

[6]: Wood, J. (2004). Usability Heuristics Explained, URL: http://iqcontent.com/publications/features/article_32/ (Note: If you have trouble opening this article from this document, copy the link and paste it into your browser.)

Appendix

The usability problem form:

Problem #: _____

Description:

What were you doing when discovering the problem? Choose a method below:
- ☐ System Scanning
- ☐ System Searching
- ☐ Goal Playing
- ☐ Method Following
- ☐ Other

Describe the method and steps involved

Violated Heuristic(s)

Which heuristic(s) does this problem violate? Circle the number(s) that apply.

1	Visibility of system status	6	Recognition rather than recall
2	Match between system and the real world	7	Flexibility and efficiency of use
3	User control and freedom	8	Aesthetic and minimalist design
4	Consistency and standards	9	Help users recognize, diagnose, and recover from errors
5	Error prevention	10	Help and documentation

A.1.3   Exercise Instructions

The Exercise

The purpose of this exercise is to practice conducting Heuristic Evaluation. You will be evaluating the usability of a Web site. In Heuristic Evaluation we carefully inspect the Web site and check against heuristics to find usability problems.

- Spend 40 minutes on the exercise. You may take breaks as needed. Please keep track of time by yourself.
- This research is about learning how Heuristic Evaluation is conducted, and part of it is to learn more about how problems are found. It is then very important to provide as many details as possible about problems.
- There is no specific number of usability problems to find, but try to fully document as many as you can.

Recording usability problems

Please use a usability problem form per problem. These file(s) will be submitted through this Web site after completing this exercise.

Getting started

- You will evaluate the Gutenberg Web site.
- Spend a couple of minutes to get an overall impression of the Web site.
- You may concentrate on evaluating a single page or part of the Web site.
- You may follow any or all of the discovery methods described in the training materials. Here are some examples:
    - **System Scanning**: Browsing the Web site without following any particular approach.
    - **System Searching**: Inspecting the Web site with some sort of structure; for example, specifically inspecting navigation aids.
    - **Goal Playing**: Setting up a goal and try to achieve it. For example, for the Gutenberg Web site a goal would be to:
        - Look for the book called "The Little Prince" and get its details.
    - **Method Following**: Similar to Goal Playing, but a step-by-step procedure is established. For example, we can establish the following procedure for finding the above book:
        1. Investigate the author of the book.
        2. Search the book by author.
        3. Get book details.

In summary, your task is:

Task: Perform a Heuristic Evaluation on the following Web site:
http://www.gutenberg.org (This will open a new window.)

Duration: 40 minutes
What to submit?: As many fully-documented usability problems as you can.
How to record problems?: Use usability problem forms

A.1.4   Task Instructions

Your Task

You will be performing a Heuristic Evaluation on **eXOP**, an online shop.

As mentioned, in Heuristic Evaluation we carefully inspect the Web site and check against heuristics to find usability problems.

As in the exercise, you can start by spending a couple of minutes to get an overall impression of the Web site, then go by following any or all of the discovery methods: **System Scanning**, **System Searching**, **Goal Playing**, and **Method Following**.

Details

- Evaluate the "**Product Page**".
- You may visit other pages in the Web site, but mainly focus on the Product Page.
- There is no maximum number of usability problems you need to report, but try to find as many as you can.
- Please work on this task for **20 minutes.** The Tester will let you know when time has elapsed.

   *You can evaluate the **Product Page** now.*

A.2   Phase 2 Instruments

A.2.1   The Static Web Interface: eXOP Web Pages

The following elements of the Static interface are included:

- WP1. Home page

- WP2. "DVD Movies" category page

- WP3. "Speed 2: Cruise Control" (DVD movie) product page

- WP3.1. "Larger image" page

- WP3.2. "Shopping Cart" page

- WP3.3. "Sign in" page

- SB. Storyboard

A.2.1.1 Web Page 1: "eXOP" Home Page

A.2.1.1.1   Web Page 1: "eXOP" Home Page – Top

A.2.1.1.2   Web Page 1: "eXOP" Home Page – Bottom

A.2.1.2   Web Page 2: "DVD Movies" Category Page

A.2.1.2.1   Web Page 2: "DVD Movies" Category Page - Top

A.2.1.2.2   Web Page 2: "DVD Movies" Category Page - Bottom

A.2.1.3   Web Page 3: "Speed 2: Cruise Control" Product Page

A.2.1.3.1 Web Page 3: "Speed 2: Cruise Control" Product Page - Top

A.2.1.3.2   Web Page 3: "Speed 2: Cruise Control" Product Page - Bottom

A.2.1.4   Web Page 3.1: "Larger Image" Page

A.2.1.5   Web Page 3.2: "Shopping Cart" Page

A.2.1.5.1   Web Page 3.2: "Shopping Cart" Page - Top

A.2.1.5.2 Web Page 3.2: "Shopping Cart" Page - Bottom

A.2.1.6   Web Page 3.3: "Sign In" Page

A.2.1.6.1   Web Page 3.3: "Sign In" Page - Top

A.2.1.6.2   Web Page 3.3: "Sign In" Page - Bottom

A.2.1.6   Storyboard

1. Home page
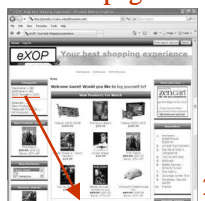
2. DVD Movies
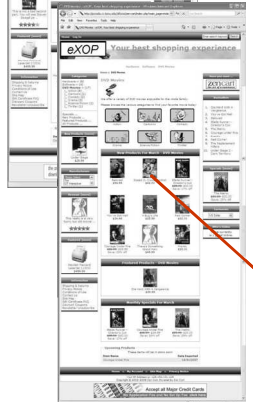
3. Speed 2: Cruise Control

3.1. Larger image

3.2. Shopping Cart

3.3. Sign in

APPENDIX B

HEASSISTANT SAVING USABILITY PROBLEM

HEAssistant v0.1 has a usability problem when saving Note contents. Saving the contents of Notes is a two-step process instead of one. The "Hide header" button in Note objects should be pressed before the Save button in the Annotator toolbar (see Figure B.1). Note content editions will be lost if only the Save button is pressed.
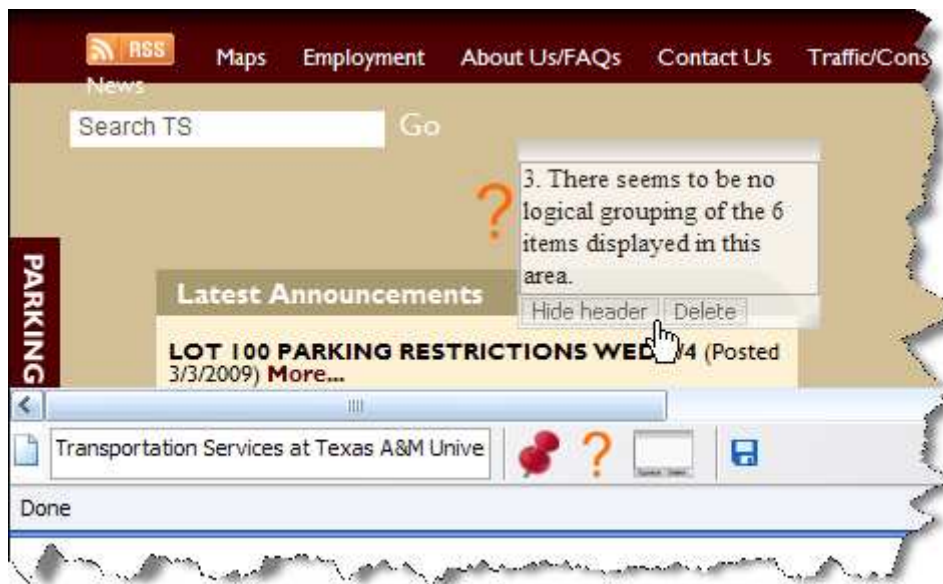
(a) Step 1. Hide Note Header
Figure B.1. Steps for Saving Notes in HEAssistant v0.1
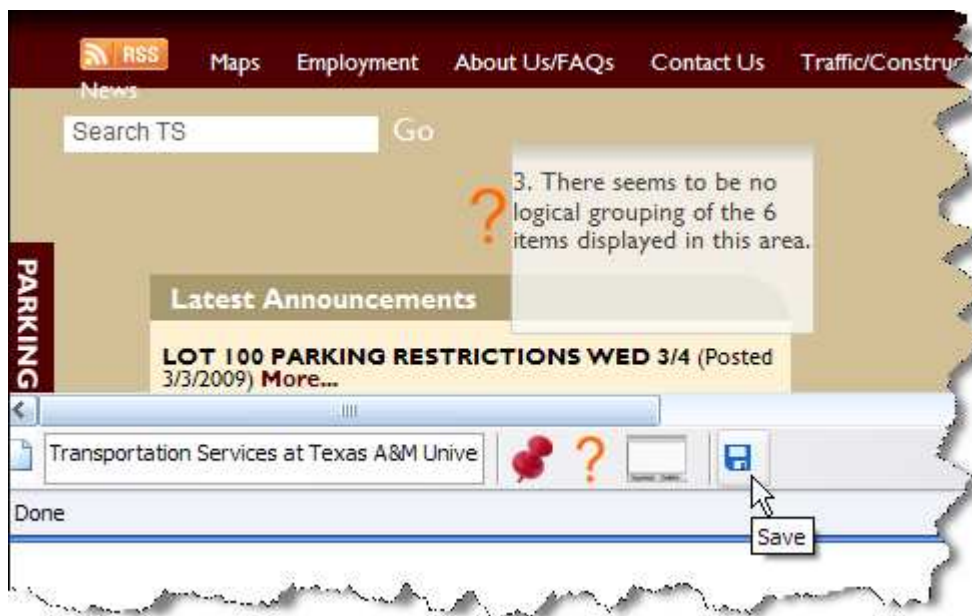
(b) Step 2. Press the Save Button in Annotator Toolbar
Figure B.1. continued

APPENDIX C

STUDY 2 INSTRUMENTS

C.1 Texas A&M University Transportation Services Web Pages

This section includes Web pages of the Web site as of February 3, 2009. The study was conducted in February and beginning of March, 2009. The site frequently gives news in the home page. The following Web pages are included:

- Home Page

- City and University Traffic Construction

- Parking Rules and Regulations

- Pay Citation

- Search Results

C.1.1 Transportation Services Home Page

C.1.1.1   Transportation Services Home Page – Top-Left

TEXAS A&M
UNIVERSITY

TEXAS A&M UNIVERSITY
TRANSPORTATION SERVICES
Moving Forward For You

Search TS [ ] Go

RSS   Maps   Employm[ent]

PARKING   TRANSIT   SPECIAL EVENTS

Latest Announcements

NEW - Lot 51 Pay-by-Space Visitor Parking (Posted 1/29/2009) More...

PARKING ADVISORY: Lot 75 (Posted 1/27/2009) More...

NEW - Lot 51 Pay-by-Space Visitor Parking (Posted 1/23/2009) More...

Added citation information on DPR site (Posted 1/23/2009) More...

REMINDER: Campus Permit Parking Rules (Posted 1/22/2009) More...

TRAFFIC ADVISORY: Ireland Street Closure (Posted 1/20/2009) More...

Aggieland Saturday

February 21, 2009

Aggieland Saturday guests may park free of charge in

Drivers Wanted: Are YOU Good Enough?

Transportation Services is now hiring. Hours are flexible and the training is free. Apply today! DRIVERS WANTED

Share a Ride To or From Campus

Transportation Services is proud to provide a FREE ride sharing service helping students, faculty and staff connect to save on gas and save the planet. Whether you're taking a one-time trip out of town or want to find a carpool buddy for a semester, sign up today! More...

C.1.1.2 Transportation Services Home Page – Top-Right

MY ACCOUNT

Sign In

ent   About Us/FAQs   Contact Us   Traffic/Construction   Committees/Forums   In the News

Visitors
Parking Regulations
Break & Summer
Night & Weekend
Retirees
Vendors
Disabled
RV
Ride Sharing
Permits
Citations
Pay Citation

Transit Regulations
On-Campus
Off-Campus
Break & Summer
Night & Weekend
Fall & Spring
Paratransit
Bus Advertising
Charters

Sporting Events
Conferences & Camps
Fun Run
Graduation

Sporting Event Parking & Traffic

Transportation Services has created parking plans for each sporting event held at Texas A&M University. Please see our Sports information to find out where you should park for the game.

Sporting Event Parking
Basketball Parking
Baseball Parking
Softball Parking

Parking Forums Announced

Faculty and Staff are invited to attend one of the following sessions with Transportation Services Executive Director Rod Weis. Question/answer sessions will follow the short presentation.

Tue **Dec 9:** 10-11:30am - Rudder 601
Wed **Jan 14:** 3-4:30pm - GSC Assembly A
Thu **Feb 5:** 11-12:30pm - Rudder 601

Parking Forum Presentation

C.1.1.3 Transportation Services Home Page – Bottom-Left

the following lots:

Lot	Near
Lot 50	Engineering
Lot 48	Rudder Auditorium
Lot 36	Vet School
Lot 61 & 74	Business and Agriculture

Permits are not required on Saturdays and spaces may be available in other lots; however, visitors may **NOT** park in reserved/restricted spaces which include Reserved Numbered, University Business, Service, Timed, Loading Spaces, and Residence Hall lots 30 and 40, which are 24 hour reserved.

Buses should park in Lot 74 on west campus near Rosenthal Meat Science Center.

There are also paid visitor parking areas if guests would prefer those facilities. The free and paid areas are highlighted on the map below.

Aggieland Saturday Parking Map
Aggieland Saturday Main Site

FLEET **BICYCLES**

C.1.1.4   Transportation Services Home Page – Bottom-Right

Move-In/Out

Leasing
Van Safety
Regulations
TAMU System Fleet
Vehicle Maintenance

Bike Regulations
State Laws
Theft & Safety
Bike Rack Locations

New Visitor Parking in Lot 51

Pay-by Space parking has been added to Lot 51 for visitors accessing the area of campus near Wisenbaker and Zachry buildings.

For more information see our Visitor Parking page.

250

ecial Events 979.862.6585   Fleet 979.845.5846

es | Fleet

University Contacts
bmaster | Contact Us

C.1.2 City and University Traffic Construction

C.1.2.1 City and University Traffic Construction – Top-left

TEXAS A&M UNIVERSITY

TEXAS A&M UNIVERSITY
TRANSPORTATION SERVICES
Moving Forward For You

Search TS [Go]

RSS   Maps   Employn

CITY AND UNIVERSITY CONSTRUCTION & TRAFFIC   RSS

Water Main Break

(Posted 1/27/2009)
PARKING ADVISORY
AREA AFFECTED: Lot 75, West Side
WHEN: 01/27/2009

SCOPE OF PROJECT:
We have been advised of a water main break in Lot 75. Physical Plant is working on the problem and should have in fixed by the end of today. It will take some time, however to resurface the area.

ALTERNATE PARKING:
Lot 75 permit holders will be allowed to park on a space available basis along the curbs in Lot 75 and Lot 36e.

MAP: Lot 75 and Lot 36e

Emerging Tech & Economic Develop Bldg Construc

(Posted 10/14/2008)
PARKING ADVISORY
AREA AFFECTED: Lot 50
WHEN: 10/14/2008 - 08/14/2009

SCOPE OF PROJECT:
University administrators have announced plans to begin construction in January 2009 of the Emerging Technologies and Economic Development Buildings (ETED) at the intersection of Bizzell and University Drive. This project will result in the closing of the northwest portion of Lot 50 and we will lose approximately 300 parking spaces. Because of this closure, we do not anticipate moving any additional customers into Lot 50 this semester and possibly through the end of this permit year. We intend to maintain the names that are currently on the

PARKING   **TRANSIT**   **SPECIAL EVENT**

C.1.2.2 City and University Traffic Construction – Top-Right

MY ACCOUNT
Sign In

About Us/FAQs   Contact Us   Traffic/Construction   Committees/Forums   In the News

Visitors
Parking Regulations
Break & Summer
Night & Weekend
Retirees
Vendors
Disabled
RV
Ride Sharing
Permits
Citations
Pay Citation

Transit Regulations
On-Campus
Off-Campus
Break & Summer
Night & Weekend
Fall & Spring
Paratransit
Bus Advertising
Charters

Sporting Events

Lot 51 Pay-by-Space Visitor Parking

(Posted 1/23/2009)
PARKING ADVISORY
AREA AFFECTED: Lot 51
WHEN: 01/26/2009

SCOPE OF PROJECT:
The newly installed Lot 51 pay-by-space equipment will be activated effective Monday, January 26, 2009. Lot 51 customers will see 54 visitor spaces at the west end of the lot designated and used for this project. Lot 51 permits are not valid in these spaces.

Lot 51 Pay-by-Space Visitor Parking

C.1.2.3 City and University Traffic Construction – Bottom-Left

Lot 50 wait list in the event construction is delayed and more permits can be sold. We wanted to make you aware of the reason that movement on this wait list has been suspended.

FLEET

BICYCLES

1250 TAMU • College Station, TX 77843-1250 • 9
Home | Parking | Transit | Special Events | Bicyc

Accessibility Policy | University Privacy and Security Policy
Glossary | Privacy Policy | Sitemap | FAQs | WebmasterWe

C.1.2.4 City and University Traffic Construction – Bottom-Right

Parking Lot 71 Changed Effective 8/16/08

(Posted 8/20/2008)

PARKING ADVISORY
AREA AFFECTED: Lot 71
WHEN: 08/16/2008 - 08/14/2009

SCOPE OF PROJECT:
Lot 71 changed effective 8/16/08- the parking area previously designated as 71b is now designated as 36f. Your permit is not valid in 36f during the day, but does authorize parking from 5:00 pm - 6:00 am according to night rules: night.aspx.html.

Lot 71 Map

Fun Run
Graduation
Move-In/Out

Leasing
Van Safety
Regulations
TAMU System Fleet
Vehicle Maintenance

Bike Regulations
State Laws
Theft & Safety
Bike Rack Locations

79.845.9700
es | Fleet

University Contacts
bmaster | Contact Us

C.1.3   Parking Rules and Regulations

TEXAS A&M UNIVERSITY

TRANSPORTATION SERVICES

Search TS   Go

Maps   Employment   About Us/FAQs   Contact Us   Traffic/Construction   Committees/Forums   In the News

PARKING RULES & REGULATIONS

The following rules and regulations were established by the University as a standard for the parking programs. Faculty, staff, students and visitors are urged to read these regulations. The Director of Transportation Services may exercise discretion to deal with special circumstances not covered by these regulations.

In developing these regulations, Transportation Services is guided by the Texas Motor Vehicle Laws and the Texas Education Code 51.202. The regulations apply to all vehicles on campus and are in effect at all times. Transportation Services, along with the University Police Department, has jurisdiction to enforce or impound any vehicle in violation of these regulations. The owner of the vehicle will be required to pay the cost of relocation and storing the vehicle.

Parking Regulations

- Texas A&M University Department of Transportation Services informs all students intending to purchase a permit to park or drive on campus that failure to register the vehicle in the state or to display a current and appropriate inspection certificate issued under Chapter 548, Transportation Code, may violate state law if the owner of the vehicle resides in this state.
- All vehicles are required to display a valid TAMU **parking permit** when parked on University property. **Proper placement**
- In the absence of a sign at the entrance to a lot, refer to the **parking map** to determine parking regulations for that lot. Any modifications to a lot during the academic year will be posted on signs at the entrance to the lot and the change will be effective immediately.
- When a parking sign conflicts with the parking map or information on the Web site, the sign will always supersede.
- Any area not specifically designated for parking is considered a no parking zone and vehicles will be cited appropriately.
- All vehicles must be parked within the marked boundaries of one parking space or stall.
- Vehicles that are parked parallel must be parked with the flow of traffic and not more than 18 inches from the curb.
- Vehicles parked in garages or in angled parking on one-way streets must be parked head-in.
- Permit holders should be aware that at certain times their space may not be reserved. These times include, but are not limited to, days when special events bring large numbers of visitors to campus such as home football games, graduation, move-in/move-out, Parent Weekend, as well as times when Transportation Services parking staff is not on duty. There is no provision for the permit for these exceptions.
- Transportation Services reserves the right to remove at the owner's expense any **public nuisance vehicle** or one that is not parked in accordance with the regulations contained in this Web site.
- Designated **handicap parking** spaces are for the exclusive use of vehicles displaying county issued handicap hangtags or disabled license plates AND a valid TAMU parking permit.
- The penalties for displaying an invalidated or altered permit, or a permit filed as lost or stolen are:
  - Immediate tow and impound of vehicle displaying an invalidated/forged/altered/lost/stolen item - with relocation and storage fees due to tow company
  - Citation with fine/time for displaying an invalidated/forged/altered/lost/stolen item
  - Theft of service fee of $40 per month, beginning the month the item was reported invalidated/forged/altered/lost/stolen
  - Confiscation of invalidated/forged/altered/lost/stolen item, with no refund
  - Revocation of TAMU parking privileges
  - Referral to Student Conflict Resolution Services or Department Head
- Faculty, staff and students are responsible for any citations that are issued to family members. Should a relative of TAMU faculty, staff or students receive parking citations, Transportation Services has the authority to bill the citations to the University affiliate.
- Vehicles of customers with three or more unpaid citations or with a $100.00 balance due may be towed and impounded at the owner's expense until the account is paid in full.
- Customers who present a check which is returned due to non-payment will be assessed a $25 returned check fee.
- **Bicycles** must be parked in the racks provided. Motorcycles, mopeds, and scooters must be parked in designated **motorcycle parking areas** and must not be parked in the parking garages or in any parking spaces striped or marked for automobiles or bicycles.
  University Bicycle Regulations

Other Parking Information

Restricted Spaces
Garage Parking
Visitor/Temporary
Vendor
Retiree
Special Events
Night & Weekend
Brook & Summer
Transit

1250 TAMU - College Station, TX 77843-1250 - 979.862.PARK (7275) - parking@tamu.edu/parking@tamu.edu
Home | Parking | Transit | Special Events | Bicycles | Fleet

Accessibility Policy | University Privacy and Security Policy | University Contacts
Glossary | Privacy Policy | Sitemap | FAQs | Webmaster/Webmaster | Contact Us

Side tabs: PARKING | TRANSIT | SPECIAL EVENTS | FLEET | BICYCLES

C.1.3.1 Parking Rules and Regulations – Top-Left

TEXAS A&M UNIVERSITY

TRANSPORTATION SERVICES
Moving Forward For You

Search TS [Go]

RSS

PARKING RULES & REGULATIONS

The following rules and regulations were established by the University as a standard for read these regulations. The Director of Transportation Services may exercise discretio

In developing these regulations, Transportation Services is guided by the Texas Motor to all vehicles on campus and are in effect at all times. Transportation Services, along impound any vehicle in violation of these regulations. The owner of the vehicle will be

Parking Regulations

- Texas A&M University Department of Transportation Services informs all students int the vehicle in this state or to display a current and appropriate inspection certificate i owner of the vehicle resides in this state.
- All vehicles are required to display a valid TAMU **parking permit** when parked on t
- In the absence of a sign at the entrance to a lot, refer to the **parking map** to determ academic year will be posted on signs at the entrance to the lot and the change will b
- When a parking sign conflicts with the parking map or information on this Web site,
- Any area not specifically designated for parking is considered a no parking zone and v
- All vehicles must be parked within the marked boundaries of one parking space or st
- Vehicles that are parked parallel must be parked with the flow of traffic and not more
- Vehicles parked in garages or in angled parking on one-way streets must be parked h
- Permit holders should be aware that at certain times their space may not be reserved large numbers of visitors to campus such as home football games, graduation, move-i parking staff is not on duty. There is no proration for the permit for these exceptions.
- Transportation Services reserves the right to remove at the owner's expense any **pub**

PARKING | TRANSIT | SPECIAL EVENTS | FLEET

C.1.3.2 Parking Rules and Regulations – Top-Right

MY ACCOUNT

Sign In

Maps Employment About Us/FAQs Contact Us Traffic/Construction Committees/Forums In the News

Visitors
Parking Regulations
Break & Summer
Night & Weekend
Retirees
Vendors
Disabled
RV
Ride Sharing
Permits
Citations
Pay Citation

Transit Regulations
On-Campus
Off-Campus
Break & Summer
Night & Weekend
Fall & Spring
Paratransit
Bus Advertising
Charters

Sporting Events
Conferences & Camps
Fun Run
Graduation
Move-In/Out

r the parking programs. Faculty, staff, students and visitors are urged to
n to deal with special circumstances not covered by these regulations.

Vehicle Laws and the Texas Education Code 51.202. The regulations apply
with the University Police Department, has jurisdiction to enforce or
required to pay the cost of relocation and storing the vehicle.

ending to purchase a permit to park or drive on campus that failure to register
ssued under Chapter 548, Transportation Code, may violate state law if the

University property. **Proper placement**

ine parking regulations for that lot. Any modifications to a lot during the
e effective immediately.

the sign will always supersede.

vehicles will be cited appropriately.

all.

than 18 inches from the curb.

ead-in.

. These times include, but are not limited to, days when special events bring
n/move-out, Parents Weekend, as well as times when Transportation Services

lic nuisance vehicle or one that is not parked in accordance with the

C.1.3.3 Parking Rules and Regulations – Bottom-Left

regulations contained in this Web site.

- Designated **handicap parking** spaces are for the exclusive use of vehicles displaying a TAMU parking permit.
- The penalties for displaying an invalidated or altered permit, or a permit filed as lost or
  - Immediate tow and impound of vehicle displaying an invalidated/forged/altered/lost/
  - Citation with $110 fine for displaying an invalidated/forged/altered/lost/stolen item
  - Theft of service fee of $40 per month, beginning the month the item was reported
  - Confiscation of invalidated/forged/altered/lost/stolen item with no refund
  - Revocation of TAMU parking privileges
  - Referral to Student Conflict Resolution Services or Department Head
- Faculty, staff and students are responsible for any citations that are issued to family m citations, Transportation Services has the authority to bill the citations to the Univers
- Vehicles of customers with three or more unpaid citations or with a $100.00 balance paid in full.
- Customers who present a check which is returned due to non-payment will be assess
- **Bicycles** must be parked in the racks provided. Motorcycles, mopeds, and scooters parked in the parking garages or in any parking spaces striped or marked for automo **University Bicycle Regulations**

Other Parking Information

Restricted Spaces
Garage Parking
Visitor/Temporary
Vendor
Retiree
Special Events
Night & Weekend
Break & Summer
Transit

BICYCLES

1250 TAMU • College Station, TX 77843-1250 • 979.8

Home | Parking | Transit | Sp

Accessibility Policy | University Privacy
Glossary | Privacy Policy | Sitemap | FAQ

C.1.3.4 Parking Rules and Regulations – Bottom-Right

Leasing
Van Safety
Regulations
TAMU System Fleet
Vehicle Maintenance

Bike Regulations
State Laws
Theft & Safety
Bike Rack Locations

county-issued handicap hangtags or disabled license plates AND a valid

or stolen are:

stolen item with relocation and storage fees due to tow company

invalidated/forged/altered/lost/stolen

embers: Should a relative of TAMU faculty, staff or students receive parking
ty's affiliate.

due may be towed and impounded at the owner's expense until the account is

ed a $25 returned check fee.

must be parked in designated **motorcycle parking areas** and must not be
biles or bicycles.

C.1.4   Pay Citation

C.1.4.1   Pay Citation – Left

TEXAS A&M
UNIVERSITY

TEXAS A&M UNIVERSITY
TRANSPORTATION SERVICES
Moving Forward For You

Search TS [   ] Go

SEARCH CITATION

PARKING

Please enter either a citation number or a license plate number to find the citation that is available for payment or appeal.

You may not find your citation in our system if you are issued a citation today. Tickets are normally loaded into the database the night after they are issued and may not always download over the weekend. You have 14 days to appeal the citation.

TRANSIT

License Plate: [   ]

State: [TEXAS ▼]

-OR-

Citation Number: [   ]

Where is the citation number?

SPECIAL EVENTS

[Submit]

BICYCLES

FLEET

1250 TAMU • College Station, TX 77843-1250 • (979)
Home | Parking | Transit | Spe

Accessibility Policy | University Privacy an
TSAC | Glossary | Privacy Policy | FAQs |

C.1.4.2   Pay Citation – Right

Maps   Employment   About Us/FAQs   Contact Us   Traffic/Construction   Committees/Forums   In the News

845-9700 • parking@tamu.eduparking@tamu.edu

ial Events | Bicycles | Fleet

d Security Policy | University Contacts

WebmasterWebmaster | Contact Us

C.1.5 Search Results

C.1.5.1   Search Results – Top

C.1.5.2 Search Results – Bottom

[pdf] EXPECTDELAYS COMMENCEMENT
... West Campus Garage 1 0 2 REED ARENA N 5/ 4/09 TEXAS A&M UNIVERSITY Transportation
Services 74 97 ... E X P E C T D E L A Y S DEPARTING REED ARENA COMMENCEMENT ...
transport.tamu.edu/specialevents/maps/Events/graduation/com.pdf - 2009-05-04 - Text Version

Goooooooooogle ►
Result Page 1 2 3 4 5 6 7 8 9 10 Next

Reed Arena [Search]

Search Within Results
Advanced Search
Search Tips

Powered by Google Search Appliance

· State of Texas - Texas Homeland Security - Statewide Search - State Link Policy - Sitemap ·

Webmaster
Legal Notices
Emergency Preparedness

C.2   Heuristic Evaluation Booklet

Heuristic Evaluation Booklet

This document describes Heuristic Evaluation and how to conduct it. It also describes the form for reporting usability problems.

1. Heuristic Evaluation

Heuristic Evaluation was proposed by Nielsen and Molich in 1990[1]. It is a simple method used to evaluate find usability problems in user interfaces.

It consists in "having a small set of evaluators examine the interface and judge its compliance with recognized usability principles ('the heuristics')[2]."

The term "**heuristic**" here refers to *general rules of thumb*, which describe common characteristics that well-designed user interfaces have. For instance, the heuristic: "Help users recognize, diagnose and recover from errors.[3]" More heuristics are cited in Section 5.

In summary, in Heuristic Evaluation we do the following:
- Carefully examine the interface, inspecting the different interface elements and the interface as a whole.
- Compare the observed interface characteristics against the heuristics.
- Report usability problems when finding violations to the heuristics (and other usability problems, design guidelines, etc.)

2. Overall Process

In a Heuristic Evaluation evaluators independently inspect the interface and meet to aggregate problems in a single report. In this collaborative effort, problem duplicates are discarded, solutions are recommended, and problems are prioritized. This report is then delivered to the development team to decide the best strategy to fix the usability problems.

Figure 1. Heuristic Evaluation overall process

3. A Typical Session

A typical session may last 1 or 2 hours. Nielsen recommends giving at least two passes to the interface:

> "The first pass would be intended to get a feel for the flow of the interaction and the general scope of the system. The second pass then allows the evaluator to focus on specific interface elements while knowing how they fit into the larger whole.[2]"

4. Usability Problem Definition

Usability problems that are identified are reported in Heuristic Evaluation, but what are usability problems? Cockton and colleagues[4] define usability problems precisely:

> A "usability problem can be described as a feature or element of the interface that by its design, implication, or use may cause the user various degrees of difficulty in progressing or completion of a particular task."

5. Ten Usability Heuristics by Nielsen

In Heuristic Evaluation evaluators inspect an interface with heuristics in mind. The most commonly used heuristics are Jakob Nielsen's ten heuristics. Please read both of the following Web pages about the heuristics:

- Nielsen's[3] 10 usability heuristics for general use:
  http://www.useit.com/papers/heuristic/heuristic_list.html
- John Wood's[5] explanation of the heuristics, but for the Web:
  http://iqcontent.com/publications/features/article_32/

6. Usability Problem Form

This section describes a general form to document usability problems. See the Appendix to find the form.

a. Problem Description

In this part, describe the problem briefly. For example, when evaluating Hotmail´s (www.hotmail.com) e-mail Web-based application a usability problem may be described as follows:

> "Sender's name is erroneously linked. To open a message from the list, I need to click on the sender's name and not on the message's subject. The message's subject is not 'clickable'."

Figure 2. Sender's name is erroneously linked

b. Violated heuristic(s)

The problem may have violated more than one heuristic. Indicate each of them.

In our example, two heuristics have been violated:

> "Heuristic 2 (Match between system and the real world) is violated because real world conventions are not followed. The user clicks on the sender's name and gets a message. Instead, it makes more sense to click on the message subject and open the message which is described by that text."

> "Heuristic 4 (Consistency and standards) is violated because there is no consistency between the link (sender's name) and its destination (message content)."

References

[1]: Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conference*, pp. 249-256.

[2]: Nielsen, J. (2005). How to Conduct a Heuristic Evaluation, URL: http://www.useit.com/papers/heuristic/heuristic_evaluation.html, as of July 17, 2005.

[3]: Nielsen, J. (2005). Ten Usability Heuristics, URL: http://www.useit.com/papers/heuristic/heuristic_list.html, as of July 17, 2005.

[4]: Cockton, G., Woolrych, A., Hall, L., and Hindmarch, M. J. (2003). Important notes for Heuristic Evaluation and Problem Report, URL: http://osiris.sunderland.ac.uk/~cs0mho/hci/hciSyllabus.html, as of December 1, 2003.

[5]: Wood, J. (2004). Usability heuristics explained, URL: http://iqcontent.com/publications/features/article_32/ (Note: If you have trouble opening this article from this document, copy the link and paste it into your browser.)

Appendix

Usability problem form

Problem _____

Description:

Violated Heuristic(s)

Which heuristic(s) does this problem violate? Circle the number(s) that apply.

1	Visibility of system status	**6**	Recognition rather than recall
2	Match between system and the real world	**7**	Flexibility and efficiency of use
3	User control and freedom	**8**	Aesthetic and minimalist design
4	Consistency and standards	**9**	Help users recognize, diagnose, and recover from errors
5	Error prevention	**10**	Help and documentation

C.3   Exercise Instructions

An Exercise

Task

- Perform a Heuristic Evaluation on the following Web site:
  http://www.gutenberg.org

Duration

- 40 minutes (you might reserve 5 minutes for preparation)

Documenting usability problems

- Use *usability problem forms* to record problems found. (See the HE materials section.)
- No need to submit these usability problems.

Heuristics

- Heuristics by Jakob Nielsen
- You might want to print them to have them handy.

Getting started

You may try one or more of the following:

- Spend a couple of minutes to get an overall impression of the Web site.
- Concentrate on evaluating a single page or some functionality of the Web site.
- Inspect a particular area of the Web page. For example, evaluate specifically the navigation aids.
- Set up a goal and try to achieve it. For example, your goal could be: looking for the book called "The Little Prince" and see if you find problems along the way.

C.4   Task Instructions

The Task

The purpose of this study is to learn how annotations are used during a Heuristic Evaluation, so please use annotations extensively whenever it is appropriate.

Apply Heuristic Evaluation to find usability problems in the following Web page:

> Home page
>
> of
>
> Transportation Services, Texas A&M University

- You can visit other Web pages in the Web site, but try to focus on the Home page.

- Use the provided *usability problem form* to describe problems that are found. (See HE materials section.)

- Spend 35 minutes in the evaluation.

- Heuristics by Jakob Nielsen. You might want to print them to have them handy.

Output of task

The output of this task is the following files:

1. Annotated Web page(s). How to export annotated Web pages in HEAssistant.

2. Usability problem reports.

Submit files on completion

Once you have completed the evaluation, please submit your files via e-mail to <<Include the e-mail address assigned for the study here.>>.

C.5   Post-Questionnaires

C.5.1   Web Developer Questionnaire

```
            W E B    D E V E L O P E R
                 POST QUESTIONNAIRE
        Supporting Heuristic Evaluation for the Web
-----------------------------------------------------------------------

You were asked to use the HEAssistant tool to annotate Web pages when
performing a Heuristic Evaluation. This questionnaire is to help me
understand your background and experience using the tool.


=================================================================

Background
==========

1. What age group are you in?
Select one.
____  a) Under 20 years
____  b) 20 - 29 years
____  c) 30 - 39 years
____  d) 40 - 49 years
____  e) 50 - 59 years
____  f) Over 59 years

2. How long have you authored Web pages or developed Web applications?
Select one.
____  a) Less than a year
____  b) 1 year
____  c) 2-3 years
____  d) 4-5 years
____  e) 6-8 years
____  f) Over 8 years

3. How often do you author Web pages or develop Web applications?
Select one.
____  a) Less than monthly
____  b) Monthly
____  c) Weekly
____  d) Daily

4. Did you know Heuristic Evaluation before participating in this
study?
Select one.
____  a) Yes
____  b) No
```

4.1. If so, how many times have you use it in projects?
Select one.
___ None
___ 1 - 2 times
___ 3 - 5 times
___ More than 5 times
___ I don't know

==

Using annotations in inspection
==============================

To answer the following questions you may need to check the annotations
added to Web pages in this session.

5. In this session, how were "notes" used when inspecting Web pages?
   Check all that apply.

   [ ] To specify heuristics being violated
   [ ] To add questions
   [ ] To add brief descriptions of usability problems
   [ ] Other way(s): _____
                      _____
                      _____

   [ ] Not Applicable
   [ ] I don't know

6. In this session, how were "push pins" used when inspecting Web
pages?
   Check all that apply.

   [ ] To mark problem interface elements/areas
   [ ] To mark problem interface elements/areas with several problems
        using multiple push pins
   [ ] Other way(s): _____
                      _____
                      _____

   [ ] Not Applicable
   [ ] I don't know

7. In this session, how were "question marks" used when inspecting Web
pages?
   Check all that apply.

   [ ] To mark problem interface elements/areas when drawing questions
        about them, e.g. "What does this mean?"
   [ ] To mark problem interface elements/areas which have something
        confusing about them
   [ ] Other way(s): _____
                      _____
                      _____

   [ ] Not Applicable
   [ ] I don't know

```
Annotating frequency in inspection
==================================


I am interested in learning the frequency with which scenarios (a) and
(b) below occurred for all annotation types:
a) Adding annotations one after the other
b) Adding annotations sporadically

Please answer the following questions using this scale:
     1) Rarely
     2) Occasionally
     3) Frequently


"Notes"
=======

8. If "notes" were used:
How often did you add several "notes" one after the other?
Select one.
____   1) Rarely
____   2) Occasionally
____   3) Frequently
____   Not Applicable
____   I don't know

9. If "notes" were used:
How often did you add "notes" sporadically?
Select one.
____   1) Rarely
____   2) Occasionally
____   3) Frequently
____   Not Applicable
____   I don't know

"Push pins"
===========

10. If "push pins" were used:
How often did you add several "push pins" one after the other?
Select one.
____   1) Rarely
____   2) Occasionally
____   3) Frequently
____   Not Applicable
____   I don't know

11. If "push pins" were used:
How often did you add "push pins" sporadically?
Select one.
____   1) Rarely
____   2) Occasionally
____   3) Frequently
____   Not Applicable
____   I don't know
```

```
"Question marks"
================
```

12. If "question marks" were used:
How often did you add several "question marks" one after the other?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

13. If "question marks" were used:
How often did you add "question marks" sporadically?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

```
Annotations in general
======================
```
Answer based on your OVERALL impression of using annotations.

14. If "annotations" were used:
How often did you add several "annotations" one after the other?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

15. If "annotations" were used:
How often did you add "annotations" sporadically?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

```
=================================================================
```

```
Documenting usability problems
==============================
```

16. Did you use annotations to document usability problems?
Select one.
___ a) Yes
       If so, describe briefly how they were used:
       _____
       _____
       _____
___ b) No
___ I don't know

17. Did you document all the usability problems you found?
Select one.
___ a) Yes
___ b) No
___ I don't know

18. Are usability problems descriptions ready for other people to read?
Select one.
___ a) Yes
___ b) No
___ I don't know

```
=================================================================
```

```
Tool user satisfaction
======================
```

19. Would you recommend the tool to somebody else?
Select one.
___ a) Possibly yes
___ b) Possibly no
___ I don't know

20. Would you use the tool again for a Heuristic Evaluation project?
Select one.
___ a) Possibly yes
___ b) Possibly no
___ I don't know

21. Which features were you expecting to see and were not available?
_____
_____
_____

```
=================================================================
```

```
Comments?
=========
```
_____
_____

C.5.2   Software Engineer Questionnaire

```
          S O F T W A R E   E N G I N E E R
               POST QUESTIONNAIRE
         Supporting Heuristic Evaluation for the Web
----------------------------------------------------------------------
```

You were asked to use the HEAssistant tool to annotate Web pages when
performing a Heuristic Evaluation. This questionnaire is to help me
understand your background and experience using the tool.
===

Background
==========

1. What age group are you in?
Select one.
___ a) Under 20 years
___ b) 20 - 29 years
___ c) 30 - 39 years
___ d) 40 - 49 years
___ e) 50 - 59 years
___ f) Over 59 years

2. How long have you developed software?
Select one.
___ a) Less than a year
___ b) 1 - 5 years
___ c) 6 - 10 years
___ d) 11 - 15 years
___ e) Over 15 years

3. How often do you develop software?
Select one.
___ a) Less than monthly
___ b) Monthly
___ c) Weekly
___ d) Daily

4. Did you know Heuristic Evaluation before participating in this
study?
Select one.
___ a) Yes
___ b) No

     4.1. If so, how many times have you use it in projects?
     Select one.
     ___   None
     ___   1 - 2 times
     ___   3 - 5 times
     ___   More than 5 times
     ___   I don't know
===

```
Using annotations in inspection
===============================


To answer the following questions you may need to check the annotations
added to Web pages in this session.

5. In this session, how were "notes" used when inspecting Web pages?
   Check all that apply.

   [ ]  To specify heuristics being violated
   [ ]  To add questions
   [ ]  To add brief descriptions of usability problems
   [ ]  Other way(s): _____
                      _____
                      _____

   [ ]  Not Applicable
   [ ]  I don't know

6. In this session, how were "push pins" used when inspecting Web
pages?
   Check all that apply.

   [ ]  To mark problem interface elements/areas
   [ ]  To mark problem interface elements/areas with several problems
        using multiple push pins
   [ ]  Other way(s): _____
                      _____
                      _____

   [ ]  Not Applicable
   [ ]  I don't know

7. In this session, how were "question marks" used when inspecting Web
pages?
   Check all that apply.

   [ ]  To mark problem interface elements/areas when drawing questions
        about them, e.g. "What does this mean?"
   [ ]  To mark problem interface elements/areas which have something
        confusing about them
   [ ]  Other way(s): _____
                      _____
                      _____

   [ ]  Not Applicable
   [ ]  I don't know

====================================================================
```

```
Annotating frequency in inspection
==================================


I am interested in learning the frequency with which scenarios (a) and
(b) below occurred for all annotation types:
a) Adding annotations one after the other
b) Adding annotations sporadically

Please answer the following questions using this scale:
     1) Rarely
     2) Occasionally
     3) Frequently


"Notes"
=======

8. If "notes" were used:
How often did you add several "notes" one after the other?
Select one.
____  1) Rarely
____  2) Occasionally
____  3) Frequently
____  Not Applicable
____  I don't know

9. If "notes" were used:
How often did you add "notes" sporadically?
Select one.
____  1) Rarely
____  2) Occasionally
____  3) Frequently
____  Not Applicable
____  I don't know

"Push pins"
===========

10. If "push pins" were used:
How often did you add several "push pins" one after the other?
Select one.
____  1) Rarely
____  2) Occasionally
____  3) Frequently
____  Not Applicable
____  I don't know

11. If "push pins" were used:
How often did you add "push pins" sporadically?
Select one.
____  1) Rarely
____  2) Occasionally
____  3) Frequently
____  Not Applicable
____  I don't know
```

```
"Question marks"
================
```

12. If "question marks" were used:
How often did you add several "question marks" one after the other?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

13. If "question marks" were used:
How often did you add "question marks" sporadically?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

```
Annotations in general
======================
```
Answer based on your OVERALL impression of using annotations.

14. If "annotations" were used:
How often did you add several "annotations" one after the other?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

15. If "annotations" were used:
How often did you add "annotations" sporadically?
Select one.
___ 1) Rarely
___ 2) Occasionally
___ 3) Frequently
___ Not Applicable
___ I don't know

```
==================================================================
```

```
Documenting usability problems
==============================

16. Did you use annotations to document usability problems?
Select one.
___  a) Yes
        If so, describe briefly how they were used:
        _____
        _____
        _____
___  b) No
___  I don't know

17. Did you document all the usability problems you found?
Select one.
___  a) Yes
___  b) No
___  I don't know

18. Are usability problems descriptions ready for other people to read?
Select one.
___  a) Yes
___  b) No
___  I don't know

=================================================================

Tool user satisfaction
======================

19. Would you recommend the tool to somebody else?
Select one.
___  a) Possibly yes
___  b) Possibly no
___  I don't know

20. Would you use the tool again for a Heuristic Evaluation project?
Select one.
___  a) Possibly yes
___  b) Possibly no
___  I don't know

21. Which features were you expecting to see and were not available?
_____
_____
_____

=================================================================


Comments?
=========
_____
_____
```

VITA

Ana Erendira Flores Mendoza

Address

Department of Computer Science and Engineering

Texas A&M University

TAMU 3112

College Station, Texas 77843-3112

Education

Ph. D. in Computer Science, Texas A&M University, College Station, Texas, August 2009.

M.S. in Computer Science, New Mexico State University, Las Cruces, New Mexico, July 1997.

B.S. in Electronics Engineering, Control and Instrumentation minor, Universidad Autonoma Metropolitana, Mexico City, Mexico, July 1994.