Towards Dissecting the Heuristic Evaluation Process

Ana Erendira Flores-Mendoza, William Lively

Department of Computer Science
Texas A&M University
TAMU 3112
College Station, TX 77843-3112 USA
{aflores,lively}@cs.tamu.edu

Abstract. This paper discusses preliminary findings of an on-going, in-depth observational study of Heuristic Evaluation applied to the Web. The goal of this study is to understand Heuristic Evaluation in order to propose better tools to support it. In this part of the study we focus on characterizing the evaluation process when applying Heuristic Evaluation on a paper-based, non-interactive Web interface. A group of evaluators were observed on video evaluating an E-commerce Web site. Several activities were found: 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. The significance of these results for tools supporting Heuristic Evaluation is then explored.

Keywords: Heuristic Evaluation, evaluation process analysis, paper interface evaluation, video analysis, Web usability evaluation tools

1. Introduction

As part of the Web development process, Web developers are confronted with evaluating the usability of Web sites. Web developers need effective and cheap approaches for Web usability evaluation.

Heuristic Evaluation is an inspection method proposed by Nielsen and Molich [11]. 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 inspect a user interface in search for usability problems. Evaluators then collaborate to aggregate the usability problems. As a guide, evaluators use a set of usability principles (the "heuristics" [7]) to focus on common problem areas in user interfaces. Its effectiveness depends on different dimensions such as the number of evaluators [10] and their expertise, the checklist of heuristics that is used [3], and the format used to record usability problems [1].

The goal of this research is to support Web developers evaluate Web interfaces using Heuristic Evaluation. Finding ways to facilitate evaluation is of interest. The first step towards this goal is to understand the process deeper to propose better tools to support it.

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 34, 2008, pp. 277-288

Received 24/03/08
Accepted 26/04/08
Final version 05/05/08

This paper discusses preliminary findings of this on-going research project. Specifically, we will describe 1) the Heuristic Evaluation process as it can be drawn from the literature, 2) detailed description of the study, 3) results and characterization of the analyzed process, and 4) implications of results for tools supporting Heuristic Evaluation.

2. Heuristic Evaluation Process Overview

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. Fig. 1 shows an overview the Heuristic Evaluation process.

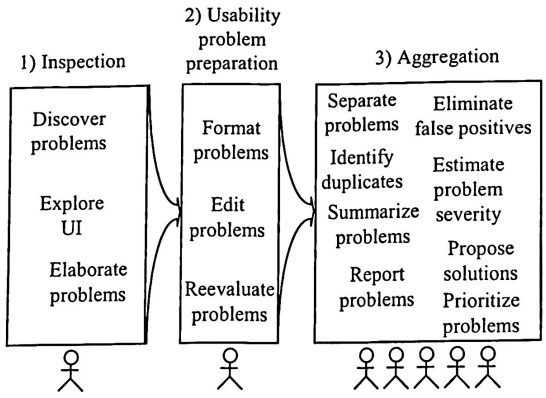

Fig. 1. Heuristic Evaluation Process Overview: 1) user interface inspection, 2) usability problem preparation, 3) usability problem lists aggregation

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 [8] 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 [12]) that allow particular ways of exploration. For example, several paper screenshots can be compared at once by positioning them side by side. Computer mockups [6], 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" [6]).

Problem search influence how interfaces are explored. Cockton et al. [1] 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.

Identifying Usability Problems. There are other factors than interface format and search strategies that may induce evaluators to notice potential problems. Inspection guidelines [5] and [16], for example, are intended to "stimulate inspectors to notice things about the software that might lead, on further reflection, to identifying a potential problem [5]." Inspection guidelines give details of how to proceed, what to focus on, if post-meetings are needed, among others [16]. 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. [5]". Evaluators can draw on different sources to elaborate problems [5]: 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. Usability Problem Preparation Phase

Evaluators may need to format, edit, and reevaluate usability problems before aggregation. Cox [2] talks about formatting problems to facilitate aggregation of problem lists.

2.3. Aggregation Phase

Cox elaborated more on what problem aggregation is. It involves not only arranging, selecting, or categorizing identified raw problems [2] (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." [2] (p. 139)

3. Study Description

This is the first part of an on-going, in-depth observational study of Heuristic Evaluation applied to the Web. We look into the inspection phase of Heuristic Evaluation when evaluating a paper-based, non-interactive Web interface. We are interested on the effects interface format has on inspection. In dissecting the process further we identify main activities evaluators exhibit in action. Tool features can then be proposed to support those activities.

3.1. 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.

3.2. 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 [13] 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[1] 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 [14] 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).

- WP1. Home page
- WP2. "DVD Movies" category page

[1] The Web site of the Zen Cart project can be found at: http://www.zencart.com/.

- 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.

3.3. Procedures

The study consisted of two phases. Phase 1 was a training phase. Participants would get 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. Participants would come to the laboratory and asked to perform a Heuristic Evaluation on the static Web interface. After the evaluation session they were interviewed about their approaches to finding problems.

A Web site was developed to guide participants through Phase 1. Participants were asked to complete a background questionnaire, download training materials, and practice applying the method. They practiced a Heuristic Evaluation on the Gutenberg[2] Web site 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 [9] ten usability heuristics and Wood's [15] explanation of the heuristics for Web interfaces were provided.

Phase 2 took about an hour. Evaluators spent 20 minutes in applying Heuristic Evaluation on the given Web interface. 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. Fig. 2 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 Fig. 3). They were asked to keep the notebook within the marked area to capture writing events.

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 [1] format. The problem attributes include a brief problem description, discovery method (i.e. System Scanning, System Searching, Goal Playing, and Method Following [1]) specification and description, specific steps taken in finding the problem, and reference to violated heuristic(s).

[2] The Gutenberg Web site can be found at: http://www.gutenberg.org/.

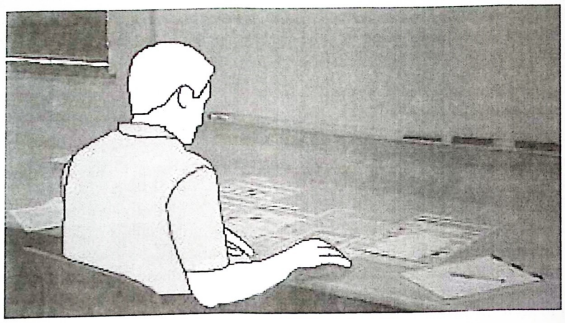

Fig. 2. Study setting

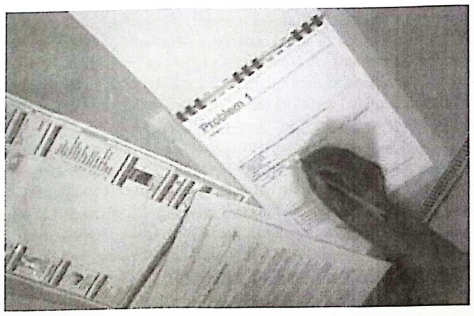

Fig. 3. Writing setting

Fig. 4 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 [9] 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.

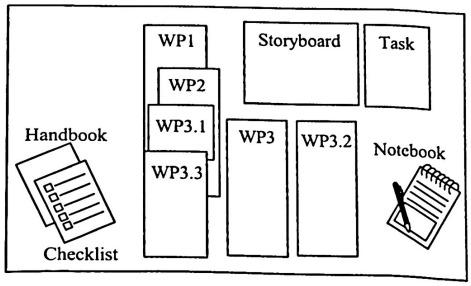

Fig. 4. Initial material arrangement. *Supplementary materials*: Heuristic Evaluation booklet, Nielsen's [9] heuristic checklist; *Interface materials*: Home page (WP1), Category page (WP2), Product page (WP3), "Larger image" page (WP3.1), "Shopping Cart" (WP3.2), "Sign-in" page (WP3.3), Storyboard; Notebook; Task description

3.4. 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 [4] 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 through out 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.

4. Characterized (Static) Heuristic Evaluation Process

After analyzing all videos, a rough characterization of the Heuristic Evaluation process was obtained. Fig. 5 shows the data aggregated from all 7 participants. 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.

4.1. Observing and Quickly Visiting the Interface

An "observe" event is defined as the time spent in carefully examining a screenshot before starting to write. We 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. We argue 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.

Our 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 spent in this activity, which is the second highest time observed. This is an indication of the importance observation has in Static evaluation.

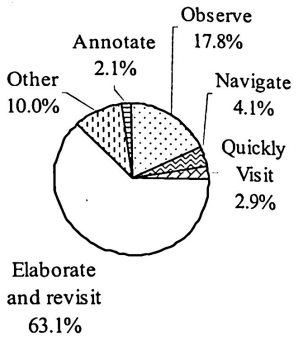

Fig. 5. A Rough Characterization of the Static Heuristic Evaluation Process

4.2. 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.

4.3. 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 our 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, that evaluators were instructed to focus on evaluating the product page. Therefore, a different time distribution could be obtained in a more free-form inspection.

4.4. Annotating the Interface

Two evaluators annotated the product page. They 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"). Note that not all evaluators were instructed about the possibility of writing on Web pages. This was a facilitator's slip. Therefore, more annotation instances could have occurred.

Some open issues were left unanswered that are worth investigating to support annotation for inspection. For instance, researching the effects of annotation in problem discovery is worth pursuing. Do evaluators tend to report more usability problems when annotation is used? What kinds of problems are discovered? Does annotation facilitate problem discovery?

5. 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 [2], and can be adopted and enhanced. Tool features for problem preparation can be proposed based on [2] and results from this research.

In this paper we describe inspection in detail when applying Heuristic Evaluation to static Web interfaces. Evaluators observe, quickly visit, navigate, and annotate the interface, as well as elaborate usability problems.

The next step in this research is to propose specialized tools to aid evaluators perform these activities. For instance, we observed that evaluators that made use of annotation in inspection used different kinds of elements to mark and put notes on the Web page in question. Tools that allow adding these elements, retrieving and analyzing them are candidates to support annotation in inspection.

6. Conclusions

The Heuristic Evaluation process has been dissected at a certain degree. We analyzed and characterized the inspection phase of Heuristic Evaluation through an in-depth observational study. Evaluators were observed on video performing a Heuristic Evaluation on a paper-based, non interactive Web interface. Evaluators were found observing, quickly visiting, navigating, and annotating the interface, and elaborating usability problems. These activities help define the types of features suitable to support inspection.

We discuss implications for tool development based on related work and Heuristic Evaluations. Future work involves specifying tool requirements, implement those requirements, and test the tools for user satisfaction.

References

1. Cockton, G., Woolrych, A., Hall, L., Hindmarch, M.: Changing Analysts' Tunes: The Surprising Impact of a New Instrument for Usability Inspection Method Assessment. In: O'Neill, E., Palanque, P., Johnson, P. (eds.) People and Computers XVII, pp.145–161. Springer-Verlag, Berlin Heidelberg New York (2003)
2. Cox, D.A.: Supporting Results Synthesis in Heuristic Evaluation. Master's thesis, Department of Computer Science, University of Calgary, Canada (1998)
3. Dykstra, D.J.: 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 (1993)
4. Kuniavsky, M.: Observing the User Experience: a Practitioner's Guide to User Research, Morgan Kaufmann, San Francisco (2003)
5. Mack, R.L., Montaniz, F.: Observing, Predicting, and Analyzing Usability Problems. In: Nielsen, J., Mack, R.L. (eds.) Usability Inspection Methods, pp.295–339. Wiley, New York (1994)
6. Nielsen, J.: Paper versus Computer Implementations as Mockup Scenarios for Heuristic Evaluation. In: Diaper, D., Gilmore, D., Cockton, G., Shackel, B. (eds.) 3rd IFIP International Conference on Human-Computer Interaction, INTERACT 90, pp.315–320. North-Holland (1990)
7. Nielsen, J.: Enhancing the Explanatory Power of Usability Heuristics. In: Adelson, B., Dumais, S., Olson, J.S. (eds.) Conference on Human Factors in Computer Systems, CHI 94, pp.152–158. ACM (1994)

8. Nielsen, J.: How to Conduct a Heuristic Evaluation,
9. http://www.useit.com/papers/heuristic/heuristic_evaluation.html
10. Nielsen, J.: Ten Usability Heuristics,
11. http://www.useit.com/papers/heuristic/heuristic_evaluation.html
12. Nielsen, J., Landauer, T.K.: A Mathematical Model of the Finding of Usability Problems. In: Ashlund, S., Mullet, K., Henderson, A., Hollnagel, E., White, T. (eds.) Conference on Human Factors in Computer Systems, CHI 93, pp.206–213. ACM (1993)
13. Nielsen, J., Molich, R.: Heuristic Evaluation of User Interfaces. In: Carrasco Chew, J., Whiteside, J. (eds.) Conference on Human Factors in Computer Systems, CHI 90, pp.249–256. ACM (1990)
14. Norman, D.A.: The Design of Everyday Things. Basic Books (2002)
15. Snyder, C.: Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces. Morgan Kaufmann, San Francisco (2003)
16. Virzi, R.A.: What Can You Learn from a Low-Fidelity Prototype? In: Human Factors Society 33rd Annual Meeting, pp.224–228. Human Factors Society (1989)
17. Wood, J.: Usability Heuristics Explained,
18. http:// iqcontent.com/publications/features/article_32/ (2004)
19. Zhang, Z., Basili, V., Shneiderman, B.: Perspective-based Usability Inspection: An Empirical Validation of Efficacy. Empirical Software Engineering 4(1), 43–69 (1999)