HUMAN TECHNOLOGY

An Interdisciplinary Journal on Humans in ICT Environments ISSN: 1795-6889

www.humantechnology.jyu.fi Volume 5(2), November 2009, 121–145

A QUANTITATIVE METHOD FOR LOCALIZING USER INTERFACE PROBLEMS: THE D-TEO METHOD

Juha Lamminen
Agora Center
University of Jyväskylä
Finland

Mauri Leppänen
Department of Computer Science and Information Systems
University of Jyväskylä, Finland

Risto Heikkinen
Department of Mathematics and Statistics
University of Jyväskylä
Finland

Anna Kämäräinen
Agora Center
University of Jyväskylä
Finland

Elina Jokisuu
Agora Center
University of Jyväskylä
Finland

Abstract: *A large array of evaluation methods have been proposed to identify Website usability problems. In log-based evaluation, information about the performance of users is collected and stored into log files, and used to find problems and deficiencies in Web page designs. Most methods require the programming and modeling of large task models, which are cumbersome processes for evaluators. Also, because much statistical data is collected onto log files, recognizing which Web pages require deeper usability analysis is difficult. This paper suggests a novel quantitative method, called the D-TEO, for locating problematic Web pages. This semiautomated method explores the decomposition of interaction tasks of directed information search into elementary operations, deploying two quantitative usability criteria, search success and search time, to reveal how a user navigates within a web of hypertext.*

Keywords: *D-TEO method, usability, quantitative method, usability testing, log-based evaluation.*

INTRODUCTION

In the last two decades, the World Wide Web (Web) has become one of the most important means of disseminating and searching for information. Companies, government agencies,

DOI: http://dx.doi.org/10.17011/ht/urn.200911234467

municipalities, communities, and individual persons maintain a plethora of Web sites on the Internet, Intranets and Extranets, and the number of sites is increasing explosively (see Netcraft, 2009). Examples of drivers fueling this progress are eGovernment initiatives and programs that foster more efficient and effective provision of government services through the Internet (Cordella, 2007; Wolf & Krcmar, 2008). Web sites are often so large and lacking integration that finding a desired piece of information appears to be quite difficult and time consuming. It is not atypical that users become disoriented and "lost" in this hypertext world (Dillon, McNight, & Richardson 1990). The primary reason for these kinds of problems stem from poor design of Web sites (Nielsen, 1993).

Various principles (e.g., Nielsen, 1993; Schneiderman, 1998; Tidwell, 2005), techniques (e.g., Goldberg, Stimson, Lewenstein, Scott, & Wichansky, 2002), and methods (e.g., Beyer & Holtzblatt, 1998; Mayhew, 1999) have been developed for designing Web sites to satisfy usability criteria. *Usability* means "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (International Organization of Standards [ISO], 1998). In our case, the *product* is a Web site composed of Web pages. Beyond the criteria indicated in the definition by ISO 9241-11 (1998), usability is seen to embrace other criteria, such as ease of learning, error rates, memorability, reliability in use, retention over time, throughput, and so on (cf. Constantine & Lockwood 1999; Nielsen, 1993; Preece et al., 1994; Schneiderman, 1992; Seffah, Donyaee, Kline, & Padda, 2006; Shackel, 1991; Wixon & Wilson 1997).

There is also a wide array of techniques (e.g., Chi et al., 2003; Paganelli & Paternò, 2002) and methods (e.g., Blackmon, Polson, Kitajima, & Lewis, 2002; Card et al., 2001) for evaluating and testing Web sites. The objective of carrying out an evaluation can be to test whether a design is appropriate, to compare alternative designs, or to check conformance to a standard (Lecerof & Paternò, 1998). Commonly applied methods are heuristic evaluation, usability testing, and log-based evaluation (cf. Matera, Rizzo, & Carughi, 2006). In a *heuristic evaluation* (Nielsen & Mack, 1994; Nielsen & Molich, 1990), the usability problems are identified in a heuristic fashion by a usability expert. The main concerns about a heuristic evaluation are that it does not include the actual end users (Nielsen & Mack, 1994) and the number of expert evaluators is often too low (Cockton & Woolrych, 2002). In *usability testing* (Dumas & Redish, 1993), the participants represent real end users and everything that they do and say during the usability test is observed and recorded. After the usability test itself, the data are analyzed and suggestions to eliminate the problems are proposed. The concerns regarding usability testing are that this process is based only on observational data and that user interface experience is needed to be able to interpret the data (Lecerof & Paternò, 1998). There is also the problem of cost and the time of the users and the observers (Lecerof & Paternò, 1998). In *log-based evaluation*, information about the performance of the users is collected automatically and stored in log files (e.g., Lecerof & Paternò, 1998). A benefit of this method is that large amounts of data can be collected in an exact form and with reduced work and cost (for more benefits, see Ivory & Hearst, 2001). The weak points of the method are that some handwork (e.g., adding code to the target system) is needed and the use environment is typically restricted to certain applications (Scholtz & Laskowski, 1998).

Usability evaluation and testing apply both qualitative (e.g., user satisfaction, easy to use) and quantitative measures (Mayhew, 1999; Stone, Jarret, Woodroffe & Minocha, 2005; Wixon & Wilson, 1997). The most common quantitative measures are task completion time, the

number of errors, and the success or failure in executing the tasks (e.g., Martin & Weiss, 2006; Masemola & De Villiers, 2006; Nielsen, Overgaard, Pedersen, Stage, & Stenild, 2006). Typically, values derived from the evaluations are compared to the predefined target values. The number of failed and successful attempts and the total number of attempts in each task are used to find out how difficult the task is. Masemola and De Villiers (2006) also use log files to record the number of mouse clicks. Others have combined quantitative measurements with qualitative evaluation to identify usability problems and evaluate the number and severity of the problems (e.g., De Angeli, Sutcliffe, & Hartmann, 2006; Duh, Tan, & Chen, 2006; White, Wright, & Chawner, 2006). Freeman, Norris, and Hyland (2006) have evaluated the navigation processes with the aim of getting a more accurate picture of a product's usability, particularly its efficiency (see more about evaluation methods in Ivory & Hearst, 2001).

Making a careful and in-depth usability evaluation of a large Web site requires significant time and resources (Dumas & Redish, 1993; Mayhew, 1999; Nielsen, 1993). Unfortunately, these often are not available in most situations. Therefore, there should be some means to first distinguish those parts of a Web site that seem to be more problematic, so that scant resources can be applied directly to a deeper evaluation of these areas only.

We propose a novel usability testing method, called D-TEO (Decomposition of Tasks into Elementary Operations), that aims to locate usability problems in the information search process in Web sites. The basic idea in D-TEO is to decompose a user task into elementary operations and define, for each task, an optimal navigation path composed of operations. In order to satisfy usability requirements, the structure and contents of a Web site should guide the users to find the optimal paths and to follow them efficiently. D-TEO helps identify Web pages that cause problems for the users and, based on this information, usability designers can focus their attention on these pages specifically.

This paper is organized as follows: In the next section, we define basic concepts related to Web sites, user tasks, information search, and search metrics. In the following section, we describe the proposed method. Later, we provide an example of the method in use, and then offer a short comparative review of related works. The final section presents a summary and conclusions.

BASIC CONCEPTS

Web Sites and Web Pages

A *Web site* is a collection of Web pages that is hosted on one or more Web servers. A *Web page* is a hypertext document, typically written in HTML or XHTML format. *Hypertext* involves data that are stored in a network of nodes connected by links. The interconnecting nodes form an interdependent web of information that is nonlinear. The nonlinearity enables great flexibility in the selection of information, but at the same time increases risks of disorientation.

There are two primary hypertext topologies (Batra, Bishu, & Donohue, 1993; Bernard, 2002). In the *strict hierarchical structure*, nodes are grouped in a hierarchical arrangement, allowing movement either up or down, but only one level at a time. In the *network topology* it is possible, in the most extreme case, to move through so-called referential hyperlinks from each node to every other node. Between these two types of topologies, there are mixed hierarchies, which allow limited movements from nodes to some other nodes at different levels within the structure.

A Web page consists of *user interface components*, such as titles, text boxes, data fields, tables, check boxes, radio and control buttons, menus, text links and image links, icons, forms, frames, and scroll bars. A user is allowed to make selections through menus or buttons, thus triggering the transmission of requests to the Web server to return the desired information in a new Web page. In the traditional Web application, communication between a client and a Web server is asynchronous, and the whole Web page is returned. In rich Internet applications, communication is synchronous and only part of the Web page can be substituted by a new one (Paulson, 2005; Preciado, Lanaje, Sanchez, & Comai, 2005).

User Actions, Tasks, and Operations

A user deploys an application as an instrument in order to improve his/her abilities to carry out some action (Saariluoma, Parkkola, Honkaranta, Leppänen, & Lamminen, 2009). *Actions* are composed of four kinds of tasks (Lecerof & Paternò, 1998). A *user task* is an action that is exclusively performed by a user, that is to say, without any interaction with the application. An *application task* is completely executed by the application. An *interaction task* is performed by the user interacting with the application. An *abstract task* requires complex actions whose performance allocation has not yet been decided. From the perspective of this paper, we are interested in interaction tasks. They can be further divided into categories, depending on the types of tasks the application makes: information search, on one hand, and information insert, update, and delete, on the other hand. Here, we only consider information search.

The tasks can be at different abstraction levels, ranging from high-level tasks to very low-level tasks. An execution of a task necessitates that all of its subtasks are carried out in a predefined manner. Decomposing a task into subtasks establishes a hierarchical tree in which subtasks on the lowest level are called *operations*; these elementary tasks focus on a single user interface component (e.g., the OK button). The execution of an operation triggers the transmission of a request to the Web server to search for the desired information and return it in a new Web page. An operation can also return a previous page (i.e., back-page button).

Information Search

Web sites show up as webs of hypertext that contain information of interest to the user. Information can be searched for in two ways (Bernard, 2002). The first type is a *directed search*, also called explicit search (Norman & Chin, 1988). The purpose of this type of search is to acquire specific information about a target item (e.g., find the title of the 1953 film that starred Audrey Hepburn and Gregory Peck). The second type is an *exploratory search* that involves the broader goals of finding and integrating information from various nodes in a web of hypertext. This is also called browsing (Canter, Rivers, & Storrs, 1985). A user explores the hypertext by continually refining his/her search until the information goal is satisfied. An example of this kind of search is "Compare the movies *Independence Day* and *Sleepers* by using the information the MovieGuide can give you" (Bernard, 2002). An exploratory search takes more time and causes disorientation more often, partly because it poses more cognitive burden (Kim & Hirtle, 1995; Norman & Chin, 1988; Smith, 1996). We focus on directed searches in this study.

A page containing the target item is called a *terminal node.* Information search proceeds from an entry node to the terminal node through hyperlinks. The shortest route to the specific

terminal node that satisfies a search task is called an *optimal path* (Bernard, 2002; Gwizdka & Spence, 2007; Norman & Chin, 1988). The length of the path depends on how many nodes (Web pages) have to be visited during the search. In a Web site following a mixed or network topology, there may be several optimal paths for one information search.

Metrics

How effectively and efficiently a desired piece of information can be found is influenced by several factors, including size of the web of hypertext, the breadth and depth of hypertext topology and its compliance with the users' mental models, the visualization of Web pages, the understandability of terms used in Web pages, and so on. Generally speaking, the effectiveness and efficiency of the search task depend upon the usability of a Web site. The literature presents a large variety of definitions and taxonomies for usability (e.g., Constantine & Lockwood, 1999; ISO, 1998; Nielsen, 1993; Preece et al., 1994; Schneiderman, 1992; Shackel, 1991). It goes beyond the scope of this paper to discuss them in more detail (see the analysis by Seffah et al., 2006). Therefore, we quote ISO 9241-11 (1998), which distinguishes between three main usability attributes: effectiveness, efficiency, and satisfaction. We are particularly interested in search efficiency.

Search efficiency is commonly measured in terms of search time, navigation accuracy, lack of disorientation, and success in finding the desired page. For instance, Bernard (2002) defined timed accuracy as the number of times a user fails to find the correct terminal node. Search efficiency is measured by examining the number of deviations from the optimal path and by the number of total back-page presses used in reaching the targeted node. Search time means the time taken to correctly complete the given task.

Our metrics of search efficiency is composed of two criteria: search success and search time. *Success* measures the extent to which a user follows the optimal path when he/she is carrying out the task. Deviations from the optimal path, or back and forth movement in the path (e.g., through the back-page button), decreases the measure of success. Usage of the back-page button suggests uncertainty in the navigation paths taken (cf. Normal & Chin, 1988). *Search time* represents the time that it takes the user to complete the task from start to finish. Since a task is decomposed into elementary operations, it is also possible to measure the time required to carry out an operation, that is, how long from the end of one operation to the end of the next operation. For each of these two criteria, a set of measures were defined and used.

METHOD

This section describes the proposed method for locating problems in Web pages. We first describe the objectives and application domain of the method. Then, we detail the steps of the method.

Application Domain

D-TEO is a usability testing method for revealing problems in user navigation in Web sites. The Web sites can be either in the prototype phase or in production. The basic idea underlying the method is to examine how closely a user follows the optimal paths and how fast he/she performs the given interaction tasks. Deviations from the optimal path and/or

delays in executions indicate problems that should be examined more closely with some other usability evaluation methods (e.g., heuristic methods).

The D-TEO can be integrated into a Web application development method or a hypermedia development method. The literature provides a large variety of these kinds of methods (e.g., OOHDM, Rossi & Schwabe, 2006; RMM, Isakowitz, Stohr, & Balasubramanian, 1995; IDM, Lee, Lee, & Yoo, 1998; W2000, Baresi, Garcotto, & Paolini, 2001; UWE, Hennicker & Koch, 2001; WHDM, Lee & Suh, 2001). For example, within the WHDM method, the D-TEO technique can be deployed to test prototypes produced by the design activities of navigation design and interface design.

Steps

The D-TEO method is composed of four steps: (a) define the goals of the test and the user profiles, (b) devise the test tasks and identify the optimal paths, (c) organize the test and collect data, and (d) analyze the data and make the conclusions (Figure1). In the following subsections, the steps are described in more detail.

Define the Goals of the Test and the User Profiles

The use of the method starts with defining the goals for the test at hand. The goal statements should describe which subset of the Web pages should be tested, which pages in this subset are particularly important, and how the results from the test are to be utilized. The goal setting is affected by whether the Web site is in the prototype phase or already in use, and the reasons triggering the test. In order to define the user profiles, it is important to identify the audience at whom the Web site is targeted. Conducting a survey or interviews among the current and potential users helps to define the typical characteristics (user profiles) of the primary user categories in terms of their skills, motivations, experience, and so on (e.g., Mayhew, 1999).

Decisions regarding which user groups, and to what extent, are included in the test are based on the goals of the test and the resources available.

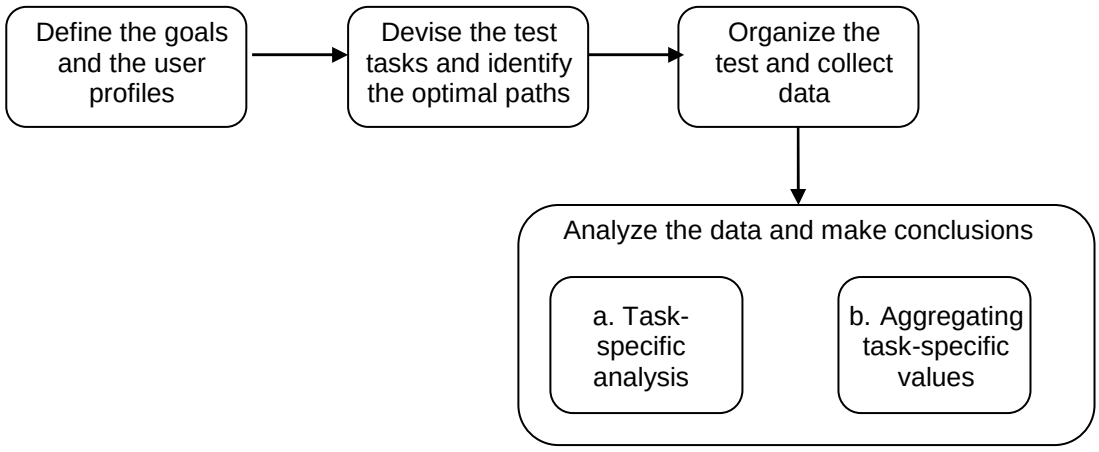

Figure 1. The steps of the D-TEO method.

Devise the Test Tasks and Identify the Optimal Paths

A *test task* is a typical interaction process carried out by a person representing an appropriate user profile. In order to devise a set of relevant test tasks, the overall structure of the Web site has to be outlined and typical interaction tasks should be recognized through a task analysis. If there is a site map describing the Web site, it can be used to ascertain that the test tasks cover a sufficient number of Web pages. Whether the coverage is sufficient or not is determined based on the goals of the test. As an example, let us assume that one of the test tasks is as follows: "There is one ringette team in the Jyväskylä region. What is the name of this team?"

After specifying the test tasks, they are decomposed into operations. As defined above, an operation is an elementary task that focuses on a single user interface component. To establish decomposition hierarchies of test tasks requires that those responsible for testing have good knowledge about the topological structure of the Web site and details of page visualization. For each test task, it is determined which Web pages should be visited and what operations should be performed, in order to reach the terminal page containing the desired piece of information. The shortest path from the entry page to the terminal page is an optimal path. Because the method is intended for testing directed searches, typically only one, or just a few, optimal paths exists for each test task. If there are several paths with the same number of operations, these paths are analyzed as equals.

As an example of the optimal path, let us consider the test task introduced above. For purposes of analysis, each Web page involved by the test tasks is coded with a number reflecting its position in the hierarchical structure of the Web site. By doing so, we have found the optimal path for this test task is as follows:

$$0 \rightarrow 4 \rightarrow 4.11 \rightarrow 4.11.4 \rightarrow 4.11.4.3.$$

In this coding, 0 means the Entry page, 4 refers to the page Services, 4.11 represents the page Sport (under the Services page), 4.11.4 means the Sports Clubs page (under the Sport page), and finally 4.11.4.3 refers to the page containing the information about the ringette team. The optimal path can be described as an ordered set of numbered Web pages visited. It also shows the operations a user must follow in order to complete the test task. The marking $p_i \rightarrow p_j$ denotes the operation by which a user navigates from one Web page (p_i) to another (p_j).

Organize the Test and Collect Data

The test participants are selected to meet the stated goals of the test and the defined user profiles. The number of participants can vary, depending on the goals of the test. Nielsen and Molich (1990) state that 50% of the most important usability problems can be identified with three users. Other authors claim that five users facilitate the discovery of 90% of the usability problems (e.g., Virzi, 1992).

The test tasks are given to the participants on a sheet of paper. No discussion between a participant and the test facilitator is needed during the test. The test equipment should record, with time stamps, all the actions the participant makes and all the Web pages he/she visits. In addition, the test facilitator can make notes on the behavior of a participant, which can be used later in the analysis of time stamped data. This is, however, optional.

To illustrate the data that is collected, let us continue with our previous example. The optimal path was defined as: 0 → 4 → 4.11 → 4.11.4 → 4.11.4.3. Table 1 depicts the unprocessed data collected about operations by three test participants (P1-P3).

We can see in Table 1 that participant P1 followed the optimal path and successfully completed the test task. Participant P3, on the other hand, carried out the operation 0 → 4 successfully but failed to execute the operation 4 → 4.11 (see 00 in Table 1). He/she also failed to complete the operation 4.11 → 4.11.4 and interrupted at 10:23.

Analyze the Data and Draw Conclusions

The collected data is processed and analyzed in two phases. First, the measure values for single tasks and single participants are derived and analyzed. After that, the operation-specific values are aggregated to concern all the tasks and participants. Based on these analyses, conclusions are drawn.

a. Task-Specific Values

The D-TEO method deploys metrics derived from two evaluation criteria, success and search time (see above). The metrics for task-specific analysis comprise two measures, success value and duration time. These are elaborated here.

For each task, and for each participant, the next questions are considered:

- *How successfully did the participant navigate from one Web page to another along the optimal path?*
  This is measured by *Success Values* (SV) that are derived by the following rule: If an operation in the optimal path was carried out in the first attempt, then SV = 1 for that operation; by the second attempt, SV = 0.5; by the third attempt, SV = 0.33, and so on. If the participant deviated from the optimal path, without returning to it, the

Table 1. Example of Unprocessed Data.

P1		P2		P3	
ID	**T**	**ID**	**T**	**ID**	**T**
0	10:00	0	14:29	0	9:41
4	10:03	4	14:33	4	9:47
4.11	10:10	4.11	15:04	00	9:55
4.11.4	10:15	00	15:16	00	10:04
4.11.4.3	11:27	4.11	15:20	000	10:23
		4.11.4	15:34		
		4.11.4.3	15:55		

Note: ID means the numeric identifier of the Web page. T stands for the clock time (in minutes and seconds) when a participant arrived at a certain Web page. We use the symbol 00 to refer to a Web page that is not on the optimal path. The symbol 000 means that the participant has interrupted the execution of the task.

operation is coded with the number 0. In the event the participant did not find the Web page that is a part of the optimal path, the operation is coded by NA.

- *What was the duration of each operation?*
  It is important to study the time that the participant spent on each Web page (i.e., performing each operation). *Duration* (D) is the difference between time the participant arrived at the page and the time when he/she left the page (by executing the operation). If the participant realized that he/she made a mistake (i.e., deviated from the optimal path) and returned to the previous page, the duration time is the sum of the duration times he/she spent on the page in each visit.

To continue with the example data, consider Table 2. It contains the durations (D) and success values (SV) for the operations of the task by three participants P1, P2, and P3. The duration values (in seconds) have been derived from the clock times in Table 1. The success values have been calculated based on the aforementioned rules. We can see in Table 2 that the participant P2 spent a relatively long time (31 seconds) in performing the operation 4 → 4.11, although he/she finally completed the task. P2 also had problems with the operation 4.11 → 4.11.4 because he/she could not find the page 4.11.4 until the second attempt (SV = 0.5). The participant P3 managed to carry out only the first operation of the task.

b. Aggregating Task-Specific Values

Here, we consider the two evaluation criteria, success and duration, through the following aggregated measures:

- *Average Success Value* (ASV) for an operation. This is obtained by calculating the average success value for the operation in the task across all the participants. The smaller the ASV, the more probable it is that the concerned Web page contains problems.
- *Average Duration* (AD) for an operation. This is derived by calculating the average duration for the operation in the task across all the participants. A large AD value indicates problems in the concerned Web page.

Table 2. Example Data Expressed in Success Values and Durations.

	P1		P2		P3	
Operation	**SV**	**D**	**SV**	**D**	**SV**	**D**
0 → 4	1	3	1	4	1	6
4 → 4.11	1	7	1	31	0	8
4.11 → 4.11.4	1	5	0.5	26	NA	NA
4.11.4 → 4.11.4.3	1	72	1	21	NA	NA

Note: SV represents how successfully the participant moved along the optimal path. Reaching the correct page on the first try results in 1, by the second try, 0.5, by the third attempt, 0.33, and so forth. D represents the time duration for the participant to successfully move to the correct next page, and includes any time spent recovering from poor choices.

- *Standard Deviation of Durations* (SD). This is calculated from the duration of the operation across all the participants. A large SD indicates problems.

The three aggregated measures calculated for the operations of one task, performed by three participants (see Table 2), are presented in Table 3. We can see that the operation 0 → 4 is the only one that is performed successfully by all the participants. In all the other operations, there have been some deviations from the optimal path or additional attempts.

The critical question to determine is when a certain aggregated value for some operation is so large (for AD and SD) or so small (for ASV) that the concerned Web page should be investigated more closely for usability problems. We approach this question by aggregating the values of the operations and examining the deviating values in the statistical distributions of these three measures. We calculate fractiles to specify the limits that are then used as the criteria for identifying the problematic Web pages.

The next issue is to determine the suitable fractile for each task. Selecting too large a fractile increases the risk of ignoring some problematic Web pages. Conversely, if too small a fractile is chosen, it may lead to selecting too large a set of problematic Web pages, thus increasing the need of resources for a closer examination. The suitable fractile depends on the situation. We recommend the use of probability theory to determine a suitable fractile. The probability that at least one of the three measures recognizes a Web page as problematic is 1-p^3 if all of the measures are independent of one another.

In the formula above, *p* stands for a fractile (decimal number) and 3 is the number of the measures (i.e., ASV, AD, SD). The assumption of independent measures is not exactly true, but we still use this formula as an approximation.

Table 4 presents the probabilities for four different fractiles. We can see that with the 75% fractile about 58% of the Web pages are regarded as problematic. Correspondingly, with the 95 % fractile about 14 % of the Web pages should be selected for further examination. In actuality, the probabilities are a bit smaller than indicated by the formula because the very problematic operations often are identified through more than one measure, due to some correlations between the measures.

Table 3. Aggregate Measures of the Example Data.

Operation	ASV	AD	SD
0 → 4	1.00	9.17	9.37
4 → 4.11	0.67	18.67	15.83
4.11 → 4.11.4	0.63	12.25	9.39
4.11.4 → 4.11.4.3	0.88	28.75	29.32

Note: ASV means average success value, AD means average duration, and SD means standard deviation of durations.

Table 4. Probabilities of Four Fractiles.

Fractile	75%	80%	90%	95%
Probability	1-0.75^3 = 0.578	1-0.80^3 = 0.488	1-0.90^3 = 0.271	1-0.95^3 = 0.143

The D-TEO method does not prescribe the use of any specific fractile because it depends on the situation and available resources. Instead, we offer some guidelines for selecting a fractile. A large fractile can be selected if

- the number of operations in the test tasks is large
- it can be assumed that there are only a few problems
- there is a limited amount of resources available for further examination

Conversely, a small fractile can be selected if

- the number of the operations is small
- if many problematic operations are expected to appear
- if there are sufficient resources for closer examination of the problem pages.

The final decision on whether to include a particular Web page in a set of problematic pages should be discussed with the user interface designer to avoid misinterpretations. Often, if the measures are calculated based on small samples, exceptional deviations from the standard values may appear. Thus, we emphasize that the values as such do not directly indicate which pages are problematic. The test results are best used to localize those areas in the structure of the Web site that should be analyzed more carefully.

After having determined the set of problematic Web pages, a variety of methods can be applied to identify the reasons for usability problems within specific Web pages. We suggest the use of the interaction design patterns of Tidwell (2005) and van Duyne, Landay, and Hong (2006). If inconsistencies or deficiencies are recurrent in the Web pages, stemming possibly from the applied screen design standards, changes should be extended to involve all the Web pages with similar structures. The screen design standards then should be updated correspondingly. After having made the changes, the improved Web pages can be heuristically inspected, if time and resources are allowed.

AN EXAMPLE OF THE D-TEO METHOD IN USE

In this section, we describe how the D-TEO method was used in testing the Web site of the Jyväskylä, Finland, region.[1] This is not a case study in a strict sense but rather an example for illustrating the application of the method. The description proceeds in a step-by-step manner.

Define the Goals of the Test and the User Profiles

In this first step, we determined who the stakeholders involved with the Web site were, how the results of the test were going to be used, what the stage of development (e.g., completed product, prototype, etc.) of the Web site was, and which parts of the Web site should be tested. It was concluded that the Web site was a finished prototype and that the results of the test would be used to finalize it prior to implementing as the final version. The Web site was to be tested in its entirety, a feasible task because the Web site was relatively small scale (approximately 1,300 pages) and hierarchically compact. When considering the user groups, it was thought that the Web site could be useful, for instance, for tourists planning trips to the Jyväskylä region. Their primary need would be, for example, to find accommodations in the region. We did not define any explicit user profiles.

Devise the Test Tasks and Identify the Optimal Paths

The Web site was aimed at providing information about living, working, studying, and traveling in the Jyväskylä region. The main menu covers living, municipalities, travel, services, recruitment, and events. It was decided that each of the main menu items should be selected for at least one task. We coded the Web pages corresponding to the main menu items with numerical codes instead of the URL addresses in order to make the analysis easier. Therefore, 1 = Living, 2 = Municipalities, 3 = Travel, 4 = Services, 5 = Recruitment, and 6 = Events.

There was no site map available, and hence we had to go through the paths to form a sufficient overview of the hypertext topology. Based on the structure of the Web site, we constructed a test story to include eight test tasks (see the Appendix for a description of the test tasks). We ensured the validity of the test tasks by checking that each of the tasks could be carried out and optimal paths could be specified. In this phase, some of the tasks had to be changed or made more detailed in order to fulfill the objectives above.

Organize the Test and Collect Data

When the aims of the testing were discussed with the client, it became apparent that no specific user group could be identified. Because no specific user group could be identified, the participants were randomly selected from a group of volunteer university students. Eleven native-Finnish-speaking participants participated in the study conducted in Finnish, one of whom took part in the pilot test to elaborate the test tasks. Thus, the results of 10 participants were included in the statistical analysis.

The tests were conducted in a usability laboratory at the university. The test data was collected using Windows Media Encoder, and the results were analyzed with the statistical software environment R. The time stamping was made manually.

Analyze the Data and Draw Conclusions

The data were analyzed in the manner of the instructions given in the Methods section. Problems in the user interface were localized by calculating the ASV, AD, and SD for every task and every operation. Table 5 presents the values for each of the 25 operations within the eight test tasks.

Figures 2, 3, and 4 represent the distributions of the values of ASV, AD and SD, respectively. Frequency in the histograms means the number of operations. In Figure 2, for instance, there are two operations with ASVs less than or equal to 0.4. Problems can be located in those Web pages that are involved by the operations situated at the extreme ends of the distributions (cf. the two operations in Figure 2), indicated by the circled areas.

To decide which Web pages should be selected for closer examination, a suitable fractile had to be determined. In order to avoid rounding problems in defining the critical values, the possible fractiles were 72% (18 of 25 Web pages), 76% (19/25), 80% (20/25) and 86% (21/25). Because we did not want to select too many Web pages, we used the 76% fractile, implying that the probability of recognizing a random Web page as problematic is 0.56 (1-0.76^3). Critical values for the three measures were determined according to the selected fractile. The critical limit of ASV is 0.60. The Web pages involved by the operations with smaller values were considered to be problematic. Correspondingly, the critical limits for AD and SD are 23.38 and 17.8.

Table 5. Average Success Values (ASV), Average Durations (AD) and Standard Deviations of Durations (SD) for Each Task and Each Operation.

Task	Operation			ASV	AD	SD
1	0	→	4	0.70	11.80	12.00
	4	→	4.3	0.60	34.90	23.30
	4.3	→	4.3.1	0.75	24.50	3.40
	4.3.1	→	4.3.1.1	1.00	3.12	0.60
2	0	→	6	0.20	19.60	39.00
	6	→	6.66	1.00	42.38	10.60
3	0	→	3	0.80	7.50	2.80
	3	→	3.2	0.80	16.20	17.80
	3.2	→	3.2.3	0.45	10.70	6.80
	3.2.3	→	3.2.3.7	0.70	5.00	2.00
4	0	→	4	0.50	7.70	4.00
	4	→	4.11	0.68	44.10	32.70
	4.11	→	4.11.4	0.50	13.25	5.60
	4.11.4	→	4.11.4.1	0.56	41.63	41.20
5	0	→	4	0.90	8.00	7.30
	4	→	4.11	0.75	17.30	13.90
	4.11	→	4.11.4	0.69	9.50	7.20
	4.11.4	→	4.11.4.3	0.81	23.38	24.20
6	0	→	1	0.40	5.20	4.00
	1	→	1.1	0.80	3.80	1.30
	1.1	→	1.25.6.1	1.00	7.13	3.30
7	0	→	4	0.80	13.30	7.10
	4	→	4.19	1.00	28.00	16.60
	4.19	→	4.19.5	1.00	22.50	16.70
8	0	→	0.100	0.70	21.40	18.40

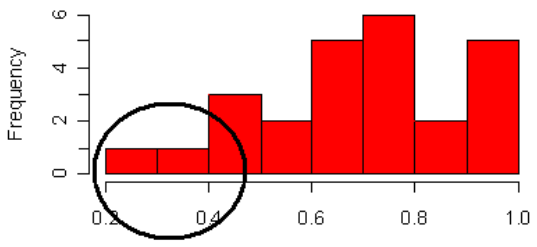

Figure 2. Histogram of average succeed values (ASV).

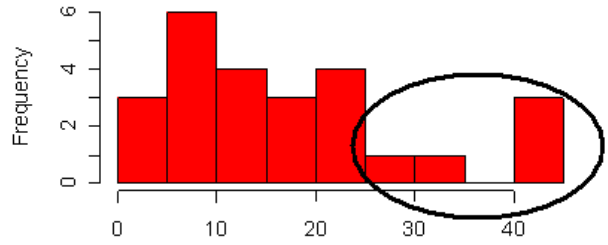

Figure 3. Histogram of average durations (AD).

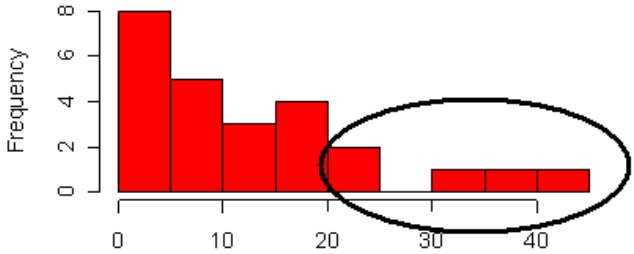

Figure 4. Histogram of standard deviations (SD).

Table 6 shows the numbers exceeding the critical limits in bold (cf. Table 5). The Column # denotes how many of the three aggregate measures suggest that the operation (Web page) is problematic. If more than one of the measures exceeds the critical limits, it is even a stronger indication of problems. The operation 4.11.4 → 4.11.4.1 appears to be a problem candidate based on all three measures. Three operations (4 → 4.3; 0 → 6; 4 → 4.11) appeared to be problem candidates based on two measures. In total, 13 out of 25 operations were selected as problem candidates.

As noted in the Methods section, the aggregated measures are correlated with one another. We calculated Spearman's rank correlation coefficient (Spearman's rho):

cor(ASV,AD) = -0.01 , $p = 0.96$

cor(ASV,SD)= -0.187, $p = 0.372$

cor(SD,AD) = 0.804 , $p < 0.001$

There is no correlation between ASD and AD, and the negative correlation between ASV and SD is not statistically significant. There is a significant correlation between SD and AD, meaning that the operations with high AD values tend to have high SD values. Because of this correlation, probability calculations are only approximations.

Hence, we distinguished 13 problem candidates for more careful consideration. Each problem candidate was mapped to a specific Web page. One of those Web pages was the Sport page (coded with 4.11.4; see Figure 5), which is the target of the operation 4.11.4 → 4.11.4.1 (cf. Table 6). When the Sport page and its UI components were analyzed more carefully interaction design patterns (see Methods section), several usability problems were found. For example, some text fields, labels, and links were not arranged in a systematic manner, the Search button was difficult to notice, and it was difficult to distinguish the labels from the text. The problems could be solved applying UI design patterns (Tidwell, 2005; van Duyne et al., 2006).

Table 6. Critical Values (Boldface) for Each Operation.

Task	Operation			ASV	AD	SD	#
1	0	→	4	0.70	11.80	12.00	
	4	→	4.3	0.60	**34.90**	**23.30**	2
	4.3	→	4.3.1	0.75	**24.50**	3.40	1
	4.3.1	→	4.3.1.1	1.00	3.12	0.60	
2	0	→	6	**0.20**	19.60	**39.00**	2
	6	→	6.66	1.00	**42.38**	10.60	1
3	0	→	3	0.80	7.50	2.80	
	3	→	3.2	0.80	16.20	17.80	
	3.2	→	3.2.3	**0.45**	10.70	6.80	1
	3.2.3	→	3.2.3.7	0.70	5.00	2.00	
4	0	→	4	**0.50**	7.70	4.00	1
	4	→	4.11	0.68	**44.10**	**32.70**	2
	4.11	→	4.11.4	**0.50**	13.25	5.60	1
	4.11.4	→	4.11.4.1	**0.56**	**41.63**	**41.20**	3
5	0	→	4	0.90	8.00	7.30	
	4	→	4.11	0.75	17.30	13.90	
	4.11	→	4.11.4	0.69	9.50	7.20	
	4.11.4	→	4.11.4.3	0.81	23.38	**24.20**	1
6	0	→	1	**0.40**	5.20	4.00	1
	1	→	1.1	0.80	3.80	1.30	
	1.1	→	1.25.6.1	1.00	7.13	3.30	
7	0	→	4	0.80	13.30	7.10	
	4	→	4.19	1.00	**28.00**	16.60	1
	4.19	→	4.19.5	1.00	22.50	16.70	
8	0	→	0.100	0.70	21.40	**18.40**	1

RELATED WORK

In this section, we make a short review of related work and discuss how our method differs from and performs among the existing methods. Our taxonomy for the review is composed of five general dimensions and three specific dimensions. The general dimensions, borrowed from Ivory & Hearst (2001), are UI, method class, method type, automation type, and effort level. UI distinguishes between WIMP (windows, icons, pointer, and mouse) interfaces and Web interfaces. Method classes are testing, inspection, inquiry, analytical modeling, and simulation. Method types include, for example, thinking aloud, log file analysis, guideline review, feature inspection and the like. Automation type is used to specify which aspects of a method are automated (i.e., capture, analysis, critique). Effort level indicates the human effort

Figure 5. The Sports page.

required by a method in use. The options are (a) minimal effort, (b) model development (M), and (c) informal (I) and formal (F). (See more about the options in Ivory & Hearst, 2001). The first specific dimension distinguishes basic concepts and constructs used to conceptualize user behavior and Web sites (e.g., user task, operation, navigation path). The second specific dimension differentiates criteria used to evaluate user interaction usability. The third specific dimension shows how the evaluators interpret the results of the evaluation.

The UI literature suggests a large array of evaluation methods (cf. Ivory & Hearst, 2001, distinguish 75 WIMP user interface evaluation methods and 57 Web user interface evaluation methods). We selected only those methods that are most relevant to our comparative review. The reviewed methods are UsAGE (Uehling & Wolf, 1995), QUIP (Helfrich & Landay, 1999), USINE (Lecerof & Paternò, 1998), RemUSINE (Paternò & Ballardin, 1999, 2000) and WebRemUSINE (Paganelli & Paternò, 2002). The results are summarized in Tables 7 and 8. The D-TEO method is included in the tables to facilitate the comparison.

In UsAGE (Uehling & Wolf, 1995) and QUIP (Helfrich & Landay, 1999), the goal is to automate the detection of serious usability problems by comparing the users' task to the task performed in the "right" manner. What constitutes the "right" manner is defined by the developer of the system. Ivory and Hearst (2001) call this kind of approach Task-Based Analysis of Log Files. In UsAGE and QUIP, the serious usability problems are localized at the level of single actions and the results are shown in a graph of the action nodes. Each node stands for an action defined to be the user action, such as menu selection or clicking the Open button. The evaluator makes the decision on usability problems, based on the graphical data. UsAGE supports only the user interfaces created with the TAE Plus user interface management system, and QUIP requires the modification of the target application source code.

USINE (Lecerof & Paternò, 1998) also deploys automated log file analysis. Ivory and Hearst (2001) call this kind of approach the Hybrid Task-Based Pattern-Matching method. USINE is an automatic usability evaluation method for Java applications, enabling the use of the task models along with log files for analyzing empirical data. Tasks are decomposed into

Table 7. Review of Alternative Methods Based on General Dimensions.

Evaluation method	UI	Method class	Method type	Automation type	Effort level
UsAGE	WIMP	Performance measurement, Usability testing	Performance measurement, Log file analysis	Capture, Analysis	IF
QUIP	WIMP	Usability testing	Log file analysis	Analysis	IF
USINE	WIMP	Usability testing	Log file analysis	Analysis	IFM
RemUSINE	WIMP	Usability testing	Log file analysis	Analysis	IFM
WebRem-USINE	Web	Usability testing	Log file analysis	Analysis	IFM
D-TEO	Web	Usability testing	Performance measurement, Log file analysis	Analysis, (Critique)	IFM

Table 8. Review of Alternative Methods based on Specific Dimensions.

Evaluation method	Concepts and constructs	Criteria	Interpretation of criteria
UsAGE	Supports only the user interfaces created with *TAE Plus* user interface management system. Each *node* stands for the action that is defined to be the *user action*, such as menu selection.	Comparing event logs for expert user and novice user. Designer is also an expert user typically. In addition to a graph, the percentage of expert nodes matched to novice nodes, ratio of novice to expert nodes, and percentage of unmatched novice nodes are analyzed.	Two files (“expert” and “novice”) are automatically compared by the tool and the results are shown graphically. Based on this, a usability analyst figures out where the usability problems exist.
QUIP	Requires the modification of the target application *source code*. Each *node* stands for the action that is defined to be the *user action*, such as menu selection or clicking the Open button.	Comparing task flows for UI designer and multiple test users. The trace of the UI designer represents the expected use. Quantitative time and trace-based information is encoded into directed graphs.	The evaluator makes the decisions based on the graphs by analyzing them manually.
USINE	Developed for usability evaluating of *WIMP interfaces*. Requires X Window environment. Requires comprehensive modeling and formalization of *user tasks*.	The accomplished tasks, the failed tasks, the never tried tasks, numerical and temporal information of the user errors, how long each task took to complete, the times for the abstract tasks, the errors occurred instances, task patterns, the test time, number of scrollbar movements, and the number of windows resized.	Evaluators make the decisions about how to improve user interface based on the simulator data. (The suggested interface changes are not drastic; e.g., there should be more difference between button and images.)

RemUSINE	Developed for remote usability evaluating of graphical Java applications. Requires the comprehensive modeling and formalization of user tasks.	**Tasks related criteria (single user session):** Completed tasks, failed tasks, never tried tasks, errors, task patterns, tasks/time, errors/time, tasks/errors, tasks/completed. **Tasks related criteria (groups of user sessions):** Completed tasks, failed tasks, never tried tasks, errors, task patterns, tasks/time, errors/time, tasks/errors, tasks/completed.	
WebRem-USINE	Java based tool developed for *remote usability evaluation* of Web sites. Requires comprehensive modeling and formalization of *user tasks*.	**Tasks related criteria (single user):** Completed tasks, missed tasks, never performed tasks, errors, task patterns, error/time, task/time, tasks/errors, tasks/completed. **Tasks related average times and standard deviations (number of users):** Total time taken by user session, number of completed tasks, number of errors, number of scrollbar movements and change dimensions events. **Pages related criteria (single user):** Visited pages, never visited pages, scroll and resize, page patterns, download time, visit time, page/access, page/scroll/resize. **Pages related criteria (number of users):** Average number of accesses in to each page, average frequency of patterns, average downloading time, average visit time.	Evaluators make the decisions based on the rich simulator data (e.g. identify what tasks create problems and what tasks are efficiently performed).
D-TEO	Developed for locating *usability problems* in *directed information search* from Web sites. Is based on defining the *optimal paths* composed of *operations* needed to navigate from the entry Web pages to the terminal Web pages in order to find the target *information items*.	**Operation-specific values (single participant):** success value, duration time. **Aggregated task-specific values (All tasks and participants):** average success value for an operation, average duration for an operation, standard deviation of durations. **Criterion for problematic Web pages:** based on fractiles.	Problematic Web pages are identified with quantitative measures. Based on this information, usability evaluators can focus their attention on those pages.

subtasks (sets of activities) that are related to each other within temporal relationships. The results derived by USINE include quite extensive numerical information about the tasks and subtasks, such as which tasks have been accomplished, which have failed, which have never been tried, user errors, and so on. Evaluators make decisions on how to improve the user interface based on the log data related to the tasks and subtasks. USINE does not enable comparing these results across study participants, so it is based on only task-related criteria of single user sessions. This means that it does not aggregate data, such as the average times of

subtasks across the participants or which tasks have been accomplished or failed across the participants. We suggest that a subtask-level comparison between the participants could bring essential knowledge to advance locating usability problems. USINE is also a rather laborious method, requiring the construction of comprehensive task models.

RemUSINE (Paternò & Ballardin, 1999, 2000), as an extension of USINE, enables capturing data remotely and comparing the results across the participants. RemUSINE employs task-related criteria for both single user sessions and groups of user sessions. In RemUSINE, evaluators make decisions regarding how to improve user interface based on the simulator and log data. RemUSINE, like USINE, was originally developed for evaluating Java applications and, as Paganelli and Paternò (2002) state, it is not suitable for evaluating how information is accessed through user interfaces in Web sites.

WebRemUSINE (Paganelli & Paternò, 2002) has its origin in USINE and RemUSINE. WebRemUSINE uses task-related criteria and page-related criteria for both single user sessions and the number of users. In WebRemUSINE, the evaluators make decisions on usability problems based on the rich simulator data. WebRemUSINE is a Java-based tool developed for remote usability evaluation of Web sites, and it requires the comprehensive modeling and formalization of user tasks. However, despite constructing a comprehensive task model and comparing the results across the participants by using rich quantitative data, neither RemUSINE nor WebRemUSINE provide an exact way to locate problematic subtasks. For example, even if the average times of the subtasks are known across the participants, the critical question about when the average time for a specific subtask is too long remains unanswered. We argue that there must be some rules, whether strict or heuristic, for helping determine some limits for crucial measures. Our suggestion is the use of the fractiles.

The D-TEO method is based on the use of two quantitative usability criteria, search success and search time, aiming at revealing how a user performs as a navigator in a web of hypertext. The former criterion is evaluated through operation-specific Success Value, calculated by the number of attempts required to find the optimal path. The latter criterion is expressed by duration time between the executions of two sequential operations. The D-TEO method is engineered to distinguish a part of a Web site that contains the most likely usability problems in directed searches. The results enable usability specialists to concentrate their efforts on making a deeper analysis of that particular area. The method does not aim at giving special guidance on the examination of what kinds of problems there are and how they are solved. Of course, if some Web page appears to be the one in which users tend to get lost, it is justifiable to expect that, for instance, the navigation, search, layout, typography, or content organization on that Web page is insufficiently designed. Similar goals are pursued by a number of interaction design patterns. What makes our method different when compared to other models is the use of heuristic rules for determining critical limits for the assessment of a certain Web page as problematic one. These rules are based on fractiles, which are selected in a situational manner. This semiautomated help is indicated in Table 7 by presenting (in parenthesis) a critique in the column of Automation type.

To summarize, the literature provides a large variety of methods for testing the usability of Web pages by observing users carrying out tasks, whether given wholly for, or as part of, their daily work. Our method differs favorably from them in the following aspects. First, the method is rather lightweight, meaning that instead of constructing a comprehensive task model, as required in USINE, RemUSINE, and WebRemUSINE, only the optimal paths for the test tasks have to be specified in D-TEO. Second, our method supports making decisions

on the limits of critical values. This is particularly beneficial in the situations where explicitly defined goals are not expressed in the quantitative measures. Finally, D-TEO distinguishes those parts of a Web site that are more problematic. As a result, scant resources can be focused on making a deeper evaluation of those areas only.

SUMMARY

The Internet holds an increasingly more important position in today's information dissemination. Diffusion of electronic commerce by enterprises and eServices provided by municipalities and government agencies have advanced the Internet as a daily means for both professionals and diverse audiences for interpersonal interaction and information searches. In attempting to serve these multiple audiences, Web site designers are challenged to meet the needs of a heterogeneous user population, from novices to heavy users, from persons having significant training for use to those with poor skills and low interest in information search. To ensure that all the people can find, with modest effort, what they are seeking, Web interfaces must meet high usability standards.

In this study, we have proposed a novel method, called D-TEO, which supports Website testing to find problematic Web pages. This semiautomated method is based on the analysis of interaction tasks in directed searches within the evaluated Web site. It provides a stepwise procedure that starts with defining the goals and user profiles and ends with analyzing the collected data and drawing conclusions. The method guides a test organizer in devising test tasks, decomposing them into elementary operations, and defining the optimal path for each task. Users are observed as they execute the test tasks and, for each operation, the time spent and the deviations from the optimal path are recorded. Using statistical methods, the collected data are analyzed to reveal which Web pages are problematic. This enables the test organizer to concentrate on more careful examination and analysis of particular small set of Web pages. Compared to most of the existing methods, D-TEO is lightweight because it does not require the comprehensive modeling and formalization of user tasks (as in Lecerof & Paternò, 1998), nor the existence of site maps. What also makes D-TEO beneficial is the support it provides regarding the situationally determined limits of critical measures for considering whether or not a Web page is included within a set of problematic Web pages.

The D-TEO method is still under research and development. At the moment, we are enhancing the method to encompass a wider variety of interaction tasks, not simply directed searches. The current procedure should be engineered toward a more fully automated mode, thus decreasing the need for human resources with required expertise in Web usability for the analysis aspect of the method. At the same time, it should be stated more clearly which type of methods—heuristic, pattern, or rule-based—are recommended as methods (e.g., Nielsen & Molich, 1990; Tidwell, 2005) applied prior to and/or following the deployment of D-TEO. In the future, we will consider how to integrate the method with qualitative methods in order to provide more flexibility for distinguishing problematic Web pages, analyzing them, and finding solutions to them. Until now, we have applied D-TEO in only small cases. To have stronger evidence of its feasibility, we will apply the method in a wider diversity of cases.

ENDNOTE

1. The Web site http://www.jyvaskylanseutu.fi was tested for the Jyväskylä region.

REFERENCES

Baresi, L., Garcotto, P., & Paolini, P. (2001). Extending UML for modeling Web applications. In *Proceedings of 34th Annual Hawaii International Conference on Systems Sciences* (pp. 1285–1294). Washington, DC, USA: IEEE Computer Society.

Batra, R., Bishu, R., & Donohue, B. (1993). Effects of hypertext topology on navigational performance. *Advances in Human Factors and Ergonomics, 19*, 175–180.

Bernard, M. (2002). *Examining a metric for predicting the accessibility of information within hypertext structures*. Doctoral dissertation, Wichita, KS, USA: Wichita State University.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: A customer-centered approach to system design.* San Diego, CA, USA: Academic Press.

Blackmon, M. H., Polson, P. G., Kitajima, M., & Lewis, C. (2002). Cognitive walkthrough for the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 463–470). New York: ACM.

Canter, D., Rivers, R., & Storrs, G. (1985). Characterizing user navigation through complex data structures. *Behaviour and Information Technology*, *4*, 93–102.

Card, S. K., Pirolli, P., Wege, M. V. D., Morrison, J. B., Reeder, R. W., Schraedley, P. K., & Boshart, J. (2001). Information scent as a driver of web behavior graphs: Results of a protocol analysis method for web usability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 498–505). New York: ACM.

Chi, E. H., Rosien, A., Supattanasiri, G., Williams, A., Royer, C., Chow, C., Robles, E., Dalal, B., Chen, J., & Cousins, S. (2003). The bloodhound project: Automating discovery of web usability issues using the InfoScent simulator. In *CHI '03: Proceedings of the SIGCHI Conference on Human Factors in Computing System* (pp. 505–512). New York: ACM.

Cockton, G., & Woolrych, A. (2002). Sale must end: Should discount be cleared off HCI's shelves? *Interactions, 9*, 13–18.

Constantine, L. L., & Lockwood, L. A. D. (1999). *Software for use: A practical guide to the models and methods of usage-centered design*. New York: ACM/Addison-Wesley Publishing Co.

Cordella, A. (2007). E-government: Towards the e-bureaucratic form? *Journal of Information Technology*, *22*, 265–274.

De Angeli, A., Sutcliffe, A., & Hartmann, J. (2006). Interaction, usability and aesthetics: What influences users' preferences? In J. M. Carroll, S. Boedker, & J. Coughlin (Eds.), *Proceedings of the 6th ACM Conference on Designing Interactive Systems 2006* (DIS2006; pp. 271–280). New York: ACM.

Dillon, A., McKnight, C., & Richardson J. (1990). Navigation in hypertext: A critical review of the concept. In D. Diaper (Ed.), *Human–Computer Interaction—INTERACT '90* (pp. 587–592). Amsterdam: Elsevier.

Duh, H. B.-L., Tan, G. C. B., & Chen, V. H.-h. (2006). Usability evaluation for mobile device: A comparison of laboratory and field tests. In M. Nieminen & M. Röykkee (Eds.), *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services 2006* (MobileHCI '06; pp. 181–186). New York: ACM.

Dumas, J. S., & Redish, J. C. (1993). *A practical guide to usability testing*. Westport, CT, USA: Greenwood Publishing Group Inc.

Freeman, M., Norris, A., & Hyland, P. (2006). Usability of online grocery systems: A focus on errors. In T. Robertson (Ed.), *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design, Activities, Artefacts and Environments* (OZCHI Vol. 206; pp. 269–275). New York: ACM.

Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., & Wichansky, A. M. (2002). Eye tracking in web search tasks: Design implications. In *ETRA '02: Proceedings of the 2002 Symposium on Eye Tracking Research & Applications* (pp. 51–58). New York: ACM.

Gwizdka, J., & Spence, I. (2007). Implicit measures of lostness and success in web navigation. *Interacting with Computers*, *19*, 357–369.

Helfrich, B., & Landay, J. A. (1999). QUIP: Quantitative user interface profiling. Retrieved on July 4, 2008, from http://www.helcorp.com/bhelfrich/helfrich99quip.pdf

Hennicker R., & Koch N. (2001). Systematic design of Web applications with UML. In K. Siau & T. Halpin (Eds.), *Unified modeling language systems analysis, design and development issues* (pp. 1–20). Hershey, PA, USA: IDEA Group Publishing.

International Organization for Standardization [ISO]. (1998, March). *Ergonomic requirements for office work with visual display terminals (VDTs), Part 11: Guidance on usability.* (Standard No. 9241-11). Geneva, Switzerland: ISO.

Isakowitz, T., Stohr, E., & Balasubramanian P. (1995). RMM: A design methodology for structured hypermedia design. *Communications of the ACM, 38*, 34–44.

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

Kim, H., & Hirtle, S. C. (1995). Spatial metaphors and disorientation in hypertext browsing. *Behaviour and Information Technology, 14*, 239–250.

Lecerof, A., & Paternò F. (1998). Automatic support for usability evaluation. *IEEE Transactions on Software Engineering, 24*, 863–888.

Lee, H., Lee, C., & Yoo, C. (1998). A scenario-based object-oriented methodology for developing hypermedia information systems. In *Proceedings of the 31st Annual Hawaii International Conference on System Sciences* (pp. 47–56). Washington, DC, USA: IEEE Computer Society.

Lee, H., & Suh, W. (2001). A workflow-based methodology for developing hypermedia information systems. *Journal of Organizational Computing and Electronic Commerce, 11*, 77–106.

Martin, R., & Weiss, S. (2006). Usability benchmarking case study: Media downloads via mobile phones in the US. In M. Nieminen & M. Röykkee (Eds.), *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services 2006* (MobileHCI'06; pp. 195–198). New York: ACM.

Masemola, S. S., & De Villiers, M. R. (2006). Towards a framework for usability testing of interactive e-learning applications in cognitive domains, Illustrated by a case study. In J. Bishop & D. Kourie (Eds.), *Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries* (SAICSIT 2006; pp. 187–197). Somerset West, South Africa: South African Institute for Computer Scientists and Information Technologists.

Matera, M., Rizzo, F., & Carughi, G. T. (2006). Web usability: Principles and evaluation methods. In E. Mendes & N. Mosley (Eds.), *Web engineering* (pp. 143–180). Berlin, Germany: Springer.

Mayhew, D. J. (1999). *The usability engineering lifecycle: A practitioner's handbook for user interface design*. San Francisco: Morgan Kaufmann Publishers.

Netcraft. (2009). January 2009 Web server survey. Retrieved January 20, 2009, from http://news.netcraft.com/archives/2009/01/16/january_2009_web_server_survey.html

Nielsen, C. M., Overgaard, M., Pedersen, M. B., Stage, J., & Stenild, S. (2006). It's worth the hassle! The added value of evaluating the usability of mobile systems in the field. In A. Moerch, K. Morgan, T. Bratteteig, G. Ghosh, & D. Svanaes (Eds.), *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles* 2006 (NordiCHI 2006; pp. 272–280). New York: ACM.

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

Nielsen J., & Mack R. (1994). *Usability inspection methods*. New York: Wiley.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People 1990* (CHI 1990; pp. 249–256). New York: ACM.

Norman K., & Chin J. (1988). The effect of tree structure on search performance in a hierarchical menu selection system. *Behaviour and Information Technology*, *7*, 51–65.

Paganelli, L., & Paternò, F. (2002). Intelligent analysis of user interactions with Web applications. In *Proceedings of the 7th International Conference on Intelligent User Interfaces* (ACM IUI'02; pp.111–118). New York: ACM.

Paternò, F., & Ballardin, G. (1999). Model-aided remote usability evaluation. In A. Sasse & C. Johnson (Eds.), *Proceedings of the IFIP TC13 7th International Conference on Human-Computer Interaction (INTERACT '99;* pp. 434–442). Amsterdam: IOS Press.

Paternò, F., & Ballardin, G. (2000). RemUSINE: A bridge between empirical and model-based evaluation when evaluators and users are distant. *Interacting with Computers*, *13*, 229–251.

Paulson, L. (2005). Building rich Web applications with Ajax. *IEEE Computer, 38*, 14–17.

Preciado, J., Lanaje, M., Sanchez, F., & Comai, S. (2005). Necessity of methodologies to model rich Internet applications. In *Proceedings of the 7th IEEE International Symposium on Web Site Evolution* (pp. 7–13). Washington, DC, USA: IEEE Computer Society.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-computer interaction*. Wokingham, UK: Addison-Wesley.

Rossi, G., & Schwabe, D. (2006). Model-based web application development. In E. Mendes & N. Mosley (Eds.), *Web engineering* (pp. 303–333). Berlin, Germany: Springer.

Saariluoma, P., Parkkola, H., Honkaranta, A., Leppänen, M., & Lamminen, J. (2009). User psychology in interaction design: The role of design ontologies. In P. Saariluoma & H. Isomäki (Eds.), *Future interaction design II* (pp. 69–86). Berlin, Germany: Springer.

Seffah, A., Donyaee, M., Kline, R. B., & Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Control, 14*, 159–178.

Schneiderman, B. (1992). *Designing the user interface: Strategies for effective human-computer interaction*. Reading, MA, USA: Addison-Wesley.

Schneiderman, B. (1998). *Designing the user interface: Strategies for effective human-computer interaction* (3rd ed.). Reading, MA, USA: Addison-Wesley.

Scholtz, J., & Laskowski, S. (1998). Developing usability tools and techniques for designing and testing web sites. Retrieved on July 2, 2008, from http://www.itl.nist.gov/iad/IADpapers/hf_web.pdf

Shackel, B. (1991). Usability: Context, framework, definition, design and evaluation. In B. Shackel & S. Richardson (Eds.), *Human factors for informatics usability* (pp. 21–38). Cambridge, MA, USA: Cambridge University Press.

Smith, P. A. (1996). Towards a practical measure of hypertext usability. *Interacting with Computers, 8*, 365–381.

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User interface design and evaluation. San Francisco*: Morgan Kaufmann Publishers.

Tidwell, J. (2005). *Designing interfaces: Patterns for effective interaction design*. Sebastopol, CA, USA: O'Reilly Media.

Uehling, D. L., & Wolf, K. (1995). User action graphing effort (UsAGE). In Proceedings of the *Conference on Human Factors in Computing Systems* (pp. 290–291). New York: ACM.

Van Duyne, D. K., Landay, J. A., & Hong, J. I. (2006). *The design of sites: Patterns for creating winning Web sites*. Upper Saddle River, NJ, USA: Prentice-Hall.

Virzi, R. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors, 34*, 457–468.

White, H., Wright, T., & Chawner, B. (2006). Usability evaluation of library online catalogues. In W. Piekarski (Ed.), *Proceedings of the 7th Australasian User Interface Conference* (Vol. 50; AUIC2006; pp. 69–72). New York: ACM.

Wixon, D., & Wilson, C. (1997). The usability engineering framework for product design and evaluation. In M. Helander, T. Landauer, & P. Prabhu (Eds.) *Handbook of human-computer interaction* (pp. 653–688). Amsterdam: Elsevier.

Wolf P., & Krcmar H. (2008). Needs driven design for eGovernment value webs. In the *Proceedings of the 41st Hawaii International Conference on System Sciences* (p 220)**.** Washington, DC, USA: IEEE Computer Society.

Authors' Note

All correspondence should be addressed to:
Juha Lamminen
Agora Center
University of Jyväskylä
PL 35, University of Jyväskylä
40014 Finland
juha.e.lamminen@jyu.fi

Human Technology: An Interdisciplinary Journal on Humans in ICT Environments
ISSN 1795-6889
www.humantechnology.jyu.fi

APPENDIX

The test tasks in the case study

1. You want to have a new hobby. What kinds of leisure activities can the Community College of the Jyväskylä region offer you in the spring 2007?

2. You promised your friend that you will take a trip together to Hankasalmi for one day in July. What kinds of events take place there on 21 July, 2007?"

3. You are going to spend the weekend in Hankasalmi and you will need a place to stay overnight. What kind of camping sites are there in Hankasalmi?"

4. Assume that you are living in Petäjävesi. What kind of sport can you exercise in the local sport clubs?

5. There is one ringette team in the Jyväskylä region. What is the name of this team?

6 The southern part of Uurainen is bit more than 20 km from the center of Jyväskylä. Which kinds of properties are there for sale in Uurainen?

7. You would like to contact the project manager of the Health and Special Sport project (TERLI). Find information about the project on the Web site of the Jyväskylä region.

8. Try to find out when the Web site of the Jyväskylä Region was opened.