

## Characterizing Usability Inspection Methods through the Analysis of a Systematic Mapping Study Extension

Luis Rivero<sup>1,2</sup>, Raimundo Barreto<sup>2</sup> and Tayana Conte<sup>1,2</sup>

<sup>1</sup>Grupo de Usabilidade e Engenharia de Software - USES,

<sup>2</sup>Instituto de Computação, Universidade Federal do Amazonas (UFAM),

Manaus, Amazonas – Brazil, 69025000

{luisrivero, rbarreto, tayana}@icomp.ufam.edu.br

### Abstract

Usability is one of the most relevant quality aspects in Web applications. A Web application is usable if it provides a friendly, direct and easy to understand interface. Many Usability Inspection Methods (UIMs) have been proposed as a cost effective way to enhance usability. However, many companies are not aware of these UIMs and consequently, are not using them. A secondary study can identify, evaluate and interpret all data that is relevant to the current knowledge available regarding UIMs that have been used to evaluate Web applications in the past few decades. Therefore, we have extended a systematic mapping study about Usability Evaluation Methods by analyzing 26 of its research papers from which we extracted and categorized UIMs. We provide practitioners and researches with the rationale to understand both the strengths and weaknesses of the emerging UIMs for the Web. Furthermore, we have summarized the relevant information of the UIMs, which suggested new ideas or theoretical basis regarding usability inspection in the Web domain. In addition, we present a new UIM and a tool for Web usability inspection starting from the results shown in this paper.

**Keywords:** Usability Inspection Methods, Web Applications, Systematic Mapping Extension, Tool Support.

### 1 Introduction

In recent years, the demand for Web applications development has grown considerably. Consequently, controlling and improving the quality of such applications has increased. Olsina et al. [22] state that when evaluating software quality, one of the most relevant aspects is usability. In the last decade, the software development industry has invested in the development of a variety of Usability Inspection Methods (UIMs) to address Web usability issues [17]. However, according to Insfran and Fernandez [11], despite the increasing number of emerging UIMs, companies are not using them. The lack of adoption of these UIMs can be one of the reasons for the low quality of Web applications.

According to Fernandez et al. [8], some papers presenting studies and comparisons of existing UIMs, like [1] and [6], show that the selection of a UIM is normally driven by the researcher's expectations. Thus, there is a need for a methodical identification of those UIMs that have been successfully applied to Web development. A literature review with a clear and objective procedure for identifying the state of art of UIMs can provide practitioners with a knowledge background for choosing a determined UIM. Moreover, this type of review can identify gaps in current research in order to suggest areas for further investigation.

Fernandez et al. [8] carried out a systematic mapping study to assess which Usability Evaluation Methods (UEMs) have been used for Web usability evaluation and their relation to the Web development process. We based our work in that study and reduced its scope from the point of view of the following research question: "*What new Usability Inspection Methods have been employed by researches to evaluate Web artifacts and how have these methods been used?*" In the systematic literature review analysis proposed in this work we have extended the study in Fernandez et al. [8]. We selected papers from that systematic mapping that address new Usability Inspection Methods. We have created new research sub-questions to thoroughly analyze each of the selected papers. This new analysis has allowed us to provide further details on the actual state of Usability Inspection Methods for the Web. Additionally, we present practitioners and researchers with useful information of the new UIMs that have emerged.

The extended analysis of the systematic mapping has allowed us to outline important issues, which are, among others: (a) at what extent the inspection process is automated; (b) which is the most commonly evaluated artifact; (c) what type of applications are being evaluated by UIMs; and (d) which basic Usability Evaluation Methods or Technologies are used by UIMs as background knowledge to carry out the inspection process. Furthermore, we have extracted relevant information from each of the selected methods and we have summarized it to provide further background information. Finally, we have also used the results in this systematic mapping extension to propose a new UIM and a tool for the usability evaluation of Web Mockups.

This paper is organized as follows. Section 2 presents the background of Usability Inspection Methods and introduces readers to the scope of this work. Section 3 describes how we extended the systematic mapping of Fernandez et al. [8]. Section 4 presents the results from the systematic mapping extension that answers the research question. In Section 5 we provide a summary of all the new UIMs for Web applications, while Section 6 discusses our findings and their implications for both research and practice. In Section 7 we introduce a new technique and a tool for the inspection of Web mockups, based on the results from this extension. Finally, Section 8 presents our conclusions and future work.

## 2 Background

In this section we provide a background of usability inspection methods, presenting some core ideas and the main methods for usability inspection. Finally, we justify the need for the proposed systematic mapping extension.

### 2.1 Usability Evaluation

The term usability is defined in the ISO 9241-11 [12] as “*the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use*”. According to Mendes et al. [18], a Web application with poor usability will be quickly replaced by a more usable one as soon as its existence becomes known to the target audience. Insfran and Fernandez [11] discuss that many usability evaluation methods (UEMs) have been proposed in the technical literature in order to improve the usability of different kinds of software systems.

UEMs are procedures composed by a set of well-defined activities that are used to evaluate the system's usability [8]. There are two main broad categories of UEMs: user testing and inspections [27]. A user testing is a user centered evaluation, in which empirical methods, observational methods, and question techniques can be used to measure usability. User testing captures usage data from real end users while they are using the product (or a prototype) to complete a predefined set of tasks. The analysis of the results can provide testers with information to detect usability problems and to improve the system's interaction model. Inspections, on the other hand, are evaluations that make use of experienced inspectors to review the usability aspects of the software artifacts. The inspectors base their evaluation in guidelines that check the system's level of achievement of usability attributes. Based on this evaluation, inspectors can predict whether there will be a usability problem or not.

When carrying out a usability evaluation, both approaches have advantages and disadvantages. User testing can be used to address many usability problems affecting real end users. However, according to Matera et al. [17], it may not be cost-effective as it requires more resources to cover the different end user profiles. Furthermore, as it needs full or partial system implementation, it is mostly used in the last stages of the Web development process. Inspections, on the other hand, can be used in early stages of the Web development process, and require fewer resources which lower the cost of finding usability problems [27].

### 2.2 Usability Inspection Methods

According to Matera et al. [17], the main Usability Inspection Methods (UIMs) are the Heuristic Evaluation [19] and the Cognitive Walkthrough [23]. The Heuristic Evaluation, proposed by Nielsen [19], assists the inspector in usability evaluations using guidelines. The evaluation process consists of a group of evaluators who examine the Graphical User Interface (GUI) using heuristics, which are a collection of rules that seek to describe common properties of usable interfaces. Initially, inspectors examine the GUIs looking for problems and if one is found, it is reported and associated with the heuristics it violated. Afterwards, inspectors can rank problems by their degree of severity.

The Cognitive Walkthrough, proposed by Polson et al. [23], is a method in which a set of reviewers analyze if a user can make sense of interaction steps as they proceed in a pre-defined task. It assumes that users perform goal driven explorations in a user interface. During the problem identification phase, the design team answers the following questions at each simulated step: Q1) *Will the correct action be made sufficiently evident to the user?*

Q2) Will the user connect the correct action's description with what he or she is trying to do? Q3) Will the user interpret the system's response to the chosen action correctly? Each negative answer for any of the questions must be documented and treated as a usability problem.

According to Rocha and Baranauska [27], there are other usability inspection methods which are based in guidelines and consistencies. Guidelines Inspections, which are similar to the Heuristic Evaluation, have their own usability set of rules and need experienced inspectors to check them. A consistency inspection verifies if an interface is consistent with a family of interfaces. Another technique, proposed by Zhang et al. [29], is the Perspective-based Usability Inspection in which every section of the inspection focuses in a subset of questions according to the usability perspective. These techniques can be used to increase the system's usability and therefore its quality.

### 2.3 Motivation for a Systematic Mapping Extension about UIMs

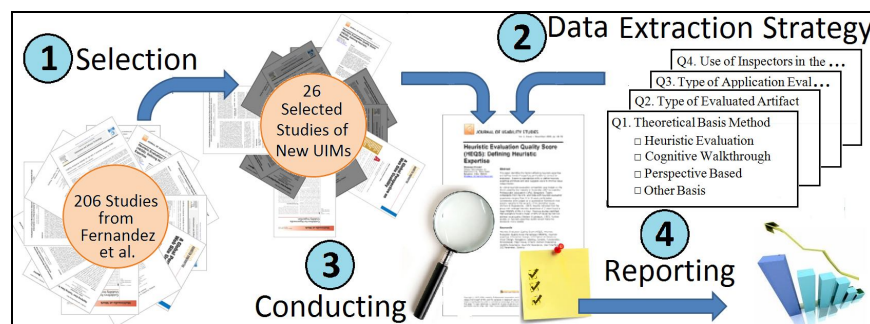
There have been several studies concerning Usability Evaluation Methods. However, there are few studies regarding the state of art of Usability Evaluation Methods. Insfran and Fernandez [11] investigated which usability evaluation methods have been used in the Web domain through a systematic mapping study. Later, Fernandez et al. [8] published an expanded version of the systematic mapping by adding new studies and further analysis. The goal of both researches was to provide a preliminary discussion about how UEMs have been applied in the Web domain.

Regarding Usability Inspection Methods, which are a subset of Usability Evaluation Methods, we found out that there have been many studies presenting and evaluating UIMs for the Web (e.g. [2], [5] and [3]). Nonetheless, we are not aware of any study discussing and analyzing the state of art of such methods. There is a need for a more specific investigation on Usability Inspection Methods to identify research opportunities. Consequently, we have extended part of the study available in Fernandez et al. [8]. We have selected the papers describing new UIMs and thoroughly analyzed them using new data extraction criteria in order to present further and useful information about new UIMs.

## 3 Procedure

This systematic mapping extension was based in the systematic mapping executed by Fernandez et al. [8]. With this extension, we aim to answer more questions regarding the state of art of UIMs for the Web.

Figure 1 shows the four stages for performing this systematic mapping extension: selection of primary studies, data extraction strategy, conducting stage and reporting. In the next subsections we describe the activities previous to the reporting stage. We will thoroughly describe the reporting stage in Section 4.



**Figure 1:** Stages for performing our systematic mapping extension

### 3.1 Selection of Primary Studies

In Fernandez et al. [8] we found a total of 206 papers which referred to UEMs. In order to better address their research question, the authors created a set of seven research sub-questions each paired with a set of possible answers. Table 1 shows the questions Fernandez et al. [8] used in order to divide the extracted papers into categories for analysis purposes.

**Table 1:** Research sub-questions and possible answers from Fernandez et al. [8] regarding UEMs

Research Sub-Question	Possible Answers	
<b>Q1. Origin of the UEMs employed</b>	<b>a) New</b>	b) Existing
Q2. Underlying usability definition of the UEMs employed	a) Standard	b) Ad-Hoc
<b>Q3. Types of UEMs employed</b>	a) User Testing c) Simulation e) Inquiry	<b>b) Inspection</b> d) Analytical Modeling
Q4. Empirical Validation of the UEMs	a) Survey c) Experiment	b) Case Study d) No
Q5. Type of evaluation performed by the UEMs employed	a) Manual	b) Automated
Q6. Phase(s) and Web artifacts in which the UEMs are applied	a) Requirements c) Implementation	b) Design
Q7. Feedback provided by the UEMs	a) Yes	b) No

Our goal with this systematic mapping extension is to analyze the current use of new UIMs in the Web domain by answering the following research question: “*What new usability inspection methods have been employed by researchers to evaluate Web artifacts, and how have these methods been used?*” If we look at the terms in bold in Table 1, we can see that papers having answers “new” and “inspection” to questions Q1 and Q3 respectively are related to our research question. Therefore, from the total of 206 papers, cited in Fernandez et al. [8], we selected those describing the application of new inspection methods.

Since our research question is too broad to thoroughly describe how UIMs are being applied, we also decomposed it into sub-questions for it to be better addressed. A sub-question, in this paper, is a question related to our main question that could give more details regarding the state of art of UIMs. Table 2 shows these research sub-questions and their motivation. Furthermore, we only selected studies describing UIMs in detail and their complete inspection process. Consequently, we discarded papers that met at least one of the following exclusion criteria:

- Papers presenting usability problems and no methodology to identify them.
- Papers describing only ideas for new research fields.
- Papers presenting techniques with no description of their execution process.

**Table 2:** Research sub-questions and motivations for our systematic mapping extension

Research Sub-Question	Motivation	Possible Answers
Q1. Theoretical Basis Method	To discover whether the Usability Inspection Methods for the Web have been developed considering well known Generic Usability Inspection Methods or whether they have been using new theoretical bases.	(a) Heuristic Evaluation (b) Cognitive Walkthrough (c) Perspective Based (d) New Basis
Q2. Type of Evaluated Artifact.	To discover what artifact is the most commonly evaluated in Usability Inspection Methods for the Web.	(a) HTML Code (b) Model (c) Application/Prototype
Q3. Type of Web Application Evaluated by the Inspection Method	To discover whether the Usability Inspection Methods for the Web have been crafted to discover generic usability problems or usability problems of a specific type of Web application.	(a) Generic (b) Specific
Q4. Use of Inspectors in the Inspection Process	To discover whether the Usability Inspection Methods for the Web have been automated to a point where inspectors are no longer necessary.	(a) Yes (b) No

### 3.2 Data Extraction Strategy

In order to extract data for further analysis, we used the answers provided in each of the research sub-questions we defined. This strategy guarantees that all selected papers will be objectively classified. In the following, we explain in more detail the possible answers to each research sub-question.

With regards to Q1 (Theoretical Basis Method), a paper can be classified in one of the following answers: (a) Heuristic Evaluation: if the presented method is a UIM based on Nielsen's [20] Heuristic Evaluation; (b) Cognitive Walkthrough: if the presented method is a UIM based on Polson's [23] Cognitive Walkthrough; (c) Perspective Based: if the presented method is a UIM based on different Perspectives (such as Zhang et al. [29] Perspective-based Usability Inspection); and (d) New Basis: if the presented method is based on a new approach.

With regards to Q2 (Type of Evaluated Artifact), a paper can be classified in one of the following answers: (a) HTML: if the presented method analyzes code in order to find usability problems; (b) Model: if the presented method analyzes models in order to find usability problems; and (c) Application/Prototype: if the presented method analyzes applications or prototypes in order to find usability problems.

With regards to Q3 (Type of Web Application Evaluated by the Inspection Method), a paper can be classified in one of the following answers: (a) Generic: if the presented method can be applied to any type of Web applications; and (b) Specific: if the presented method evaluates specific types of Web applications.

With regards to Q4 (Use of Inspectors in the Inspection Process), a paper can be classified in one of the following answers: (a) Yes: if the presented method needs the intervention of experienced inspectors in order to find usability problems; and (b) No: if the presented method is fully automated and can carry an usability evaluation without the supervision of experienced inspectors.

### 3.3 Conducting Stage

From the initial set of 206 papers in Fernandez et al. [8], we selected 37 papers by using their answers to questions Q1 and Q3. However, 5 papers were unavailable for download forcing us to reduce the initial set to 32.

After reading each study, we discarded 6 studies taking into account the exclusion criteria defined in Subsection 3.1. Therefore, we identified 26 papers describing new UIMs for Web applications and their complete inspection process which represent 15.53% of the papers in [8]. Furthermore, 6 of the selected papers were published in a journal while 20 papers were published in a conference.

The Selected Primary Studies List (See Appendix A) at the end of this paper shows the 26 selected papers that we analyzed in this systematic mapping extension. We used the data extraction strategy defined in the previous subsection to address the current stage of UIMs for the Web. We discuss our results in Section 4.

## 4 Results

Table 3 shows the classification and summary of the primary studies. We based these results on counting the number of answers to each research sub-questions. Readers must take note that questions Q1 and Q2 are not exclusive. A study can be classified in one or more of the answers of these questions and, therefore, the sum of the percentages is over 100%. Furthermore, it is noteworthy that we have not used any tool support to assess the data, which can be consider a validity threat to the results of this systematic mapping extension. Nevertheless, the results were checked by the authors of this paper.

**Table 3: Results of the classification analysis**

		Paper																								Total Studies	%			
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24			25	26	
Q01	A	x									x	x	x					x			x					x			7	26.92
	B						x	x	x						x		x												5	19.23
	C									x				x											x		x		4	15.38
	D		x	x	x	x					x				x		x			x	x	x	x	x	x		x	x	15	57.69
Q02	A		x																			x		x				x	4	15.38
	B				x															x			x		x				4	15.38
	C	x			x		x	x	x	x	x	x	x	x	x	x	x	x		x	x		x			x	x		20	76.92
Q03	A		x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	23	88.46
	B	x				x																					x		3	11.54
Q04	A	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		24	92.31	
	B		x																									x	2	7.69

### Categorization of Primary Studies

**Q01.- Theoretical Basis Method:** (a) Heuristic Evaluation (b) Cognitive Walkthrough (c) Perspective Based (d) Other Basis

**Q02.- Type of Evaluated Artifact:** (a) HTML (b) Model (c) Application/Prototype

**Q03.- Type of Web Application Evaluated by the Inspection Method:** (a) Generic (b) Specific

**Q04.- Use of Inspectors in the Inspection Process: (a) Yes (b) No**

In the following subsections we present the analysis of the results for each research sub-question, describing how the analyzed methods have been used. We will reference the papers listed in the Selected Primary Studies List (See Appendix A) by using an S followed by the paper's sequence number.

### 4.1 Theoretical Basis Method

The results for sub-question Q1 (Theoretical Basis Method) revealed that, with respect to already known usability inspection methods, around 27% of the reviewed papers were based on Nielsen's [20] Heuristic Evaluation, around 19% were based on Polson's [23] Cognitive Walkthrough, and around 15% were based on Perspectives.

Papers based on Nielsen's [20] Heuristic Evaluation, mainly focused on better describing or self-explaining how or in which situation each heuristic could be applied. Papers S11 and S24 provide inspectors with a set of examples in order to better address usability problems while using heuristics. Papers S12 and S20 make improvements in Nielsen's original set of heuristics by combining them with other methods.

Regarding the use of the Cognitive Walkthrough, two new inspection methods were developed: Blackmon's CWW described in papers S06, S07 and S08; and Filgueiras' RW described in S14. Moreover, paper S16 validates how eye tracking and cognitive Walkthrough can be applied together in a study carried out by Habuchi et al. (S16).

Studies S09, S12, S23 and S25 make use of perspectives to help focus on each usability attribute when carrying out the inspection. This basis was applied along with other methodologies to guarantee better performance.

The remaining 58% of the studies reported that other techniques are being proposed to focus on the Web domain. Most studies describe techniques based on heuristics specifically proposed for the Web domain. Another approach is the transformation of usability principles found in standards like ISO 9241-11 [12].

These results may indicate that new approaches are being designed. Nevertheless, many of the proposed UIMs still make use of general approaches and were adapted to the Web domain. In fact, some of the selected papers (S12, S20, S23 and S25), that make use of adapted methods, combine two or more approaches to improve inspectors' performance during the usability evaluation.

## 4.2 Type of Evaluated Artifact

The results for sub-question Q2 (Type of Evaluated Artifact) revealed that most UIMs analyze applications or prototypes. Around 77% of the reviewed papers reported a UIM that made use of a functional prototype/system or a paper based prototype. Inspectors carry out the evaluation process by analyzing the interaction provided by the

prototype or product while executing a task on it. These results may indicate that unless the prototype was paper based, most of the usability inspection techniques are being used in stages where at least part of the system is functional.

The remaining papers are divided into HTML analysis (15%) and Model analysis (15%). Papers S02, S21, S23 and S26 describe automated techniques in which HTML code can be analyzed in order to, based on usability guidelines and patterns, discover usability problems. The selected papers show that HTML analysis is generally used in the final stages of Web development where the actual system's code is available for inspection. Automating HTML analysis can reduce the cost of this type of inspection. However, that does not imply that correcting the usability problems will be inexpensive, since they will be found in the last stages of the development process. Studies S04, S18, S21 and S23, show UIMs that make use of models to identify usability problems. Models aid to represent human computer interaction while the inspector verifies if the model meets rules for interaction in the Web domain. This type of UIMs can help find usability problems on early stages of the development process, thus reducing the cost of correcting them. Nevertheless, according to the selected studies, these UIMs still make require usability experts to analyze Web artifacts.

#### 4.3 Type of Web Application Evaluated by the Inspection Method

The results for sub-question Q3 (Type of Web Application Evaluated by the Inspection Method) revealed that around 88% of the selected studies could be applied to any Web application. For instance, papers S06, S12 and S14 described UIMs that adapted generic usability inspections to the Web domain. Studies S18, S19 and S25 are examples of techniques that created their own Web usability features but still remained broad to analyze any type of Web application.

The remaining papers (S01, S05 and S24), around 12% of the selected studies, focused the proposed UIMs to a specific type of Web application in order to be more effective in identifying usability problems. Allen et al. (S01) describe a paper based technique to evaluate medical Web applications. Basu (S05) proposes a new framework to evaluate e-commerce applications in order to find usability problems based on the applications' context of use. In paper S24, Thompson and Kemp evaluate Web applications developed within the Web 2.0 context.

#### 4.4 Use of Inspectors in the Inspection Process

The results for sub-question Q4 (Use of Inspectors in the Inspection Process) revealed that around 92% of the selected studies required inspectors to carry out the UIMs evaluation process. The remaining 8% did not require any inspectors at all. A further analysis shows that there is a relationship between the UIM's evaluated artifact and the degree of automated process. We noticed that HTML analysis could be fully automated but other type of UIMs still depend on inspectors. Despite using HTML analysis, papers S21 and S23 describe partially automated UIMs. These studies propose hybrid types of UIMs and consequently require inspectors to verify the overall inspection process.

### 5 Analysis of the New UIMs for Web Applications

We identified a total of 21 new approaches for Web Usability Inspection after reviewing each of the papers in this systematic mapping extension. Table 4 shows the name of these UIMs and their theoretical basis. Furthermore, the column "Evolution Paper", indicates whether the paper is describing the method for the first time (No – Main Paper), or if it describes an evolution of the method (Yes – Short Description of the Extra Contribution). For instance, Paper S01 describes the Paper-Based Heuristic Evaluation for the first time, while Paper 08 describes a new version of the Cognitive Walkthrough for the Web method by adding tool support. It is noteworthy that we have organized the papers according to the year in which they were published.

**Table 4:** Method described by the selected papers

Paper	Year	Name of the Technique/Technology/Method for Usability Inspection	Theoretical Basis	Evolution Paper
S10	1998	Adapting the Heuristic Evaluation	Heuristic Evaluation	No - Main Paper
S22	1999	AT Based Inspection - SUE	Other Basis	Yes - Usability Problems Introduction

Paper	Year	Name of the Technique/Technology/Method for Usability Inspection	Theoretical Basis	Evolution Paper
S13	2001	AT Based Inspection - SUE	Other Basis	No - Main Paper
S21	2002	Automatic Reconstruction of the Underlying Interaction Design of Web Applications	Other Basis	No - Main Paper
S06	2002	Cognitive Walkthrough for the Web	Cognitive Walkthrough	No - Main Paper
S07	2003	Cognitive Walkthrough for the Web	Cognitive Walkthrough	Yes - Evolution
S05	2003	Framework For Measuring Web Usability	Other Basis	No - Main Paper
S26	2004	Automated Evaluation of Web Usability and Accessibility	Other Basis	No - Main Paper
S11	2004	Heuristic Evaluation Plus	Heuristic Evaluation	No - Main Paper
S08	2005	Cognitive Walkthrough for the Web	Cognitive Walkthrough	Yes - Adding Tool Support
S23	2005	Comprehensive Model for Web Sites Quality	Perspective Based, Other Basis	No - Main Paper
S25	2005	Milano Lugano Evaluation Method	Perspective Based, Other Basis	No - Main Paper
S04	2005	Model-Based Usability Validator	Other Basis	No - Main Paper
S03	2006	AT Based Inspection - SUE	Other Basis	Yes - Adding Tool Support
S01	2006	Paper-Based Heuristic Evaluation	Heuristic Evaluation	No - Main Paper
S19	2006	Software Measurement Ontology	Other Basis	No - Main Paper
S09	2007	Milano Lugano Evaluation Method	Perspective Based, Other Basis	Yes - Criteria for Assessing Quality of UIMs
S16	2008	Combining CWW with Eye-Tracking	Cognitive Walkthrough	No - Main Paper
S17	2008	Heuristic Evaluation Quality Score	Heuristic Evaluation	No - Main Paper
S15	2008	Methodology for Assessing Cultural Markers for Web Application	Other Basis	No - Main Paper
S20	2009	Assessment Methodology for Usability of Web-Based Information Systems - UWIS	Heuristic Evaluation, Other Basis	No - Main Paper
S24	2009	Heuristics Evaluation for Web 2.0 sites	Heuristic Evaluation	No - Main Paper
S02	2009	HTML analyzer for the study of web usability	Other Basis	No - Main Paper
S18	2009	Integrating Usability Requirements into Model Driven Engineering	Other Basis	No - Main Paper
S14	2009	Recoverability walkthrough	Cognitive Walkthrough	No - Main Paper
S12	2009	Web Design Perspective Based Usability Evaluation	Heuristic Evaluation, Perspective Based	No - Main Paper

In order to obtain complete data from each of the new Usability Inspection Methods for Web application, we designed a Data Extraction Form. We used this form to document the characteristics of each of the 21 encountered UIMs for the Web. Table 5 shows an example of this form in which we filled the information for the Cognitive Walkthrough for the Web method (Papers S06, S07 and S08). Readers must note that the complete list of the data



extraction forms is available in [24]. In the next paragraphs, we will summarize the information gathered using the Data Extraction Forms for each of the 21 UIMs for the Web.

**Table 5:** Filled data extraction form for the Cognitive Walkthrough for the Web (S06) method

Method Detailed Extraction Form	
<b>Paper:</b>	<p><b>Main Paper</b> - Cognitive Walkthrough for the Web (S06).</p> <p><b>Related Paper 1</b> - Repairing usability problems identified by the cognitive walkthrough for the Web (S07).</p> <p><b>Related Paper 2</b> - Tool for accurately predicting Website navigation problems, non-problems, problem severity, and effectiveness of repairs (S08).</p>
<b>Name:</b>	Cognitive Walkthrough for the Web (CWW).
<b>Is the UIM for the Web based in an existent method?</b>	Yes. The method is based in the Cognitive Walkthrough [23].
<b>Theoretical Basis of the UIM for the Web:</b>	<p>The method considers two theoretical basis to propose a new usability evaluation method:</p> <p><b>The CoLiDeS model</b> [14] - Comprehension-based Linked model of Deliberate Search. This model divides the human computer interaction in two phases: (a) the attention phase, in which the user selects a sub-region of the interface that is related to his/her goals; and (b) the action selection phase, in which the user chooses to act over a part of the sub-region that was previously chosen.</p> <p><b>The Latent Semantic Analysis - LSA mathematic technique</b> [15]. This technique allows the estimation of the semantic similarity between the texts in the interface with the user goals.</p>
<b>Description of the UIM for the Web:</b>	<p>The Cognitive Walkthrough for the Web uses the same questions as the Cognitive Walkthrough:</p> <p>Q1) Will the correct action be made sufficiently evident to the user?</p> <p>Q2) Will the user connect the correct action's description with what he or she is trying to do?</p> <p>Q3) Will the user interpret the system's response to the chosen action correctly?</p> <p>The main difference relies in Q2, which is expanded into two questions:</p> <p>Q2A) Will the user link the sub-region of the page to his/her goal, using the available information and his/her understanding of the page layout correctly?</p> <p>Q2B) Will the user link his/her goal with the correct widget in the chosen sub-region using labels and other type of descriptive information?</p> <p>The Cognitive Walkthrough for the Web can be applied in three main aspects:</p> <p>More complex and elaborated goals.</p> <p>What matters is what the user read and not what was written.</p> <p>The user focuses in sub-regions which description is similar to his/her goal.</p>

Method Detailed Extraction Form	
<b>Execution Process of the UIM for the Web:</b>	1) Compile a set of real user Goals and Intentional Selections. These goals must be described using from 100 to 200 words including: personas and their goals in the Web site. 2) Use the LSA to estimate the semantic similarity between the goals, labels and links. 3) Identify labels and links that are confusing of unfamiliar to the users. 4) When a link or label that is not necessary to achieve a goal possesses high LSA levels (above 80%), it must be marked as a competitive link or label. 5) Rename links or labels with a high level of LSA.
<b>Does the UIM for the Web have tool support?</b>	Yes. The LSA calculus can be done automatically in a Web application: <a href="http://lsa.colorado.edu/">http://lsa.colorado.edu/</a> . However, the overall inspection process is not automated.
<b>Does the UIM for the Web provide feedback?</b>	Yes. The information of the LSA can allow the identification of competitive links and help rename them. However, it is necessary to replicate the LSA evaluation to verify if the competition among the links and labels was reduced.
<b>Has the UIM for the Web been empirically evaluated?</b>	Yes. There have been three empirical studies where the authors compare the prediction of usability problems using the CWW and the mean of participants who chose the correct link/label in the first click: 1st Empirical Study: A Web page with 16 links and another with 32 links. These links were related to 177 goals, which were carried out by 6 subjects. 2nd Empirical Study: The same Web pages as the 1st empirical study. These pages were related to 63 goals, carried out by 23 subjects. 3rd Empirical Study: A Web page with 51 links, related to 32 goals. These goals were carried out by 20 subjects.
<b>Limitations of the UIM for the Web</b>	Each goal must be described using at least 100 words and a maximum of 200. The CWW is not ready to overcome failed searches, because it assumes that the user will be guided correctly. The CWW gives more priority to the analyses of the Web page's text rather than its distribution among it.

The following paragraphs summarize each of the 21 encountered UIMs for the Web. We have chronologically ordered the Usability Inspection Method from this systematic mapping extension according to the date of the main paper in which it was described.

In paper S10, Burton and Johnston adapt the Heuristic Evaluation [20] in order to perform usability evaluations in Web applications. In this method they propose questions that could be used to find usability problems in Web applications. Burton and Johnston also performed an empirical study in which the questions were used to address the usability of Web systems. Furthermore, the results were stored in a repository in order to be used in the development of further systems.

In paper S13, Costabile and Matera describe the Abstract Tasks (AT) Based Inspection. This UIM is a guideline inspection method based on the Hypermedia Design Model [10] that suggests which Web application's objects should be focused during an evaluation. Costabile and Matera describe 40 Abstract Tasks (AT) that are composed by: title and classification code, focus of action, intention, description, output and example. During the evaluation process, the inspectors identify applications components and which ATs must be verified. Afterwards, the inspectors do activities related to the application while writing down their observations. A report is then generated in order to combine all inspectors' observations.

Paper S06 describes Blackmon's Cognitive Walkthrough for the Web (CWW). This method, also described in S07 and S08, is an evolution of the CW - Cognitive Walkthrough [23]. The CWW uses the CoLiDeS model [14] to

divide interaction in two phases: attention and selection of action. It also uses the Latent Semantic Analysis (LSA) mathematical technique [15] to estimate semantic similarity. Blackmon et al. (S06) state that the CWW uses the same evaluation process as the CW. The evaluation process consists of a set of questions to be asked while simulating step by step user behavior. However the most critical questions for successful navigation are: Q2a) *Will the user connect the correct sub-region of the page with the goal using heading information and her understanding of the sites page layout conventions?* and Q2b) *Will the user connect the goal with the correct widget in the attended to sub-region of the page using link labels and other kinds of descriptive information?* These questions are used to identify problematic, competitive and confusing links. Inspectors must use the LSA to categorize and rename them in order to solve usability problems.

Paper S21 describes the Automatic Reconstruction of the Underlying Interaction Design of Web Applications. In that paper, Paganelli and Paterno suggest the automatic production of a task model based on the analysis of HTML code. Initially, the code is scanned, looking for interaction elements (links, buttons, among others) and group elements (forms, field sets, among others). The analyzer has a set of embedded rules that transform the Web system into a task model. This model is used as input data into another system that allows the identification of the differences between: (a) the way in which the Web system is expected to be used; and (b) the way in which the system is really used by its users. The obtained results are then stored in a log and can be used by usability experts to make decisions in order to improve the usability of the system.

The Framework for Measuring Web Usability was proposed by Basu in paper S05. This UIM is specifically developed for e-commerce Web sites which have three dimensions that can be used to measure usability: e-business context, supported online processes and usability defining factors. The inspection method aids in identifying the purpose of the Web application's processes: search, authentication, valuation and price discovery, payment, logistics, customization and usability. Each factor must be weighted depending on the processes that the Web application intends to support. Usability problems can be found by analyzing factors' weights and if the application meets the usability characteristics to support them.

Chattratchart and Brodie proposed the Heuristic Evaluation Plus in paper S11. In this method they evolve the Heuristic Evaluation [20] by instantiating it. In paper S11, the authors perform a usability inspection of Web pages of shopping malls. In order to do this they measure three aspects: (a) thoroughness, which is the number of real identified problems divided by the number of real problems that exist; (b) validity, which is the number of real identified problems divided by the number of identified problems; and (c) effectiveness which is the product of thoroughness and validity. In S11, a problem is an issue identified by the inspector that might or might not be a usability problem, while a real problem is a problem that can prejudice usability. In this sense the identified real problems are a subset of the identified problems.

In paper S26, Vanderdonck et al. proposed the Automated Evaluation of Web Usability and Accessibility. They present an automated usability and accessibility inspection tool that analyzes HTML code in search for usability problems. In order to do so, the tool possesses a set of guidelines of usability and accessibility. However, due to the limitations of HTML code, other inspection approaches must be used to measure the satisfaction of the user, consistency and organization of the information. The method can be used during the implementation phases and it offers information regarding the encountered usability problems and how to fix them.

Paper S04 proposes the Model-Based Usability Validator. In that paper, Atterer and Schmidt, propose to include usability attributes within web models so that these attributes can be addressed by tool validators. In order to do this, they suggest the inclusion of the following attributes in web models: timing, purpose of the site and target group. When measuring these attributes the validator should verify if the guidelines for each attributes are violated. For instance in the timing attribute the guidelines are: (a) overall contact time of a user with the site; (b) contact time per visit; (c) how long will the user need for the main tasks; and (d) what is the maximum time for delivery of a page.

In paper S23, Signore proposes the Comprehensive Model for Web Sites Quality. This UIM is a hybrid method which makes use of HTML and model analysis. It uses five perspectives in order to evaluate usability: correctness, presentation, content, navigation and interaction. The correctness perspective is directly related to code quality while the other perspectives are related to the users' opinion. During the HTML analysis, a tool verifies correctness problems. During the model analysis, inspectors relate correctness problems to other perspectives and identify usability problems. The lessons learnt from each inspection are kept in a repository to be used in future inspections or company decisions.

Studies S09 and S25 present the Milano Lugano Evaluation Method (MiLE+). Bolchini and Garzotto proposed this method, based on the MiLE [4] and the SUE [16] usability evaluation methods, to evaluate Web application under two perspectives: technical and user experience. In order to evaluate the technical perspective, inspectors use a total of 81 heuristics divided in four aspects: navigation, content, technology and interface design. Regarding the user experience perspective, inspectors use a total of 20 indicators divided in three aspects: experience in content,

experience in navigation and experience in operational flow. When using the proposed UIM, inspectors create a set of scenarios and use the set of indicators to verify the application dependent usability characteristics within it. In order to verify the application independent usability characteristics inspectors use the set of heuristics.

In S01, Allen et al. present the paper-based Heuristic Evaluation, which is a UIM specifically designed for assessing the degree of usability of medical Web applications. The authors suggest the use of five heuristics: consistency, match, minimalism, memory and language. These heuristics are the result of the combination of Nielsen's [19] Heuristics and Shneiderman's [28] Golden Rules. The inspection method is similar to the one used in the Heuristic Evaluation. Nevertheless, in the Paper-Based Heuristic Evaluation, inspectors produce mockups or Web application's print screens to carry out the inspection.

Moraga et al. (S19) proposed the Software Measurement Ontology which is based in the recompilation of used terms regarding usability. The authors state that there are two types of Web applications: (a) first generation: static Web pages with fixed functionalities and structure; and (b) second generation: Web page composed of portlets, which are mini-applications with limited functionalities and that can be edited in order to compose a bigger system. The usability evaluation is based in the concept of choosing the best portlets according to their degree of usability. Inspectors use the ontology and its measurable concepts to identify usability requirements. These requirements are measured using quality standards that have been proposed by the authors. Moraga et al. have also propose four indicators to measure the usability of the portlets: understandability, ability to learn, customization and compliance.

In paper S15, Fraternaly and Tisi projected a Methodology for Identifying Cultural Markers for Web Applications. That UIM proposes to aid in the identification of which usability factors affect which end users depending on their cultural background. In order to do that, the authors describe how to run a series of studies to rank Web applications and identify the cultural markers that have a direct impact over the perceived usability. Fraternaly and Tisi (S15) discovered that the organization, colors and navigational patterns of the Web application must reflect the end user's culture, or the Web application will not be usable. An inspector can carry out this UIM by comparing Web applications of the same type within different cultural contexts. The comparison must help identify checkpoint that can become cultural markers. In order to validate the checkpoints a test with real end users must be used. The cultural markers can be used to suggest improvements, or as usability references to carry out new inspections.

Paper S16 suggests combining the CWW with Eye-Tracking. In that paper, Habuchi et al. relate the CWW (S06) with the Eye Tracking method [7]. The Eye Tracking is the process of measuring either the point of gaze or the motion of an eye relative to the head, in order to obtain data about the effects of changes in the visual presentation of information [7]. In that method, the identified problems of the CWW are evaluated in order to identify their effect over the user. The authors identified that when the user found either a weak or a confusing link, he/she would waste time viewing the interface because he/she would not know what to do. On the other hand, if the user found competitive links, he would waste time viewing the interface in order to find the best solution, rather than trying to understand what it is about.

In paper S17, Kirmani proposes the Heuristic Evaluation Quality Score. This method is based on the heuristic evaluation and evolves it in terms of defect categorization. The paper addresses that the main difficulty of inspectors is in the categorization of usability problems. Inspectors can find the problems, but they do not know if they are catastrophic, serious, minor or cosmetic. Therefore, he proposes to add two perspectives: user and environment, in the description of the categorization. Then, the focus of the inspection is to identify catastrophic, serious and minor usability problems. According to Kirmani, a usability inspector is to be considered experienced if he/she is able to find more than 15% of the known usability defects. Furthermore, a inspector is to be considered above the average, if he/she is able to find from 8% to 15% of the known defects in one hour.

Paper S02 presents an HTML analyzer for the study of web usability. This analyzer was proposed by Alonso-Rios et al. based on the usability problems reported by Nielsen [20]. According to the authors, despite being an HTML analyzer, it includes usability aspects that are not considered by other HTML analyzers. For instance, the proposal verifies if the links are properly highlighted so that users can perceive they are links. Furthermore, the proposal makes use of a tool in order to point usability problems and suggest solutions.

The Web Design Perspective (WDP) based usability evaluation, described in S12, is a technique proposed by Conte et al. that combines two approaches: the use of heuristics and perspectives. Based on the Heuristic Evaluation [19], the WDP technique groups the heuristics under three main perspectives: navigation, conceptual and presentation. The usability inspection process consists of the selection of representative tasks of the Web application to be evaluated. Afterwards, inspectors execute the related tasks while using the heuristics for each perspective to find possible usability problems and generate a list. Later, the inspection team will combine all lists to generate a major list containing all possible usability problems found. Finally, the inspectors will have a meeting to decide whether a possible problem within the list is a real problem or not.

The Recoverability Walkthrough (RW) was proposed by Filgueiras et al. in paper S14. Like the CWW (S06), the RW is based on the Cognitive Walkthrough – CW [23]. However, the RW considers that a usability error is the result of exploration learning. In order to include this idea, Filgueiras et al. (S14) added two questions to the original set proposed by the CW: Q3) *If the user does not do the right thing, can he or she tell that they are in the wrong path?* and Q4) *If the user realizes he or she is in the wrong path, will he or she be able to recover the previous state?* Before carrying out the inspection end users must be modeled and inspectors are selected based on this modeling. During the inspection process each inspector must defend an end user making sure that the interaction is the best for the user he/she is defending. The most peculiar fact in that UIM is that inspectors do not suggest improvements. They are taped so designers can view what usability problems have arisen and make modifications to prevent them.

In paper S18, Molina and Troval suggest a method to integrate usability requirements to Web application requirements. The UIM is called Model Driven Engineering of Web Information Systems and uses the “weaving” process to combine the navigational model and usability features with the requirement model. That method allows inspectors to verify four aspects within the navigational model: importance levels definition, distance between nodes, connectivity restrictions and navigation restrictions. When a new meta-model is obtained through the “weaving” process, the inspectors use a total of 50 metrics to evaluate if the model meets usability features.

Oztekin et al. (S20) proposed the Usability for Web Information Systems (UWIS) methodology. In this UIM, usability can be measured considering three dimensions: efficiency, effectiveness and satisfaction. However, in order to achieve better results they must be used along with quality checklists. The UWIS methodology suggests that it is possible to identify the relationship between quality and usability factors. UWIS makes use of a 27 question list which answers can be statistically analyzed. The analysis aids in identifying which usability problems have the most and least harmful effect over the evaluated Web application. In order to carry out an inspection, one must answer all questions and use the Structural Equation Modeling [13] to identify the relationship between quality and usability factors.

## 6 Discussion

This section summarizes the principal findings of this systematic mapping extension. We also present the implications for both research and practice.

### 6.1 Principal Findings

The goal of this systematic mapping extension was to examine the current use of new usability inspection methods that have emerged to evaluate Web artifacts. The principal findings of this study are the following:

- UIMs for the Web are emerging by: (a) evolving previous generic UIMs and adapting them to the Web domain; and (b) creating new theories for the usability inspection of Web applications. However, as these UIMs evaluate different usability dimensions depending on the basis they use, none of them can address all circumstances and types of Web artefacts. Consequently, the efficiency and effectiveness of the method depends on what is being evaluated: (a) code, (b) prototyped or finished interfaces, or (c) models. A combination of methods could be used to enhance the evaluation results.
- There is a higher number of UIMs for generic Web applications compared to the number of UIMs for specific Web applications. Generic UIMs focus on finding usability problems that can be applied to every Web application but most of them do not provide feedback on how to treat a violation of usability. On the other hand, UIMs that evaluate specific types of Web applications provide evaluators with more data regarding that type of application. This data enables inspectors and practitioners to focus on usability problems that can affect the usability of that specific Web application but cannot be applied to all Web applications.
- The automation of the inspection process is not yet possible in techniques involving judging and human interaction. Consequently, techniques using model and prototype analysis are not being automated but enhanced by using tools to provide inspectors with means of reducing evaluation effort. However, UIMs that use code analysis to find inconsistencies in color or patterns are now being automated. Nevertheless, the evaluated usability aspects of these UIMs are less than the UIMs that make use of inspectors.

### 6.2 Implications for Research and Practice

The results of our systematic mapping extension have implications for both researchers and practitioners. In this subsection we will discuss suggestions for: (a) researchers planning new studies of usability inspections of Web

applications; and (b) practitioners working in the Web development industry and interested in integrating UIMs into their Web development process in an effective manner.

For researchers, we discovered that most UIMs focused on different aspects not always present on all techniques. New UIMs should take into account all existing usability aspects in order to include them, not duplicate them, and not confuse them; therefore allowing the UIM to provide more consistent and complete evaluations.

Our results show that there is a relationship between the type of evaluated artifact and their degree of automation (see Table 3). We noticed that most UIMs that make use of model and prototype analysis are not being automated. However, UIMs making use of code analysis are partially or fully automated. We therefore consider there is a shortage of UIMs able to automatically address usability when evaluating models or prototypes of Web applications. The main problem seems to be that most UIMs are not being able to evaluate usability application dependent factors without the intervention of an inspector.

Our findings regarding the evaluation of generic and specific attributes of Web applications allowed us to determine that there is a small number of UIMs for specific Web applications. The categorization made by this type of UIMs is focused on the context of use, allowing the evaluation of aspects not considered by generic UIMs. For instance, some papers (S05 and S24) categorize Web applications within the subset they are evaluating and allow inspectors to filter usability factors in order to better address usability problems. We consider that there is a need for more UIMs for specific categories of Web applications in order to increase accuracy regarding each type of Web applications.

According to Fernandez et al. [8] most UIMs for the Web are being employed in the requirements, design and implementation phase. We found out the actual UIMs mostly focus on the design and implementation phases. This can be observed in Table 3 as most types of UIMs basically make use of model, code or prototype analysis. Since correcting usability problems during the last stages of the development process can be a costly activity, new research should be oriented towards evaluating early Web artifacts such as the requirement specifications. Practitioners must consider that none of the presented UIMs is able to address all usability problems. However, they can be combined to improve evaluation results and find more usability problems. Therefore, practitioners can apply these techniques in different development phases to increase the effectiveness of the usability evaluation.

In order to improve efficiency many tools have been developed to enhance the speed of usability inspections. Moreover, practitioners can use HTML analyzers to automatically detect color or patterns that produce usability problems. Nevertheless, these analyzers cannot substitute inspectors to find more subjective and intuitive usability issues. Practitioners must bear in mind that each Web application has its specific attributes and therefore UIMs must be adapted to fit users' expectations within a determined context of use. Consequently, in order to execute more consistent and complete evaluations, practitioners must consider all usability factors within the UIMs they apply. This information can be very useful in detecting usability problems and enhancing the evaluation results.

## **7 The Web Design Usability Evaluation technique**

We used the findings from this Systematic Mapping Extension to suggest a set of features that an emerging UIM for the Web should possess in order to meet the actual needs of the software development industry. Readers must note that these features have been thoroughly described in [26]. In this section, we propose the Web Design Usability Evaluation (Web DUE), a usability inspection technique that aims to meet the actual needs of the software development industry regarding Web usability inspection; and a tool to support its inspection process and enhance its results.

### **7.1 The Web DUE proposal**

Our results in [26] suggested that emerging UIMs for Web applications should provide three main features in order to meet the actual needs of the software development industry:

- Ability to find problems in early stages of the development process.
- Ability to find specific problems of the Web Domain.
- Provide a tool to enhance the performance of the inspection process.

In order to meet these features, we proposed the Web Design Usability Evaluation (Web DUE) technique. The Web DUE is a checklist-based technique which aims to increase the number of identified usability problems during the first stages of the development process. To do so, the Web DUE evaluates the usability of Mockups.

One of the main features of the Web DUE technique is that it guides the usability inspection process through typical Web page zones. According to Fons et al. [9], a Web page zone is a piece that is used to compose a Web page. Table 6 shows a list of the Web page zones we used to craft the Web DUE technique. This list was based in the list provided in [9]. However, we noticed the authors did not consider the creation of a Help Zone. A Help Zone could be a zone in which it is possible to obtain access to information about the use of the application and how it works. Moreover, according to Nielsen [19], providing help to users is an important usability attribute. Consequently, we decided to include this zone in order to verify such attribute. In Table 6 we also show the suitability of including a zone in a Web page: (a) mandatory, which means that the Web page must always provide such zone; (b) optional, which means that the Web page could provide such zone, but its absence will not affect the provided functionality; and (c) depends on the functionality, which means that considering the purpose of the Web page, the designers should or could need to include such zones. In other words, we provide information about the necessity of adding a determined Web page zone to a Web Page to provide Usability.

**Table 6:** List of Web page zones based on [9], their contents and the suitability of adding them to a Web page

Zone	Suitability	Contents
Navigation	Mandatory	Navigation Links.
System's State	Mandatory	Information about the application's state and how we got there.
Information	Depends on the Functionality	Information from the application's data base.
Services	Depends on the Functionality	Access to functionalities related to the information zone.
User Information	Depends on the Functionality	Information about the logged user.
Direct Access	Depends on the Functionality	Links to common Web functionalities (Login, Logout, Home).
Data Entry	Depends on the Functionality	Provides a form to input data to execute certain operations.
Institution	Optional	Information about the institution that is responsible for the Web application.
Custom	Optional	Domain-independent content.
Help	Mandatory	Information about how to use the application and how it works.

The main advantage of using Web page zones to direct the inspection process is that inspectors will only need to verify the zones related to the pages within the application he/she is evaluating. In other words, the inspectors do not need to waste time evaluating usability attributes which are not related to the evaluated application. Furthermore, in order to aid inspectors, the Web DUE technique provides a set of usability verification items for each Web page zone. These items will be useful in identifying usability problems.

The verification items were crafted based on the Web Design Perspective-Based Usability Evaluation (S12 – See also Section 5) technique which provides hints that aid inspectors in the usability evaluation of Web applications. We related these hints to the Web page zones and crafted a set of usability verification per each of the web page zones. This usability verification items must be checked in order to identify usability problems. Table 7 shows part of the list of verification items for the Data Entry Zone. In this Table, each verification item is also accompanied by an example/explanation to assist inspectors in case a doubt arises. The complete list of the usability verification items per web page zone can be found in Appendix B.

**Table 7:** Description of the data entry zone and part of the list of its usability verification items

Data Entry Zone	
<b>Description:</b> This zone is responsible for providing the user with a form to input data in order to execute certain operations. Then, a submit-like button links the input data with the associated functionality.	
Usability Verification Items	Examples / Explanations
It is easy to see (find, locate)	The user can easily see where to input data in the system.

**Data Entry Zone**

**Description:** This zone is responsible for providing the user with a form to input data in order to execute certain operations. Then, a submit-like button links the input data with the associated functionality.

**Usability Verification Items**      **Examples / Explanations**

the zone to input data in the system.

It is easy to understand that this zone is for data input.

The input fields (Combo Box, Line Edit, Dial, Horizontal Slider, others) must be easily recognized. The user must know that they are used to input data.

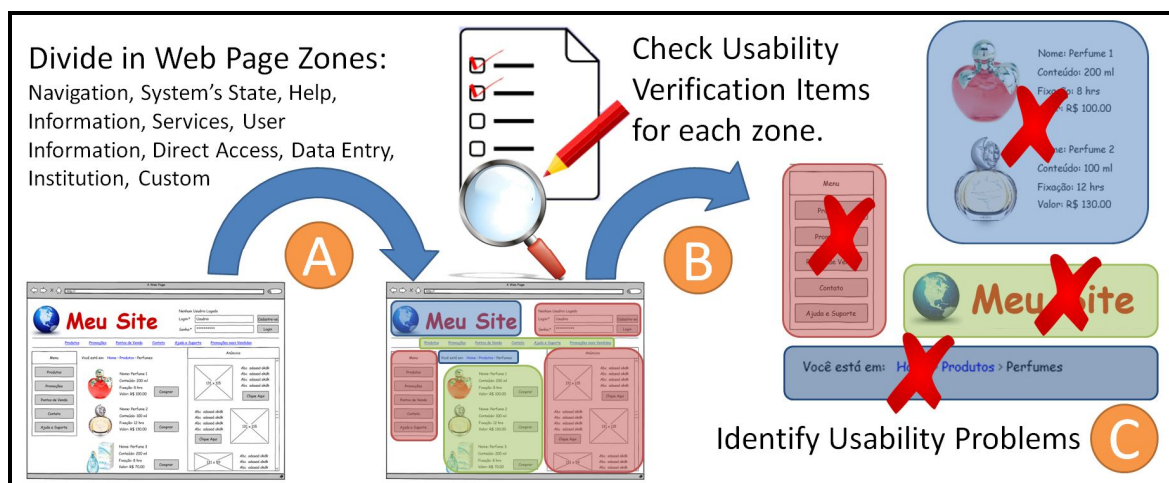
The interface indicates which data must be mandatory filled.

For example, the system indicates mandatory input data with a “\*” or a “mandatory” next to the field.

The interface indicates the correct format for a determined data entrance.

The data fields must contain/provide hints on how to fill them. For example, a “Date” entry field could have the next hint: “dd/mm/yy”.

During the inspection process the inspectors carry out the activities in Figure 2. First, the inspectors must divide the Web mockups into Web page zones (See Figure 2 Step A). Then, while interacting with the mockups, inspectors must examine the identified Web page zones using the technique’s usability verifications items (See Figure 2 Step B). Finally, if a verification item is violated, the inspectors must point the usability problem in the mockup by drawing an “X” or a circle around the problem (See Figure 2 Step C). In the next section we will provide information regarding the Mockup DUE tool, a usability inspection assistant to support the execution process of the Web DUE technique.



**Figure 2:** Stages for performing the inspection process of the Web DUE technique

## 7.2 The Mockup DUE tool

As previously discussed, in [26] we identified that emerging UIMs for Web applications should provide a tool to enhance the performance of the inspection process. Therefore we proposed the Mockup Design Usability Evaluation (Mockup DUE) tool to support three main activities of the Web DUE technique:

- Mapping the interaction between the Web mockups.
- Visualizing and Interacting with the Web mockups.
- Supporting the usability problems detection phase of the Web DUE technique.

In order to support these activities, the Mockup DUE tool works in two stages: planning and detection. During the planning stage, the moderators, which prepare the mockups for inspection, can load their mockups in the tool and connect them by adding links. Furthermore, they can visualize and simulate the interaction steps by clicking in

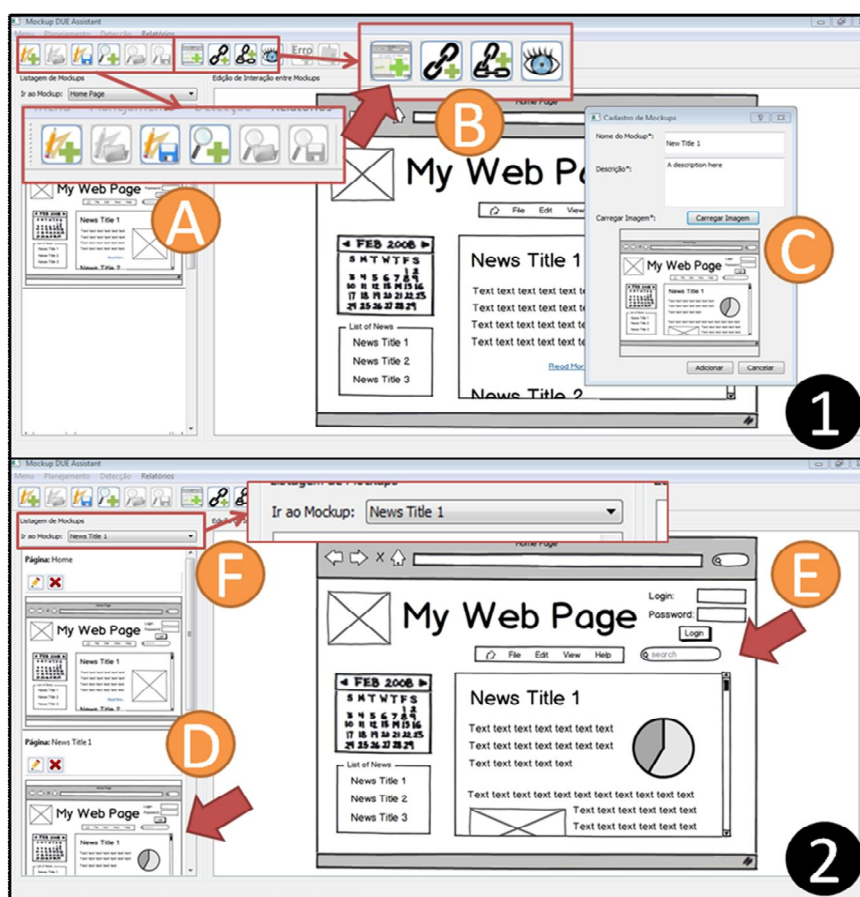


the mockups. This feature allows the creation of clickable mockups, which can provide similar interaction experience as real Web applications.

During the detection phase, inspectors use the mockups, which have been previously mapped by a moderator(s), and perform an inspection using the Web DUE technique. In this stage, the inspector selects a Web page zone and the Mockup DUE tool presents the inspector with a list of usability verification items. The inspector interacts with the mockups while examining them using the usability verification items. If he/she identifies a usability problem, he /she can add that problem and point it out in the mockup. Furthermore, the inspector can add notes with suggestions or considerations at any time.

The following Figures: 3, 4 and 5, show the supported activities from the Mockup DUE tool. We have taken these print screens to show how can the user interact with the tool to perform three main activities: (a) load and map mockups; (b) visualize and interact with the previously mapped mockups; and (c) perform a usability inspection using the Web DUE technique. We will thoroughly describe these Figures, and their parts and elements in the following paragraphs.

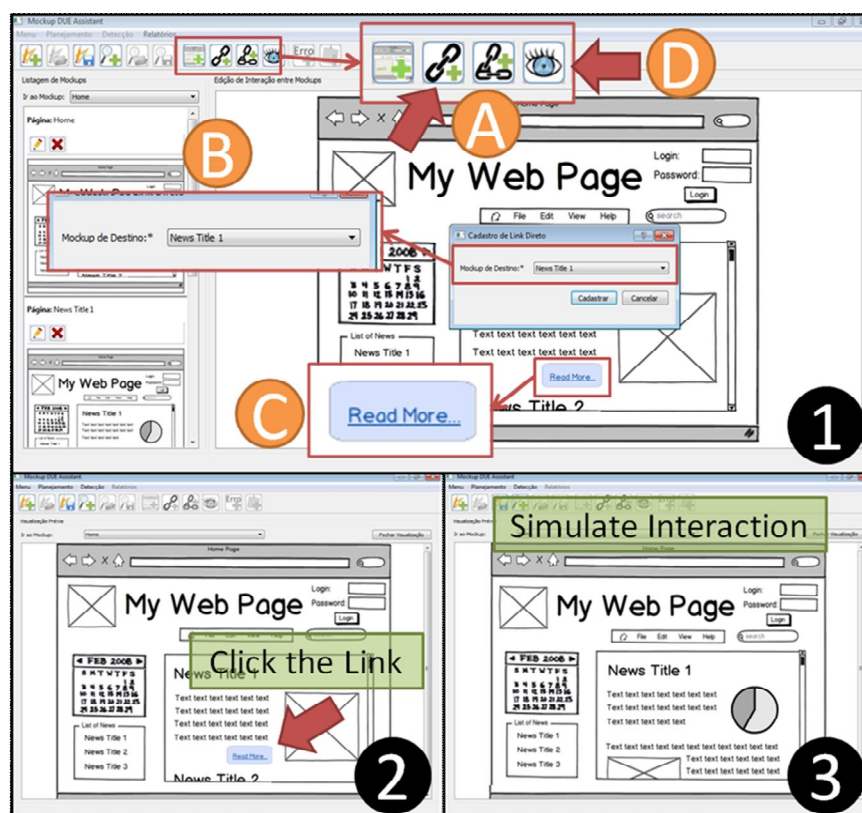
In Figure 3, Part 1 shows the main functionalities of the Mockup DUE tool and the process of uploading Web mockups, while Part 2 shows how the tool presents a mockup once it has been uploaded. In Figure 3, Element A shows that, as mentioned before, there are two main activities to be performed in the tool: planning and detection. In this print screen we are currently performing the functionalities related to the planning stage. Element B from Figure 3 shows the buttons from the planning stage: upload mockups, create links and visualize the mapping of the mockups. In Figure 3 Element C, we have clicked in the “add mockup” button to upload a mockup and entered the required information (name, description and image of the mockup). When finishing uploading a mockup, the tool automatically loads it. As shown in Figure 3 element E, the added mockup is now shown in big scale so it can be better visualized. Furthermore, the user can navigate through the loaded mockups by selecting its name in the combobox in Figure 3 Element F. Furthermore, the user can also view mockups by clicking directly on their miniature pictures as shown in Figure 3 Element D.



**Figure 3:** Loading mockups with the Mockup DUE tool (In Portuguese)

In order to map the interaction between the previously added mockups the user must add links. The process of adding links can be seen in Figure 4 part 1, while the process of interacting with the mockups is illustrated through the transitions from Figure 4 Part B to Figure 4 Part C. The user must click in the “add link” button (see Figure 4 Element A). Then the Mockup DUE tool shows a window and the user will inform the destination of the link (see Figure 4 Element B). In other words, the user must indicate, which mockup will be loaded when the link is clicked. When the user selects the destination and adds the link, a blue semitransparent rectangle will appear in the mockup. This rectangle represents a link in the Mockup DUE tool. The user will be able to resize it and locate it according to his needs. In Figure 4, Element C shows the link that will be activated when the “Read more...” link from the mockup is clicked. Consequently, in order to match the “read more...” link, we have accordingly located and resized the link from the Mockup DUE tool (see the blue semitransparent rectangle in Figure 4 Element C).

The user can also visualize the mapping of the mockups and interact with them by clicking in the “visualize” button (see the button with the eye icon in Figure 4 Element D). Then the Mockup DUE tool enters in what we call, the interact/visualize mode. In this mode, the user clicks in the previously mapped links and can jump from one mockup to another, simulating the interaction among mockups. For instance, when the user clicks the link we previously added (see Figure 4 Part 2), he/she will be taken to the mockup in Figure 4 Part 3.



**Figure 4:** Adding links and visualizing the interaction among mockups with the Mockup DUE tool (In Portuguese)

Figure 5 shows the usability problems detection phase. The detection phase is activated when the user loads a previously mapped set of mockups. In this phase the user will visualize the mockups but, instead of loading mockups or adding links to them, he will use the Web DUE technique to find usability problems (see Figure 5 Part A). After identifying the problems, the inspector will point them in the mockups or add notes (see Figure 5 Part B).

The Web DUE technique is embedded in the Mockup DUE tool, which means that the tool shows both the Web page zones and the usability verification items lists. While carrying out the inspection, the inspector can use the buttons in Figure 5 Element A to add errors or notes. In order to point usability problems the inspector is guided by the Web page zones. As he evaluates each Web page zone (see Figure 5 Element B), the tool shows the usability verification items for the current zone (see one element from the list in Figure 5 Element C). Readers must note

Different inspection methods have been proposed in the past decade to evaluate the usability of Web applications. However, practitioners are not using them [11]. A possible reason for this lack of usage can be the small amount of systematic and analyzed information available about them. Despite researchers' efforts to summarize the current knowledge regarding UEMs, these few existing reviews do not thoroughly describe the actual state of the new proposed UIMs, which are a subset of UEMs that are cheaper and do not require special equipment or laboratory to

be performed. Therefore, there is a need for a more methodological identification of UIMs for the Web in order to meet practitioners' needs.

This paper has extended a systematic mapping study on UEMs for the Web by thoroughly describing UIMs applied on the evaluation of Web artifacts. From the initial set in Fernandez et al. [8], we selected 26 papers addressing new UIMs. We have created new research sub-questions to exhaustively analyze each of the selected papers in order to extract and categorize UIMs. The analysis of the results has allowed us to extract conclusions regarding the state of art in the field of Usability Inspection Methods. For instance, UIMs for the Web are following the steps of generic UIMs like the Heuristic Evaluation and the Cognitive Walkthrough, and are using perspectives to carry out inspections. Furthermore, some of them are focusing on specific types of Web applications like medical and e-commerce. We also identified that the fully automated UIMs are based on HTML analysis. Model and Application/Prototype analysis still require inspectors to carry out the inspection process, and greatly depend on their expertise. However, the expertise provided by inspectors can allow the evaluation of several usability attributes that may not be assessed by automated UIMs.

UIMs for the Web are being mainly applied in the design and implementation phases. If the usability problems are found in later stages of the Web development process, the cost of correcting them can increase. However, new suggestions and approaches are being proposed (S04, S18, S21, S24) to enhance the effectiveness and efficiency in finding usability problems in the early stages of the development process.

This analysis has shown that there are many solutions that can be applied regarding UIMs. Nonetheless, there is still room for improvement. New UIMs for the Web should be able to: find usability problems in earlier phases of the development process, find usability problems that are specific of the Web domain, and provide tool support. To meet these necessities, we proposed the Web DUE technique and the Mockup DUE tool. We are currently evaluating the feasibility of both the technique and tool through empirical studies. Furthermore, we intend to use the extracted data to improve both proposals.

We provided an outline to ensure that usability evaluation methods can be applied in order to evaluate Web artifacts. The results of our analysis can be useful in the promotion and improvement of both the current state-of-the-practice and state-of-the-art of Web usability. We also proposed a technique and a tool, which can help in the evaluation of Web applications in early stages of the development process, improving their quality at a lower cost. The validation of the proposed technique and tool are beyond the scope of this paper. The interested reader can refer to [26] to find further information about the empirical validation of the technique. Further studies are being held to evaluate the feasibility of the tool, and the analysis of their results will be used to improve the inspection process of the Web DUE technique.

### Acknowledgements

We would like to acknowledge the support granted by FAPEAM, and by CNPq process 575696/2008-7. Furthermore, we would like to thank the financial support granted by CAPES to the first author of this paper.

### References

- [1] Alva, M., Martínez, A., Cueva, J.M., Sagastegui, C., Lopez, B.: Comparison of methods and existing tools for the measurement of usability. *Proc. Web. 3rd International Conference on Web Engineering*, Spain, 2003, pp. 386-389.
- [2] Blackmon, M., Polson, P., Kitajima, M., Lewis, C.: Cognitive walkthrough for the Web. *Proc. SIGCHI Conference on Human Factors in Computing Systems*, USA, 2002, pp. 463-470.
- [3] Bolchini, D., Garzotto, F.: Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+. *Proc. International Workshop on Web Usability and Accessibility*, France, 2007, pp. 481-492.
- [4] Bolchini, D., Triacca, L., Speroni, M.: MiLE: a Reuse-oriented Usability Evaluation Method for the Web. *Proc. HCI International Conference*, Greece, 2003, pp. .
- [5] Conte, T., Massollar, J., Mendes, E., Travassos, G.: Web usability inspection technique based on design perspectives. *IET Software*, Volume 3, Issue 2, 2009.
- [6] Cunliffe, D.: Developing usable Web sites – a review and model. *Electronic Networking Applications and Policy*, Volume 10, Issue 4, 2000.
- [7] Cutrell, E., Guan, Z.: What are you looking for? An eye-tracking study of information usage in web search. *Proc. Conference on human factors in computing systems CHI'2007*, USA, 2007, pp. 407-416.

- [8] Fernandez, A., Insfran, E., Abrahao, S.: Usability evaluation methods for the Web: A systematic mapping study. *Information and Software Technology*, Volume 53, Issue 8, 2011.
- [9] Fons, J., Pelechano, V., Pastor, O., Valderas, P., Torres, V.: Applying the OOWS model-driven approach for developing Web applications: The internet movie database case study. In: Rossi, G., Schwabe, D., Olsina, L., Pastor, O.: *Web Engineering: Modeling and Implementing Web Applications*, Springer, 2008.
- [10] Garzotto, F., Paolini, P., Schwabe, D.: HDM— A Model-Based Approach to Hypermedia Application Design *ACM Trans. Information Systems*, Volume 11, Issue 1, 1993.
- [11] Insfran, E., Fernandez, A.: A Systematic Review of Usability Evaluation in Web Development. *Proc. Second International Workshop on Web Usability and Accessibility*, New Zealand, 2008, pp. 81-91.
- [12] International Organization for Standardization, ISO/IEC 9241-11: *Ergonomic Requirements for Office work with Visual Display Terminals (VDTs) – Part 11: Guidance on Usability*, 1998.
- [13] Jöreskog, K.: A general method for analysis of covariance structure. *Biometrika*, Volume 57, Issue 2, 1970.
- [14] Kitajima, M., Blackmon, M., Polson, P.: A Comprehension-based model of Web navigation and its application to Web usability analysis. *Proc. People and Computers XIV*, London, 2000, pp. 357-373.
- [15] Landauer, T., Dumais, S.: A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, Volume 104, 1997.
- [16] Matera, M., Costable, M.F., Garzotto, F., Paolini, P.: SUE Inspection: An Effective Method for Systematic Usability Evaluation of Hypermedia. *IEEE Transactions on Systems, Men, and Cybernetics*, Volume 32, Issue 1, 2002.
- [17] Matera, M., Rizzo, F., Carughi, G. T.: Web Usability: Principles and Evaluation Methods. In: Mendes, E., Mosley, N.: *Web Engineering*, Springer, 2006.
- [18] Mendes, E., Mosley, N., Counsell, S.: The Need for Web Engineering: An Introduction. *Web Engineering*. In: Mendes, E., Mosley, N.: *Web Engineering*, Springer, 2006.
- [19] Nielsen J.: Finding usability problems through heuristic evaluation. *Proc. CHI'92*, UK, 1992, pp. 373-380.
- [20] Nielsen, J.: Heuristic evaluation. Jakob Nielsen, Mack, R. L. (eds), *Usability inspection methods*, Heuristic Evaluation, 1994.
- [21] Offutt, J.: Quality attributes of Web software applications. *IEEE Software: Special Issue on Software Engineering of Internet Software*, Volume 19, Issue 2, 2002.
- [22] Olsina, L., Covella, G., Rossi, G.: Web Quality. In: Mendes, E., Mosley, N.: *Web Engineering*, Springer, 2006.
- [23] Polson, P., Lewis, C., Rieman, J., Wharton, C.: Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, Volume 36, Issue 5, 1992.
- [24] Rivero, L., Conte, T.: Technical Report. A Systematic Mapping Extension on UIMs for Web Applications. Technical Report RT-USES-2012-005, 2012. Available at: [http://www.dcc.ufam.edu.br/uses/index.php/publicacoes/cat\\_view/69-relatorios-tecnicos](http://www.dcc.ufam.edu.br/uses/index.php/publicacoes/cat_view/69-relatorios-tecnicos).
- [25] Rivero, L., Conte, T.: Using an Empirical Study to Evaluate the Feasibility of a New Usability Inspection Technique for Paper Based Prototypes of Web Applications. *Proc. 26th Brazilian Symposium on Software Engineering*, Brazil, 2012, pp. 81-90.
- [26] Rivero, L., Conte, Tayana.: Using the Results from a Systematic Mapping Extension to Define a Usability Inspection Method for Web Applications. *Proc. 24th International Conference on Software Engineering and Knowledge Engineering*, USA, 2012, pp. 582-587.
- [27] Rocha, H., Baranauska, M.: *Design and Evaluation of Human Computer Interfaces (Book)*, Nied, 2003. (in Portuguese)
- [28] Shneiderman B.: *Designing the user interface*. 3rd ed. Reading, MA: Addison-Wesley, 1998.
- [29] Zhang, Z., Basili, V., Shneiderman, B.: Perspective-based Usability Inspection: An Empirical Validation of Efficacy. *Empirical Software Engineering*, Volume 4, Issue 1, 1999.

## Appendix A – Selected Primary Studies List

- [S01] Allen, M., Currie, L., Patel, S., Cimino, J.: Heuristic evaluation of paper-based Web pages: A simplified inspection usability methodology. *Journal of Biomedical Informatics*, Volume 39, Issue 4, 2006.
- [S02] Alonso-Rios, D., Vazquez, I., Rey, E., Bonillo, V., Del Rio, B.: An HTML analyzer for the study of Web usability. *Proc. IEEE International Conference on Systems, Man and Cybernetics, USA, 2009*, pp. 1224-1229.
- [S03] Ardito, C., Lanzilotti, R., Buono, P., Piccinno, A.: A tool to support usability inspection. *Proc. Working Conference on Advanced Visual Interfaces, Italy, 2006*, pp. 278-281.
- [S04] Atterer, R., Schmidt, A.: Adding Usability to Web Engineering Models and Tools. *Proc. 5th International Conference on Web Engineering, Australia, 2005*, pp. 36-41.
- [S05] Basu, A.: Context-driven assessment of commercial Web sites. *Proc. 36th Annual Hawaii International Conference on System Sciences, USA, 2003*, pp. 8-15.
- [S06] Blackmon, M., Polson, P., Kitajima, M., Lewis, C.: Cognitive walkthrough for the Web. *Proc. SIGCHI Conference on Human Factors in Computing Systems, USA, 2002*, pp. 463-470.
- [S07] Blackmon, M., Kitajima, M., Polson, P.: Repairing usability problems identified by the cognitive walkthrough for the Web. *Proc. SIGCHI Conference on Human Factors in Computing Systems, USA, 2003*, pp. 497-504.
- [S08] Blackmon, M., Kitajima, M., Polson, P.: Tool for accurately predicting Website navigation problems, non-problems, problem severity, and effectiveness of repairs. *Proc. SIGCHI Conference on Human Factors in Computing Systems, USA, 2005*, pp. 31-40.
- [S09] Bolchini, D., Garzotto, F.: Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+. *Proc. International Workshop on Web Usability and Accessibility, France, 2007*, pp. 481-492.
- [S10] Burton, C., Johnston, L.: Will World Wide Web user interfaces be usable? *Proc. Computer Human Interaction Conference, Australia, 1998*, pp. 39-44.
- [S11] Chattratichart, J., Brodie, J.: Applying user testing data to UEM performance metrics. *Proc. of the Conference on Human Factors in Computing Systems, Austria, 2004*, pp. 1119-1122.
- [S12] Conte, T., Massollar, J., Mendes, E., Travassos, G.: Web usability inspection technique based on design perspectives. *IET Software*, Volume 3, Issue 2, 2009.
- [S13] Costabile, M., Matera, M.: Guidelines for hypermedia usability inspection. *IEEE Multimedia*, Volume 8, Issue 1, 2001.
- [S14] Filgueiras, L., Martins, S., Tambascia, C., Duarte, R.: Recoverability Walkthrough: An Alternative to Evaluate Digital Inclusion Interfaces. *Proc. Latin American Web Congress, Mexico, 2009*, pp. 71-76.
- [S15] Fraternali, P., Tisi, M.: Identifying Cultural Markers for Web Application Design Targeted to a Multi-Cultural Audience. *Proc. 8th International Conference on Web Engineering, USA, 2008*, pp. 231-239.
- [S16] Habuchi, Y., Kitajima, M., Takeuchi, H.: Comparison of eye movements in searching for easy-to-find and hard-to-find information in a hierarchically organized information structure. *Proc. Symposium on Eye Tracking Research & Applications, USA, 2008*, pp. 131-134.
- [S17] Kirmani, S.: Heuristic Evaluation Quality Score (HEQS): Defining Heuristic Expertise. *Journal of Usability Studies*, Volume 4, Issue 1, 2008.
- [S18] Molina, F., Toval, A.: Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems. *Advances in Engineering Software*, Volume 40, Issue 12, 2009.
- [S19] Moraga, M., Calero, C., Piattini, M.: Ontology driven definition of a usability model for second generation portals. *Proc. 1st International Workshop on Methods, Architectures & Technologies for e-Service Engineering, USA, 2006*.
- [S20] Oztekin, A., Nikov, A., Zaim, S.: UWIS: An assessment methodology for usability of Web-based information systems. *Journal of Systems and Software*, Volume 8, Issue 12, 2009.
- [S21] Paganelli, L., Paterno, F.: Automatic reconstruction of the underlying interaction design of Web applications.

Proc. 14th International Conference on Software Engineering and Knowledge Engineering, Italy, 2002, pp.439-445.

[S22] Paolini, P.: Hypermedia, the Web and Usability issues. Proc. IEEE International Conference on Multimedia Computing and Systems, Italy, 1999, pp. 111-115.

[S23] Signore, O.: A comprehensive model for Web sites quality. Proc. 7th IEEE International Symposium on Web Site Evolution, Hungary, 2005, pp. 30-36.

[S24] Thompson, A., Kemp, E.: Web 2.0: extending the framework for heuristic evaluation. Proc. 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction, New Zeland, 2009, pp. 29-36.

[S25] Triacca, L., Inversini, A., Bolchini, D.: Evaluating Web usability with MiLE+. Proc. 7th IEEE International Symposium on Web Site Evolution, Hungary, 2005, pp. 22-29.

[S26] Vanderdonckt, J., Beirekdar, A., Noirhomme-Fraiture, M.: Automated Evaluation of Web Usability and Accessibility by Guideline Review. Proc. 4th International Conference on Web Engineering, Munich, 2004, pp. 28-30.

**Appendix B – Usability Verification Items from the Web DUE technique**

**Table 7:** Usability Verification Items for the Navigation Zone

Navigation Zone
There is a navigation zone in the system (For example, A navigation Menu).
It is easy to see (find, locate) the System Navigation Zone.
It is easy to understand that that is the navigation zone of the system.
It is easy to understand how to navigate through the system.
It is easy to understand the words/terms/symbols used in the navigation zone.
During the navigation, there are terms and symbols related to the domain of the application. (For example, in order to access the frequent ask questions section, the label uses a correct symbol, like a question mark).
When navigating, the paths that can be followed are presented in a logic and organized way.
It is possible to differentiate between the possible paths when navigating through the application.
The navigation is standardized (there are established colors, layouts, symbols and terms) through the application.
According to the navigation Design:
a) It is pleasant (I like it).
b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
c) It is simple (There is no information that disturbs the navigation).
In case an error occurs (For example, choosing the wrong path):
a) The user will identify that he was mistaken.
b) The user will identify what type of error he made. (For example, wrong click, choosing the wrong menu)
c) The user will be able to undo it.
d) The user will be able to correct it.



**Table 8:** Usability Verification Items for the System's State Zone

System's State Zone
There is a System's State Zone (For example, the system shows a title related to the activity the user is carrying out, or his/her location in the web application).
It is easy to see (find, locate) the System's State Zone.
It is easy to understand that that is the System's State zone of the system.
It is easy to understand what is the actual state of the system (in which part of the web application the user is and what he is doing there).
It is easy to understand the words/terms/symbols used in the System's State zone. (For example, login, search).
In order to show the system's state, the application uses terms and symbols related to the domain of the application (For example, being in the "shopping state", a shopping cart appears).
When system's state is shown in a logic and organized way.
The System's State Zone is standardized (there are established colors, layouts, symbols and terms) through the application.
According to the System's State Design:
a) It is pleasant (I like it).
b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
c) It is simple (There is no information that disturbs the understanding of the System's State).

---

**Table 9:** Usability Verification Items for the Help Zone

Help Zone
<p>There is a Help zone in the system. (For example, a button which leads to the System's Help, a button to access frequent asked questions, or the system possesses help hints that aid users when using it).</p> <p>It is easy to see (find, locate) the Help Zone.</p> <p>It is easy to understand that that is the Help Zone of the system.</p> <p>It is easy to understand how the Help works.</p> <p>It is easy to understand the words/terms/symbols used in the Help Zone.</p>
<p>When accessing the Help within the system:</p> <p>The terms and symbols used to solve questions are related to the domain (For example, the symbol of "frequent asked questions" is presented by a "?").</p> <p>The help is presented in a logic and organized way.</p> <p>The help is presented is standardized (there are established colors, layouts, symbols and terms) through the application.</p> <p>It is possible to differentiate that there are several helps during the application (For example, help for submitting information, or help regarding the problem domain).</p>
<p>According to the Help Design:</p> <p>a) It is pleasant (I like it).</p> <p>b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).</p> <p>c) It is simple (There is no information that disturbs the understanding of the provided help).</p>
<p>In case an error occurs (For example, finding an information that does not answer the user's doubts):</p> <p>a) The user will identify that he was mistaken.</p> <p>b) The user will identify what type of error he made. (For example, the actual help does not answer his/her questions).</p> <p>c) The user will be able to undo it.</p> <p>d) The user will be able to correct it (Search for another help).</p>

**Table 10:** Usability Verification Items for the Information Zone

Information Zone
There is an information zone in the system (For example, the system shows information relevant to the execution of an activity, or show data in text, table or image format).
It is easy to see (find, locate) the Information Zone.
It is easy to understand that that is the Information Zone of the system.
It is easy to understand the information (For example, text, graphics and tables).
The terms and symbols used to solve questions are related to the domain (For example, the symbol for money is "\$").
To show information, the system uses terms and symbols related to the problem domain (For example, in order to send a gift from the application, the user must click in the button with the gift symbol).
The information is presented in a logic and organized way.
The presented information is standardized (there are established colors, layouts, symbols and terms) through the application.
According to the Information Zone Design:
a) It is pleasant (I like it).
b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
c) It is simple (There is nothing that disturbs the understanding of the relevant information).

---

**Table 11:** Usability Verification Items for the Services Zone

Services Zone
There is a services zone (For example, a zone where the user can find functionalities related to the available information within the web application).
It is easy to see (find, locate) the Services Zone.
It is easy to understand that that is the Services Zone of the system.
It is easy to understand how to use the services within the application.
É fácil entender as palavras/termos/símbolos usados na zona de serviços.
When accessing the services, the system uses terms and symbols related to the problem domain (For example, the symbol of the buying button is represented by a shopping cart).
The provided services are presented in a logic and organized way.
It is possible to differentiate between services (For example, the user will know he is searching for a product's information rather than buying it).
The services zone is presented is standardized (there are established colors, layouts, symbols and terms) through the application.
According to the Services Zone Design:
a) It is pleasant (I like it).
b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
c) It is simple (There is no information that disturbs the usage of the services).
In case an error occurs (For example choosing the wrong service):
a) The user will identify that he was mistaken.
b) The user will identify what type of error he made. (For example, the service is not working, that was not the correct service).
c) The user will be able to undo it.
d) The user will be able to correct it.

**Table 12:** Usability Verification Items for the User Information Zone

User Information Zone
<p>There is a ser information Zone (For example, the system shows data of the logged user: login, name).</p> <p>It is easy to see (find, locate) the User Information Zone.</p> <p>It is easy to understand that that is the User Information Zone of the system.</p> <p>It is easy to understand if someone is logged in as a user and who he is.</p> <p>It is easy to understand the words/terms/symbols used to indicate who is logged in the web application (For example, “Welcome, John Doe”; or “Your are logged as: JohnDoe@email.com”).</p> <p>It is easy to understand the words/terms/symbols used to indicate when someone is not logged in the web application (For example, “Welcome, Start your session”; or “Welcome, you have not yet logged in, please click here”).</p> <p>In order to show information about the logged user, the system uses terms and symbols related to the domain of the application (For example, User: JohnDoe@email.com).</p> <p>User information is presented in a logic and organized way.</p> <p>The presented user information is standardized (there are established colors, layouts, symbols and terms) through the application.</p>
<p>According to the User Information Design:</p> <p>a) It is pleasant (I like it).</p> <p>b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).</p> <p>c) It is simple (There is nothing that disturbs the understanding of the information regarding the logged user).</p>

**Table 13:** Usability Verification Items for the Direct Access Zone

Direct Access Zone
There is a Direct Access Zone (For example, a menu with links or shortcuts).
It is easy to see (find, locate) the Direct Access Zone.
It is easy to understand that that is the Direct Access Zone of the system.
It is easy to understand how to use the shortcuts and what they do.
It is easy to understand the words/terms/symbols used to indicate shortcuts within the Direct Access Zone.
The shortcuts' names use terms and symbols related to the domain of the application. (For example, the login button has a door symbol).
The shortcuts are presented in a logic and organized way.
It is possible to differentiate between shortcuts (For example, login shortcut, search shortcut, buy shortcut).
The presented direct access zone is standardized (there are established colors, layouts, symbols and terms) through the application.
According to the Direct Access Zone Design:
a) It is pleasant (I like it).
b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
c) It is simple (There is no information that disturbs the usage of the services).
In case an error occurs (For example, clicking in the wrong link):
a) The user will identify that he was mistaken.
b) The user will identify what type of error he made. (For example, wrong click, wrong shortcut).
c) The user will be able to undo it.
d) The user will be able to correct it.

**Table 14:** Usability Verification Items for the Data Entry Zone

**Data Entry Zone**

There is a Data Entry Zone (For example, a input form).

It is easy to see (find, locate) the Data Entry Zone.

It is easy to understand that that is the Data Entry Zone of the system.

It is easy to understand how to fill data in the system (For example, marking, typing, and tapping).

It is easy to understand the words/terms/symbols used to indicate shortcuts within the Data Entry Zone.

When entering the data, the input fields are presented in a logic and organized way.

It is easy to understand what is the correct format in which that data must be filled in (For example, A date must be input like this: dd/mm/yy).

It's easy to understand which data need to be mandatory filled (For example, mandatory data are marked with a "\*").

It's easy to differentiate between the fields in which the user can type in.

The system allows the user to reuse the data he/she has previously filled in (For example, using bank account data from a bank account the user has used before).

Data entrance is standardized (there are established colors, layouts, symbols and terms) through the application.

**According to the Data Entry Zone Design:**

- a) It is pleasant (I like it).
- b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
- c) It is simple (There is no information that disturbs the usage of the forms).

**In case an error occurs (For example, filling data incorrectly or not filling mandatory fields):**

- a) The user will identify that he was mistaken.
- b) The user will identify what type of error he made. (For example, filling data incorrectly or not filling mandatory fields).
- c) The user will be able to undo it.
- d) The user will be able to correct it.

**Table 15:** Usability Verification Items for the Institution Zone

Institution Zone
There is a zone that shows information about the Institution (For example, its logo, name, information).
It is easy to see (find, locate) the Institution Zone.
It is easy to understand that that is the Institution Zone of the system.
It is easy to understand which is the institution running the web application.
It is easy to understand the words/terms/symbols used to indicate the institution.
 The institution is presented in a standardized way (there are established colors, layouts, symbols and terms) through the application.
According to the Institution Zone Design:
a) It is pleasant (I like it).
b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
c) It is simple (There is no information that distract me from understanding what is the Institution running the web application).



**Table 16:** Usability Verification Items for the Custom Zone

Custom Zone
There is a custom zone in the system (For example, a zone which content's is NOT related to the purpose of the application, like external applications and advertisement).
It is easy to see (find, locate) the Custom Zone.
It is easy to understand that that is the Custom Zone of the system.
It is easy to understand what can be done in the Custom Zone.
It is easy to understand the words/terms/symbols used within the Custom Zone.
The terms and symbols used within the Custom Zone are related to the domain of the application. (For example, an advertisement makes sense).
The components within the Custom Zone are presented in a logic and organized way.
It is possible to differentiate the many components of the custom zone.
It is possible to find shortcuts to enhance interaction.
The custom content is presented in a standardized way (there are established colors, layouts, symbols and terms) through the application.
According to the Custom Zone Design:
a) It is pleasant (I like it).
b) Does not produce physical discomfort. (For example, it does not use strong colors or small, illegible letters).
c) It is simple (There is no information that disturbs the usage of the custom zone).
In case an error occurs (For example, choosing the wrong path):
a) The user will identify that he was mistaken.
b) The user will identify what type of error he made. (For example, wrong click)
c) The user will be able to undo it.
d) The user will be able to correct it