Fernández Martínez, A.; Abrahao Gonzales, SM.; Insfrán Pelozo, CE.; Matera, M. (2013). Usability Inspection in Model-Driven Web Development: Empirical Validation in WebML. Lecture Notes in Computer Science. 8107:740-756. doi:10.1007/978-3-642-41533-3_45.

Usability Inspection in Model-driven Web Development: Empirical Validation in WebML

Adrian Fernandez[1], Silvia Abrahão[1], Emilio Insfran[1] and Maristella Matera[2]

[1] ISSI Research Group, Universitat Politècnica de València, Spain
{afernandez, sabrahao, einsfran}@dsic.upv.es
[2] Politecnico di Milano, Italy
matera@elet.polimi.it

Abstract. There is a lack of empirically validated usability evaluation methods that can be applied to models in model-driven Web development. Evaluation of these models allows an early detection of usability problems perceived by the end-user. This motivated us to propose WUEP, a usability inspection method which can be integrated into different model-driven Web development processes. We previously demonstrated how WUEP can effectively be used when following the Object-Oriented Hypermedia method. In order to provide evidences about WUEP's generalizability, this paper presents the operationalization and empirical validation of WUEP into another well-known method: WebML. The effectiveness, efficiency, perceived ease of use, and satisfaction of WUEP were evaluated in comparison to Heuristic Evaluation (HE) from the viewpoint of novice inspectors. The results show that WUEP was more effective and efficient than HE when detecting usability problems on models. Also, inspectors were satisfied when applying WUEP, and found it easier to use than HE.

Keywords: Model-driven Web development, Usability inspections, Measure operationalization, Empirical validation, WebML

1 Introduction

Usability is considered as one of the most important quality factors for Web applications: the ease or difficulty experienced by users largely determines their success or failure [26]. The challenge of developing more usable Web applications has promoted the emergence of a large number of usability evaluation methods [24]. However, most of these approaches only consider usability evaluations after the Web application is fully implemented and deployed. Studies such as that of Matera et al. [20] and Juristo et al. [18] however claim that usability evaluations should also be performed at early stages of the Web development (e.g., at modeling time) in order to detect early how to improve the user experience and decrease maintenance costs. This is in line with the results of a recently performed systematic mapping study on usability evaluation methods for Web applications [9], which revealed a lack of usability evaluation methods that have been empirically validated and that can be properly used to evaluate analysis and design models of a Web application under development.

In order to address these issues, we have proposed a usability inspection method (Web Usability Evaluation Process – WUEP [10]), which can be instantiated and integrated into different *model-driven Web development processes*. The peculiarity of these development processes is that Platforms-Independent Models (PIMs) and Platform-Specific Models (PSMs) are built to represent the different views of a Web application (e.g., content, navigation, presentation); finally, the source code (Code Model - CM) is obtained from these models using model-to-text automatic transformations. In this context, inspections of the PIMs and PSMs can provide early *usability evaluation reports* to identify potential usability problems that can be corrected prior to the generation of the source code.

In our view, comparative empirical studies are useful to evaluate and improve any newly proposed evaluation method, since valuable information can be achieved when a method is compared to others. Several empirical studies for validating Web usability evaluation methods have been reported in literature (e.g., [8]). However, they focus on traditional Web development processes, while only few studies address model-driven Web development processes (e.g., [1][19][27]). Among these studies, we presented in [12] an operationalization and validation of WUEP in a specific process based on the Object-Oriented Hypermedia (OO-H) method. In this study, WUEP was compared against Heuristic Evaluation (HE) [25]. The results showed that WUEP is more effective and efficient than HE in supporting the detection of usability problems.

In order to verify the generalization of WUEP into another process, we also operationalized WUEP for its application to the Web Modeling Language (WebML) [6], one of the most well-known industrial model-driven Web development process. We adapted generic measures, taken from the Web Usability Model [10] on which WUEP is based, to specific WebML modeling constructs, as a means to *predict* and *improve* the usability of Web applications generated from these models. A pilot experiment was conducted to analyze the feasibility and validity of this operationalization [11]. In this paper, we present the results of an experiment replication aimed at providing further analysis about its effectiveness, efficiency, perceived ease of use, and satisfaction in comparison to HE.

This paper is structured as follows. Section 2 discusses existing work that addresses usability evaluations in model-driven Web development. Section 3 provides an overview of WUEP. Section 4 describes how WUEP has been instantiated for use with WebML. Section 5 describes the experiments designed to empirically validate WUEP. Section 6 shows the analysis of the results obtained and discusses threats to the validity. Finally, Section 7 presents our conclusions and further work.

2 Related Work

Despite the fact that several model-driven development (MDD) methods have been proposed since late 2000 for developing Web-based interactive applications, few work address usability evaluations in this type of processes (e.g., [2],[13],[22]).

Atterer and Schmidt [2] proposed a prototype of a model-based usability validator. The aim was to perform an analysis of operative Web applications by previously

tagging its sections in order to build a page model. This model is then compared by patterns extracted from usability guidelines.

Fraternali et al. [13] presented the Web Quality Analyzer, a framework which is able to automatically analyze the XML specification of Web applications designed through WebML for identifying the occurrence of some design patterns, and calculating metrics revealing if they are used consistently throughout the application.

Molina and Toval [22] presented a method to integrate usability goals in model-driven Web development by extending the expressiveness of navigation models to incorporate these usability goals. A meta-model was defined in order to describe the requirements to be achieved for these navigational models.

These works represent the first steps to incorporate usability evaluation in model-driven Web development, however from them it does not emerge any systematic process. Furthermore, only few of them have been validated through empirical studies to show evidence about the effectiveness of performing usability evaluations on models (e.g., [1] [19] [27]).

There are some studies in literature that compare usability evaluation methods through empirical studies. Abrahão et al. [1] present an empirical study which evaluates the user interfaces generated automatically by a model-driven development tool. This study applies two usability evaluation methods: an inspection method (Action Analysis) and an empirical method (User Testing) with the aim of comparing what types of usability problems the two methods are able to detect in the user interfaces, and what their implications are for transformations rules and PIMs.

Matera et al. [19] presented the empirical validation of the Systematic Usability Evaluation (SUE) method for hypermedia applications based on the adoption of operational guidelines called Abstract Tasks. The experiment showed the major effectiveness and efficiency of the inspection method with respect to traditional heuristic evaluation techniques.

Panach et al. [27] provided metrics to evaluate the understandability attributes of Web applications (i.e., a usability sub-characteristic) as result of a model-driven development process. Metrics values were aggregated to obtain indexes which were compared to the perception of these same attributes by end users. However, the study did not consider any performance measure of method usage. As indicated by Hornbæk [14], for assessing the quality of usability evaluation methods it is important to consider not only the evaluators' observations and satisfaction with the methods under evaluation but also the performance of the methods (e.g., in terms of number of usability detected problems).

The analysis of the previous works highlights a lack of empirical validations of usability inspection methods for model-driven Web development processes. This motivated us to conduct a family of experiments to validate our usability inspection method when it was applied to the Object-Oriented Hypermedia (OO-H) method [12]. However, generalizations about the usefulness of WUEP require it to be instantiated and validated in other model-driven Web development methods. Hence, this paper focuses on the operationalization of WUEP to another method, WebML, and on its validation through an experimental study.

3 Web Usability Evaluation Process

The Web Usability Evaluation Process (WUEP) has been defined by extending and refining the quality evaluation process that is proposed in the ISO 25000 standard [16]. The aim of WUEP is to integrate usability evaluation into model-driven Web development processes by employing a Web Usability Model as the principal input artifact. This model breaks down the usability concept into 16 sub-characteristics and 66 measurable attributes, which are then associated with 106 measures in order to quantify them. These measures provide a generic definition, which should be operationalized in order to be applied to models obtained at different abstraction levels (PIMs, PSMs, and, CMs) in different MDWD processes (e.g., WebML, OO-H).

The aim of applying measures is to reduce the subjectivity inherent to existing inspection methods. It is important to remark that by applying measures, the evaluators inspect models in order to predict usability problems (i.e., to detect problems that would be experienced by end-users when using the generated Web application). We are not intended to evaluate the usability of the models themselves. Therefore, inspection of these models (by considering the traceability among them) allows us to discover the source of the detected usability problems and facilitates the provision of recommendations to correct these problems at earlier stages of the Web development process.

We are aware that not all usability problems can be detected based on the evaluation of models since they are limited by their own expressiveness and, most importantly, they may not predict the user behavior or preferences. However, studies such as that of Hwang and Salvendy [15] claim that usability inspections, applying well-known usability principles on software artifacts, may find around 80% of usability problems. In addition, the use of inspection methods for detecting usability problems in models can be complemented with other evaluation methods performed with end-users before releasing a Web application to the public.

The main stages of WUEP are:

1. In the *establishment of the evaluation requirements* stage, the evaluation designer defines the scope of the evaluation by (a) establishing the purpose of the evaluation; (b) specifying the evaluation profiles (type of Web application, Web development method employed, context of use); (c) selecting the Web artifacts (models) to be inspected; and (d) selecting the usability attributes from the Web usability model which are going to be evaluated.
2. In the *specification of the evaluation* stage, the evaluation designer operationalizes the measures associated with the selected attributes in order for them to be applied to the models to be evaluated. This operationalization consists of establishing a mapping between the generic definition of the measure and the concepts that are represented in the Web artifacts (modeling primitives of models or UI elements in the final Web application). In addition, thresholds are established for ranges of values obtained for each measure by considering their scale type and the guidelines related to each measure whenever possible. These thresholds provide a usability problem classification based on their severity: low, medium, or critical. It is important to note that the operationalization needs to be performed once within a

specific model-driven Web development method, and can be reused in further evaluations that involve Web applications developed using the same method.

3. In the *design of the evaluation* stage, the template for usability reports is defined and the evaluation plan is elaborated (e.g., number of evaluators, evaluation constraints).
4. In the *execution of the evaluation* stage, the evaluator applies the operationalized measures to the selected Web artifacts (i.e., models) in order to detect usability problems by considering the rating levels established for each measure.
5. In the *analysis of changes* stage, the Web developer analyzes all the usability problems in order to propose changes with which to correct the affected artifacts from a specific stage of the Web development process. The changes are applicable to the previous intermediate artifacts (i.e., PIMs, PSMs and model transformations if the evaluation is performed on the final Web user interface).

4 Instantiation in WebML

This section presents how WUEP can be instantiated for evaluating the usability of Web applications developed using the Web Modeling Language (WebML) method. This method is complemented by the WebRatio tool, which offers visual editors for the definition of the models and transformation techniques for code generation in different platforms. WebML was selected because: i) it is a well-known model-driven Web development method in industry with several success stories reported [28], ii) it offers conceptual models of real Web applications and their corresponding generated source code, and iii) it can be considered a representative method of the whole set of model-driven Web development methods [23] .

In the rest of this section, we first give a short overview about WebML to present its main modeling primitives. Secondly, we provide some examples of how some generic measures were operationalized in WebML models. Finally, we also provide a proof of concept about how WUEP can be applied in a WebML-based Web application in order to detect and report usability problems at early stages of the Web development process.

4.1 Overview of WebML

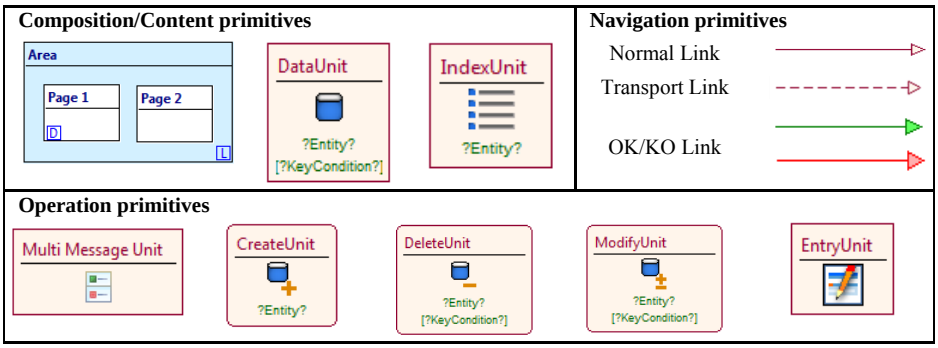
WebML is a domain-specific language for specifying the content structure of Web applications (i.e., Data Model) and the organization and presentation of their contents in one or more hypertexts (i.e., Hypertext Model). Considering that the Hypertext Model is obtained early in the Web development process, it plays a relevant role in ensuring the usability of the final Web application since it describes how data resources are assembled, interconnected and presented into information units and pages. Table 1 shows some of the most representative modeling primitives provided by the Hypertext Model. These primitives are classified according to three perspectives: a) Composition, defining pages and their internal organization in terms of elementary interconnected units; b) Navigation, describing links between pages and content units to be provided to facilitate information location and browsing; and c)

Operation, specifying the invocation of external operations for managing and updating content.

Composition primitives are based on containers called *Pages* (which can be grouped by *Areas*) and a set of building blocks called Content units. *Pages* and *Areas* can be marked as *Homepage (H)*, *Landmark (L)*, or *Default (D)*. The content units represent one or more instances of the entities of the structural schema, typically selected by means of queries over the entity attributes or over their relationships. In particular, they allow representing a set of attributes for an entity instance (*DataUnits*), and list of properties of a given set of entity instances (*IndexUnits*).

Navigation primitives are based on links that connect units and pages, thus forming the hypertext. Links connect units in several configurations, yielding to composite navigation mechanisms. They can be activated by a user action (*Normal Link*); the Web application (*OK or KO Link*); or even can be employed only as transport of parameters between modeling primitives (*Transport Link*).

Table 1. WebML Hypertext modeling primitives



Operation primitives enable managing the messages that are prompted to the user after any operation (*MultiMessageUnit*), expressing built-in update operations, such as creating, deleting or modifying an instance of an entity (respectively represented through the *CreateUnit*, *DeleteUnit* and *ModifyUnit*), and collecting input values into fields (*EntryUnits*). From the user point of view the execution of an operation is a side effect of navigating a contextual link. Operations may have different incoming links, but only one is the activating-one.

The Data Model and Hypertext Model are taken as input of a model compiler that is able to automatically generate the Web application source code. This is supported by the WebRatio tool which also provides predefined presentation templates to customize the presentation of the final Web application.

4.2 Operationalizing measures for WebML

The operationalization of measures is a mean to establish a mapping between the generic definition of the measure and the modeling primitives that are represented in a specific model defined during a specific MDWD process.

For WebML, we have operationalized a total of 16 measures for the Hypertext model (`http://www.dsic.upv.es/~afernandez/MODELS13/operationalization`).

As an example, Table 2 presents two measures (i.e., PAE and UOC) from the Web Usability Model and shows their operationalization for the WebML Hypertext Model. The details regarding the generic definition of the measure are provided by the five first rows: *name, attached usability attribute*, *generic description*, *measurement scale*, and *interpretation*. The details regarding the operationalization of the measure are provided in the last two rows: *operationalization* and *thresholds* established in order to detect a usability problem (UP). In these examples, thresholds were established by dividing the range of obtained values in convenient intervals. However, other examples of measures provide empirically validated thresholds (e.g., navigation depth). Domain experts (Web designers) have validated these values. The mapping between each element from the generic measure definition and the modeling primitives is highlighted in bold and marked with asterisks (*).

Table 2. Examples of operationalized measures to be applied in WebML

Measure Name	**Proportion of actions with error messages associated (PAE)**
Usability Attribute	Appropriateness recognizability / User guidance / Message availability
Generic Description	Ratio between the number of **user actions (*)** without an **error message (**)** to provide feedback and the total number of user actions.
Scale	Ratio between 0 and 1
Interpretation	The higher the value worse is the guidance (in terms of messages) that is provided to the user..
Operationalization	Let HM : Hypertext Model $PAE(HM) = \frac{\text{Number of } \textbf{Operation Units (*)} \text{ that not provide a KO link leading to a } \textbf{MultiMessage Unit (**)}}{\text{Total number of } \textbf{Operation Units (*)}}$ (1) Where Operation Units can be any CreateUnit, ModifyUnit and DeleteUnit
Thresholds	[PAE = 0]: No UP [0.3 < PAE ≤ 0.6]: Medium UP [0 < PAE ≤ 0.3]: Low UP [0.6 < PAE ≤ 1]: Critical UP

Measure Name	**User operation cancellability (UOC)**
Usability Attribute	Operability / Controllability / Cancel support
Generic Description	Proportion between the number of **implemented functions (*)** that cannot be **cancelled by the user (**)** prior to completion and the total number of functions requiring the pre-cancellation capability.
Scale	Ratio between 0 and 1.
Interpretation	The higher the value the worse controllability the WebApp presents due to the fact that it is necessary to use external operations (i.e., browser actions) in order to go back to a previous state if user wants to cancel the current operation.
Operationalization	Let HM : Hypertext Model $OUC(HM) = \frac{\text{Number of } \textbf{Operation Units (*)} \text{ reached by a unit which has not a } \textbf{Normal Link (**)} \text{ to its predecessor unit}}{\text{Total number of } \textbf{Operation Units (*)}}$ (2) Where *Operation Units* can be any *CreateUnit*, *ModifyUnit* and *DeleteUnit*
Thresholds	[UOC = 0]: No UP [0.3 < UOC ≤ 0.6]: Medium UP [0 < UOC ≤ 0.3]: Low UP [0.6 < UOC ≤ 1]: Critical UP

4.3 Applying WUEP into Practice with WebML

We here show a proof of concept about the feasibility of WUEP by applying it to evaluate the usability of a WebML-based Web application. We follow the steps introduced in Section 3.

Establishment of evaluation requirements. The purpose of the evaluation is to perform an early usability evaluation during the development of an e-commerce Web application. The application selected is a furniture online store aimed at supporting two types of users: potential customers, and the website administrator. The Web artifact to be evaluated is the Hypertext Model HM0 (see Figure 1), which covers the Store editing functionality issued by the administrator. The Area *Store editing* allows the administrator to access all the stores (IndexUnit *All Stores*) and their details (Normal Link *expand* and DataUnit *Store details*), adding new stores (Normal Link *new*, EntryUnit *New Store*, and CreateUnit Create store); removing existing stores (Normal Link *drop* and DeleteUnit *Delete store*), and modifying existing stores (EntryUnit *Modify Store*, Normal Link *apply*, and CreateUnit *Create store*). All the operations include their OK and KO links after its completion.

The usability attributes to be evaluated are *Message availability* and *Cancel support*. These attributes were selected because of their relevance for any data-intensive Web applications [7].

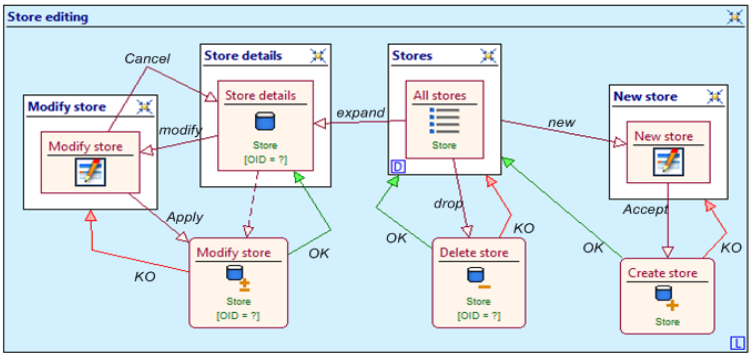


Fig. 1. Hypertext Model HM0.

Specification of the evaluation. The generic measures selected were the ones presented in Table 2.

Design of the evaluation. A template for reporting usability problems (UP) is defined by considering the following fields: ID, description of the UP, affected usability attribute, severity level, artifact evaluated, source of the problem, occurrences, and recommendations.

Execution of the evaluation. The operationalized measures are applied in the Web artifacts in order to detect usability problems:

Proportion of actions with error messages associated (PAE). Applying this measure (Table 2, Equation 1), we obtain the value 3/3 = 1 since from a total of three Operation Units (*Create Store*, *Modify Store*, and *Delete Store*) none of them has its KO link connected to a MultiMessageUnit. This means that a critical usability problem was detected (and reported as UP01 in Table 3(a)) since the value obtained is in the threshold [$0.6 < PAE \leq 1$].

User operation cancellability (UOC). Applying this measure (Table 2, Equation 2), we obtain the value 2/3 = 0.66 since from the total of three Operation Units (*Create Store*, *Modify Store*, and *Delete Store*) only two OperationUnits (*Create Store*, and

Delete Store) are not reached by a unit with a return link to its predecessor. This means that a critical usability problem was detected (and reported as UP02 in Table 3(b)) since the value obtained is in the threshold [0.6 < UOC ≤ 1].

Table 3. Usability report

a) ID	**UP01**
Description	There are no messages that help Web designers to identify which types of errors have occurred during performing operations
Affected attribute	Appropriateness recognisability / User guidance / Message availability
Severity level	Critical: [0.6 < PAE=1 ≤ 1]:
Artifact evaluated	Hypertext Model HM0
Problem source	Hypertext Model HM0
Occurrences	3 Operation Units: *Create Store*, *Modify Store*, and *Delete Store*.
Recommendations	Connect a MultiMessage Unit to the KO link for each Operation Unit.

b) ID	**UP02**
Description	There some operations that cannot be cancelled by the user
Affected attribute	Operability / Controllability / Cancel support
Severity level	Critical: [0.6 < UOC=0.66 ≤ 1].
Artifact evaluated	Hypertext Model HM0
Problem source	Hypertext Model HM0
Occurrences	2 Operation Units: *Create Store*, and *Delete Store.*
Recommendations	In relation to the OperationUnit *Create Store*: add a new Normal Link *cancel* from the EntryUnit *New Store* to the Page *All Stores*. With regard the OperationUnit *Delete Store*: add a EntryUnit *confirmation* between the IndexUnit *All stores* and the OperationUnit itself. The new EntryUnit *confirmation* would have a new Normal Link *cancel* from itself to the Page *All Stores*.

Analysis of changes. The changes proposed by this report are analyzed by the Web developers (e.g., cost, impact, difficulty) and lately corrected. Figure 2 shows the Hypertext Model which was manually corrected by the Web developer considering the usability report. However, we aim at automatizing the application of changes. By considering the traceability between the Hypertext Model and the final Web application, the corrections proposed are aimed at obtaining a more usable Web application by construction [1], where each model of a Web application is inspected and improved before the source code is generated.

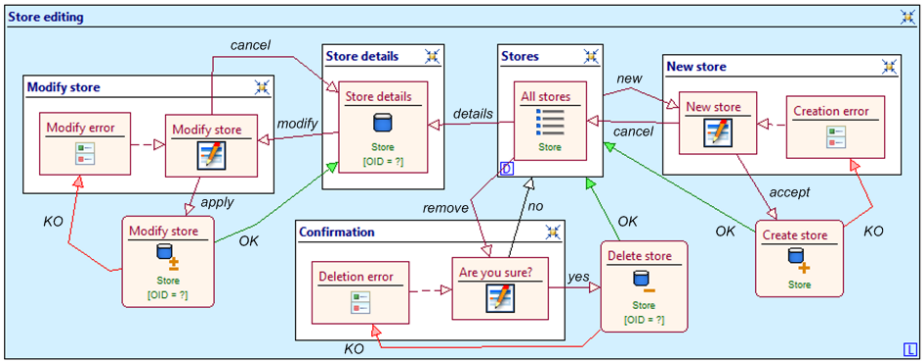


Fig. 2. Hypertext Models corrected after the usability evaluation.

5 Empirical Validation

This section first presents an overview of the original experiment, then the design and execution of the experiment replication. The results obtained in both experiments are also presented and discussed. We followed the guidelines proposed by Wohlin et al. [29] and Juristo and Moreno [17].

5.1 Overview of the Original Experiment (EXP)

According to the Goal-Question-Metric (GQM) paradigm [3], the goal of the experiment was to analyze the WUEP operationalization for the WebML development process, for the purpose of evaluating it with regard to its effectiveness, efficiency, perceived ease of use, and the evaluators' perceived satisfaction of it in comparison to Heuristic Evaluation (HE) from the viewpoint of a group of novice usability evaluators. The context of the experiment is the evaluation of two Web applications performed by novice inspectors. This context is determined by the Web applications to be evaluated, the usability evaluation methods to be applied and the subject selection.

The Web applications selected were a Web Calendar for meeting appointment management, and an e-commerce application for a Book Store. They were developed by the WebRatio company using the WebML model-driven development process. Two different functionalities of the Web Calendar application (appointment management and user comments support) were selected for defining the experimental object O1, whereas two different functionalities of the Book Store application (search and shopping) were selected for defining the experimental object O2. Each experimental object contains two Web artifacts: a Hypertext model (HM) and a Final User Interface (FUI) generated from the model. We selected these four functionalities since they are relevant to the end-users and similar in size and complexity.

The usability inspection methods to be evaluated were WUEP and HE. Since the context of the experiments was from the viewpoint of a group of usability inspectors, we evaluated the execution stages of both methods. Two of the authors therefore performed the evaluation designer role in both methods in order to design an evaluation plan. In critical activities such as the selection of usability attributes in WUEP, we required the help of two external Web usability experts. In the case of the HE, all 10 heuristics were selected. In the case of the WUEP, a set of 20 usability attributes were selected as candidates from the Web Usability Model through the consensus reached by the two evaluator designers and other two Web usability experts. The attributes were selected by considering the evaluation profiles (i.e., which of them would be more relevant to the type of Web application and the context in which it is going to be used). Only 12 out of 20 attributes were randomly selected in order to maintain a balance in the number of measures and heuristics to be applied. The associated measures from the 12 attributes were operationalized to be applied at the selected Web artifacts (6 measures for HMs and 6 measures for FUIs).

The experiment was conducted in the context of an Advanced Software Engineering course from September 2011 to January 2012 at the Universitat

Politècnica de València (UPV). Specifically, the subjects were 30 fifth-year students enrolled in the undergraduate program in Computer Science.

The experiment has two independent variables: the evaluation method (WUEP and HE) and the experimental objects (O1 and O2). There are two objective dependent variables: effectiveness, which is calculated as the ratio between the number of usability problems detected and the total number of existing (known) usability problems; and efficiency, which is calculated as the ratio between the number of usability problems detected and the total time spent on the inspection process. There are also two subjective dependent variables: perceived ease of use and evaluators' perceived satisfaction. Both were calculated by closed questions from a five-point Likert-scale questionnaire (i.e., arithmetic mean from 5 questions assigned to each variable), which also includes open-questions to obtain feedback from the evaluators.

The hypotheses of the experiment were the following:

- $H1_0$: There is no significant difference between the effectiveness of WUEP and HE / $H1_a$: WUEP is significantly more effective than HE.
- $H2_0$: There is no significant difference between the efficiency of WUEP and HE / $H2_a$: WUEP is significantly more efficient than HE.
- $H3_0$: There is no significant difference between the perceived ease of use of WUEP and HE / $H3_a$: WUEP is perceived to be significantly easier to use than HE.
- $H4_0$: There is no significant difference between the evaluators' perceived satisfaction of applying WUEP and HE / $H4_a$: WUEP is perceived to be significantly more satisfactory to use than HE.

The results of the experiment show that WUEP was more effective and efficient than HE in the detection of usability problems in artifacts obtained using a specific model-driven Web development process. In addition, the evaluators were satisfied when they applied WUEP, and found it easier to use than HE. Preliminary results of this experiment have been reported in [11]. The experimental material is available for download at `http://www.dsic.upv.es/~afernandez/MODELS13/instrumentation`.

5.2 The Experiment Replication (REP)

We conducted a strict replication of the experiment using a group of more experienced students in software modeling (i.e., Master students). The same materials used in the original experiment were used in the replication experiment. Strict replications are needed to increase confidence in the conclusion validity of the experiment. The subjects were 24 students enrolled on the "Quality of Web Information Systems" course on the Masters in Software Engineering, Formal Methods and Information Systems at the UPV. The alternative hypotheses tested were the same as the original experiment. It also was analyzed the order influence of the method and the two experimental objects employed.

The experiment was planned as a balanced within-subject design with a confounding effect, signifying that the same subjects use both methods in a different order and with different experimental objects (the subjects' assignation was random). Table 4 shows the schedule of the experiment operation in more detail. In addition, before the controlled experiment, a control group was created in order to provide an initial list of usability problems by applying an ad-hoc inspection method, and to

determine whether the usability problems reported by the subjects were real or false positives. This group was formed of two independent evaluators who are experts in usability evaluations, and one of the authors of this paper. Several documents were designed as instrumentation for the experiment: slides for training session, an explanation of the methods, gathering data forms, and two questionnaires.

Table 4. Schedule of the replication experiment

	Group 1 (6 subjects)	Group 2 (6 subjects)	Group 3 (6 subjects)	Group 4 (6 subjects)
1st Day (120 min)	1st: WebML Introduction; 2nd: Training with HE; and 3rd: Training with WUEP			
2nd Day (30 + 90 min)	1st:WebML Introduction; 2nd:Training with WUEP; and 3rd Training with HE			
	WUEP in O1	WUEP in O2	HE in O1	HE in O2
	Questionnaire for WUEP		Questionnaire for HE	
3rd Day (30 + 90 min)	1st: WebML Introduction; 2nd: Training with HE; and 3rd: Training with WUEP			
	HE in O2	HE in O1	WUEP in O2	WUEP in O1
	Questionnaire for HE		Questionnaire for WUEP	

6 Analysis of Results

After the execution of each experiment, the control group analyzed all the usability problems detected by the subjects. If a usability problem was not in the initial list, this group determined whether it could be considered as a real usability problem or a false positive. Replicated problems were considered only once. Discrepancies in this analysis were solved by consensus.

6.1 Quantitative and qualitative results

The quantitative analysis was performed by using the SPSS v16 statistical tool and $\alpha = 0.05$. Table 5 summarizes the overall results of the usability evaluations. Mean and standard deviation were used also for the subjective variables being the five-point Likert scale adopted for their measurement as an interval scale [5].

Table 5. Overall results of the usability evaluations from both experiments

Statistics	**Method**	**EXP (N=30)**		**REP (N=24)**	
		Mean	**SD**	**Mean**	**SD**
Number of problems per subject	HE	3.29	1.08	4.29	0.99
	WUEP	**6.50**	1.14	**6.91**	1.24
False positives per subject	HE	1.38	1.24	1.91	1.24
	WUEP	**0.54**	0.66	**0.29**	0.46
Replicated problems per subject	HE	0.88	0.80	1.50	0.93
	WUEP	**0.00**	0.00	**0.00**	0.00
Duration (min)	HE	**70.13**	13.52	**67.66**	14.01
	WUEP	80.88	18.46	72.75	11.14
Effectiveness (%)	HE	33.04	10.85	37.24	8.04
	WUEP	**65.32**	11.54	**60.16**	10.32
Efficiency (Problems / min)	HE	0.05	0.02	0.06	0.02
	WUEP	**0.08**	0.02	**0.09**	0.02
Perceived Ease of Use	HE	3.38	0.73	3.73	0.76
	WUEP	**3.80**	0.72	**3.94**	0.65
Perceived Satisfaction of Use	HE	3.63	0.67	3.74	0.73
	WUEP	**3.92**	0.75	**4.08**	0.53

The overall results obtained have allowed us to interpret that WUEP has achieved the subjects' best performance in about all the analyzed statistics (see cells in bold), The only exception is the duration of the evaluation session, which however was longer for WUEP due to the longer time required to read the material containing the WUEP description. As indicated by the results, WUEP tends to provide a low degree of false positives and replicated problems. The lack of false positives can be explained by the fact that WUEP tends to minimize the subjectivity of the evaluation. The lack of replicated problems can be explained by the fact that WUEP provides operationalized measures that are classified to be applied in one type of Web artifact.

The boxplots with the distribution of each dependent variable per subject per method (see Figure 3) show that WUEP was more effective and efficient than HE, and WUEP was also perceived by the evaluators as being easier to use and more satisfactory than HE.

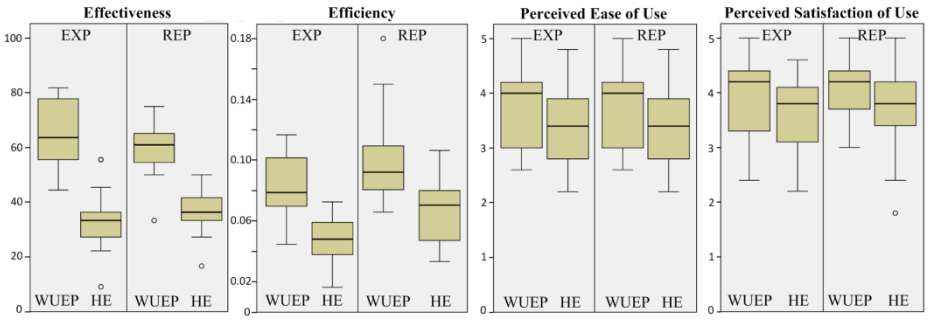


Fig. 3. Boxplots for each dependent variable in both experiments

We applied the Shapiro-Wilk test to verify whether the data was normally distributed with the aim to select which tests are needed in order to determine whether or not these results were significant. Table 6 provides the results of all the hypothesis verifications. We applied the Mann-Whitney non-parametric test for variables that resulted not normally distributed (i.e., In EXP: Effectiveness(WUEP) with p-value 0.021; and in REP: Efficiency(WUEP) with p-value 0.011, and Perceived Ease of Use(HE) with p-value 0.012). We applied the 1-tailed *t*-test for variables that resulted normally distributed. All the alternative hypotheses were accepted except H4 in EXP and H3 in REP. We believe this may be caused owing the subjects would need more training with WebML artifacts in order to perceived it more useful.

Table 6. *p*-values obtained for the test of hypothesis

		Significance Test	***p*-value**	**Accept Alternative Hypothesis?**
EXP	**H1**	Mann-Whitney	**0.000** (< 0.05)	**YES** *(WUEP more effective than HE)*
	H2	1-tailed t-test	**0.000** (< 0.05)	**YES** *(WUEP more efficient than HE)*
	H3	1-tailed t-test	**0.026** (< 0.05)	**YES** *(WUEP more easier to use than HE)*
	H4	1-tailed t-test	0.086 (> 0.05)	NO *(no significant differences in satisfaction)*
REP	**H1**	1-tailed t-test	**0.000** (< 0.05)	**YES** *(WUEP more effective than HE)*
	H2	Mann-Whitney	**0.000** (< 0.05)	**YES** *(WUEP more efficient than HE)*
	H3	Mann-Whitney	0.202 (> 0.05)	NO *(no significant differences in ease of use)*
	H4	1-tailed t-test	**0.036** (< 0.05)	**YES** *(WUEP more satisfactory than HE)*

In order to strengthen our analysis, we used the method suggested in [4] to test the effect of the order of both independent variables (usability evaluation methods and experimental objects). We used the Diff function: $Diff_x$ = $observation_x$(A) - $observation_x$(B), where x denotes a particular subject, and A, B are the two possible values of one independent variable. We created Diffs variables from each dependent variable. Finally, we verified that there were no significant differences between Diff functions since that would signify that there was no influence in the order of the independent variables (all the *p*-values obtained were > 0.05).

Finally, a qualitative analysis was performed by analyzing the open-questions that were included in the questionnaire. This analysis revealed some important issues which can be considered to improve WUEP (e.g., "*WUEP might be more useful if it were automated by a tool, especially the calculation of certain metrics*"), and it also collected positive impressions from the participants (e.g., "*I was surprised because I was able to systematically detect usability problems without previous experience*").

6.2 Threats to the Validity

The main threats to the internal validity of the experiment are: learning effect, evaluation design, subject experience, method authorship, and information exchange among evaluators. The learning effect was alleviated by ensuring that each subject applied each method to different experimental objects, and all the possible order combinations were considered. The evaluation design might have affected the results owing to the selection of attributes to be evaluated during the design stage of WUEP. We attempted to alleviate this threat by considering relevant usability attributes involving experts. Subject experience was alleviated due to the fact that none of the subjects had any experience in usability evaluations. The possibility of students knowing about our WUEP's authorship might have biased the results. We attempted to alleviate this threat by not disclosing more information; we also intend to conduct external replications with different conductors. Information exchange might have affected the results since the experiment took place over two days, and it is difficult to be certain whether the subjects exchanged any information with each other.

The main threats to the external validity of the experiment are: representativeness of the results, and duration of the experiment. Despite the fact that the experiment was performed in an academic context, the results could be representative with regard to novice evaluators with no experience in usability evaluations. However, the previous selection of usability attributes with their operationalized measures and the selection of the Web application might have affected the representativeness. To alleviate these issues, we intend to carry out a survey with Web designers to determine the relative importance of the usability attributes for different categories of Web applications. Since the duration of the experiment was limited to 90 min, only 2 representative software artifacts were selected from the different available types , although WUEP can be instantiated in more artifacts such as layout position-grids and style-templates.

The main threats to the construct validity of the experiment are: measures that are applied in the quantitative analysis and the reliability of the questionnaire. Measures that are commonly employed in this kind of experiment were used in the quantitative analysis [8]. The reliability of the questionnaire was tested by applying the Cronbach

test. Questions related to the Perceived Ease of Use obtained a Cronbach's alpha of 0.80 and 0.82, in EXP and REP respectively, whereas Perceived Satisfaction of Use obtained a Cronbach's alpha of 0.78 and 0.75, in EXP and REP respectively. These values are higher than the acceptable minimum (0.70) [21].

The main threat to the conclusion validity of the experiment is the validity of the statistical tests applied. This was alleviated by applying the most common tests that are employed in the empirical software engineering field [17].

7 Discussion and outlook

This paper presented the operationalization and empirical validation of a usability inspection method (WUEP) for its use within the WebML development process. From a practical point of view, our usability inspection strategy enables the development of more usable Web applications *by construction* [1]. Usability by construction means that each model built at different stages of a model-driven Web development process (PIM, PSM, Code) satisfies a certain level of usability of the corresponding Web application, thereby reducing the effort of fixing usability problems when the Web application is generated.

The effectiveness, efficiency, perceived ease of use and satisfaction of WUEP were compared in two experiments against a widely-used inspection method: Heuristic Evaluation (HE). The results show that WUEP was more effective and efficient than HE in the detection of usability problems in WebML models. Although the evaluators found it easier to use than HE and they were also more satisfied when applying WUEP, these variables resulted not statistically significant in some cases. These results confirmed our previous findings [12] when an instantiation of WUEP into the OO-H method was compared against HE, strengthening the case for using WUEP rather than HE, at least in contexts with fairly inexperienced usability evaluators. Although the experimental results provided good results on the usefulness of WUEP as a usability inspection method for Web applications developed through MDWD processes, we are aware that more experimentation is needed to confirm these results, since they need to be interpreted with caution being them only valid within the context established in these experiments. However, the replication presented here significantly adds to the existing validation of WUEP. We also obtained valuable feedback from these experiments based on which we can improve our proposal.

As future work, we plan to replicate this experiment with practitioners with different level of experience in usability evaluations, and to analyze in depth the empirical evidences collected by identifying which type of usability problems are most detected in models in order to suggest new mechanisms (modeling primitives, model-transformations, or patterns) to directly support some usability attributes. We also plan to validate the completeness of problem prediction through experiments in which the results of the evaluations obtained at the model level will be compared to the ones obtained when users interact with the generated Web applications.

Acknowledgements. This paper has been funded by the MULTIPLE project (MICINN TIN2009-13838) and the Erasmus Mundus Programme of the European Commission under the Transatlantic Partnership for Excellence in Engineering – TEE Project.

References

1. Abrahão, S., Iborra, E.,Vanderdonckt, J.: Usability Evaluation of User Interfaces Generated with a Model-Driven Architecture Tool. In: Maturing Usability: Quality in Software, Interaction and Value, Springer, pp. 3–32 (2007)
2. Atterer, R., Schmidt, A.: Adding Usability to Web Engineering Models and Tools. In: Proceedings of the 5th International Conference on Web Engineering (ICWE'05), pp. 36–41(2005)
3. Basili, V., Rombach, H.: The TAME Project: Towards Improvement-Oriented Software Environments. In: IEEE Transactions on Software Engineering 14(6), pp. 758-773 (1988)
4. Briand, L., Labiche, Y., Di Penta, M., Yan-Bondoc, H.: An experimental investigation of formality in UML-based development. IEEE TSE, 31(10), pp. 833–849 (2005)
5. Carifio, J., Perla, R.: Ten Common Misunderstandings, Misconceptions, Persistent Myths and Urban Legends about Likert Scales and Likert Response Formats and their Antidotes. In: Journal of Social Sciences 3(3), pp. 106-116 (2007)
6. Ceri, S., Fraternali, P., Bongio, A.: Web modeling language (WebML): a modeling language for designing Web sites. In: 9th International World Wide Web Conference, pp. 137–157 (2000)
7. Ceri, S., Fraternali, P., Acerbis, R., Bongio, A., Butti, S., Ciapessoni, F., Conserva, C., Elli, R., Greppi, C., Tagliasacchi, M., Toffetti, G.: Architectural issues and solutions in the development of data-intensive Web applications. In: Proceedings of the 1st Biennial Conference on Innovative Data Systems Research, Asilomar, CA (2003)
8. Conte, T., Massollar, J., Mendes, E., Travassos, G. H.: Usability Evaluation Based on Web Design Perspectives. In: Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM'07), pp. 146–155 (2007)
9. Fernandez, A., Insfran, E., Abrahão, S.: Usability evaluation methods for the Web: a systematic mapping study. In: Information and Software Technology 53, pp. 789–817 (2011)
10. Fernandez, A., Abrahão, S., Insfran, E.: A Web usability evaluation process for model-driven Web development. In: Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE'11). Springer, pp. 108–122 (2011)
11. Fernandez, A., Abrahão, S., Insfran, E., Matera, M.: Further Analysis on the Validation of a Usability Inspection Method for Model-Driven Web Development. In: 6th International Symposium on Empirical Software Engineering and Measurement (ESEM'12), pp. 153–156 (2012)
12. Fernandez, A., Abrahão, S., Insfran, E.: Empirical Validation of a Usability Inspection Method for Model-Driven Web Development. In: Journal of Systems and Software 86, pp. 161–186 (2013)
13. Fraternali, P., Matera, M., Maurino, A.: WQA: an XSL Framework for Analyzing the Quality of Web Applications. In: Proceedings of IWWOST'02 - ECOOP'02 Workshop, Malaga, Spain (2002)
14. Hornbæk, K.: Dogmas in the assessment of usability evaluation methods. In: Behaviour & Information Technology 29 (1), pp. 97–111 (2010)
15. Hwang, W., Salvendy, G.: Number of people required for usability evaluation: the 10±2 rule. In: Communications of the ACM 53(5), pp. 130–13 (2010)
16. International Organization for Standardization: ISO/IEC 25000, Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE (2005)

17. Juristo, N., Moreno, A.M.: Basics of Software Engineering Experimentation. In: Kluwer Academic Publishers (2001)
18. Juristo, N., Moreno, A., Sanchez-Segura, M.I.: Guidelines for eliciting usability functionalities. In: IEEE Transactions on Software Engineering 33 (11), pp. 744–758 (2007)
19. Matera, M., Costabile, M. F., Garzotto, F., Paolini, P.: SUE inspection: an effective method for systematic usability evaluation of hypermedia. In: IEEE Transactions on Systems, Man, and Cybernetics, Part A 32 (1), pp. 93–103 (2002)
20. Matera, M.; Rizzo, F., Carughi, G.: Web Usability: Principles and Evaluation Methods. In: Web Engineering, Springer, pp. 143–180 (2006)
21. Maxwell, K.: Applied Statistics for Software Managers. In: Software Quality Institute Series, Prentice Hall (2002)
22. Molina, F., Toval, A.: Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems. In: Advances in Engineering Software 40 (12), pp. 1306–1317 (2009)
23. Moreno, N., Vallecillo, A.: Towards interoperable Web engineering methods. In: Journal of the American Society for Information Science and Technology 59 (7), pp. 1073–1092 (2008)
24. Neuwirth, C.M., Regli, S.H.: IEEE Internet Computing Special Issue on Usability and the Web 6 (2), (2002)
25. Nielsen, J.: Heuristic evaluation. In: Usability Inspection Methods, John Wiley & Sons, NY (1994)
26. Offutt, J.: Quality attributes of Web software applications. In: IEEE Software: Special Issue on Software Engineering of Internet Software, pp. 25–32 (2002)
27. Panach, I., Condori, N., Valverde, F., Aquino, N., Pastor, O.: Understandability measurement in an early usability evaluation for MDD. In: International Symposium on Empirical Software Engineering (ESEM'08), pp. 354–356 (2008)
28. Webratio. Success stories. http://www.webratio.com/portal/content/en/success-stories (Online article)
29. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Weslen, A.: Experimentation in Software Engineering - An Introduction, Kluwer (2000)