Towards an Early Usability Evaluation for Web Applications*

Jose Ignacio Panach, Nelly Condori-Fernández, Francisco Valverde, Nathalie Aquino, Oscar Pastor

Department of Information Systems and Computation
Technical University of Valencia
Camino de Vera s/n, 46022 Valencia, Spain.
{jpanach, nelly, fvalverde, naquino, opastor}@dsic.upv.es
Phone: +34 96 387 7000, Fax: +34 96 3877359

Abstract. In the Human-Computer Interaction (HCI) community, the usual way to measure usability is through a user test. The disadvantage of this way is that the system must be implemented before performing the test. As a consequence, developers must solve the usability issues in the last stages of the development process. Currently, the model-driven software development is gaining popularity as a solution to reduce changes impact. In this paper, a usability model is proposed to evaluate early usability from the conceptual schemas that represents a Web Application. This approach allows to incorporate usability improvements before the implementation of the final web application. We evaluate the usability of artefacts modelled with OOWS, a model-driven web engineering method. In addition, two case studies are used to verify the instruments proposed to evaluate our early usability model.

Keywords: Usability, conceptual models, metrics, usability patterns, Web engineering methods, automatic code generation.

1. Introduction

According to ISO/IEC 9126-1 [12], usability is one of the characteristics that define Software Quality together with Functionality, Reliability, Efficiency, Maintainability, and Portability. The user can reject a system with poor usability even though functionality is adequate. Usability is therefore an essential software characteristic.

In order to decide whether a system is usable or not, metrics need to be defined to measure system usability. In the past, most of these measurements have been carried out in the final system [27][8] by means of user testing. However, with this approach, changes aiming at improving the system are difficult to apply because they affect several phases throughout the software development process. Moreover, user testing re-

* This work has been developed with the support of MEC under the project SESAMO TIN2007-62894.

quires too much time and effort to carry it out, so few software systems can evaluate their usability. For this reason we propose an early usability evaluation, on the basis of an existing generic usability model [7][1]. This paper uses the usability model in order to evaluate the entire OOWS generation process [9], a development method that allows automatic generation of Web applications from a Conceptual Model through a Model Compiler. The proposed usability evaluation is based on the conceptual primitives that represent the system.

The usability model includes a set of sub-characteristics that are composed of several attributes. The attributes can be measured at three levels: conceptual model, model compiler, and final system.

- The attributes from the generated system are measured with questionnaires that are completed by the users. These attributes represent subjective usability aspects.
- The attributes that measure usability aspects inserted by the Model Compiler are applied in each generated system in the same way. These attributes measure usability aspects that do not depend on the analyst choices; they only depend on a decision of the Model Compiler. Therefore, it has not sense to measure these attributes, because their values are always the same in all the systems.
- The attributes that can be measured in the Conceptual Model represent the measurable attributes in an *early evaluation*. The usability measure of these attributes depends on the way that the Conceptual Model is built.

This paper focuses on this last attributes group. Early usability evaluation proposed includes some attributes that affects only the Conceptual Model. Figure 1 shows as example that *Understandability* sub-characteristic is composed of *Brevity*, *Readability* and *Legibility* attributes, and each of these attributes is measurable at a different level.

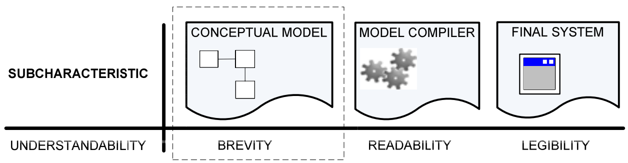

Fig. 1. Decomposition of sub-characteristics in attributes

The contributions of this paper are:

(1) to detect measurable attributes in the Conceptual Model;
(2) to define metrics to measure the usability of the detected attributes;
(3) to interpret the measures obtained;
(4) to add together all the indicators to obtain the usability of the entire system.

The paper is structured as follows. In section 2, this paper presents a survey of other usability models proposed in the literature. Section 3 describes the web engineering method used for the early usability evaluation. Section 4 shows a proposal to measure usability in the Conceptual Model. Section 5 explains the a priori validation of the early usability model proposed. Finally, we discuss conclusions and future work.

2. Other Usability Models

In recent years, some usability models have been proposed in the Software Engineering (SE) community and in the Human-Computer Interaction (HCI) community. On the one hand, the SE community has defined a standard usability model called ISO/IEC 9126-1 [12]. On the other hand, in the HCI community, ISO 9241-11 [11] explains how usability can be specified in terms of user performance and satisfaction. Both points of view are general; therefore, these models must be made more specific in order to be able to use them in the usability measure of a specific system.

Some authors have proposed their own usability models, such as Dromey [6]. He defined two properties: *behaviours* and *uses*. Behaviour is something that the software exhibits when it is executed under the influence of a set of inputs. A use is what different interest groups do with the software. In the same way, Dix et al. [5] propose a usability model based on three main categories: *learnability, flexibility,* and *robustness*. This model starts from categories and arrives at specific principles. This approach is similar to the usability model that is proposed in this paper. In our proposed model, the categories are equivalent to sub-characteristics, and specific principles are equivalent to attributes.

The model proposed by Nielsen [20] is more detailed than the Dix model. It focuses on social and practical acceptability. Usability is considered to be a sub-characteristic of usefulness, which is, in turn, a sub-characteristic of practical acceptability. A similar model, which only differs from the Nielsen model in the terminology used, is proposed by Shneiderman [25]. Shneiderman defines usability as "five measurable human factors that are central to the evaluation of human factor goals". These factors are: *speed of performance*, *learning time*, *retention over time*, *rate of errors by users*, and *subjective satisfaction*.

Moreover, there are other proposals based on standard extensions, like the Van Welie et al. [27] model. These researchers propose a model with three related layers: *Usability, Usage indicators,* and *Means*. A similar approach is proposed by Fitzpatrick [8], who defines three strands to identify the attributes of usable software: *Software quality, Statutory obligations,* and *Human-computer interaction*. The Van Welie and Fitzpatrick models are centred on user tests, ignoring an early evaluation such as the one proposed in our work.

Finally, some usability measures have been designed exclusively for Web environments. An example is the Web-site QEM methodology defined by Olsina [22]. This methodology is essentially quantitative, flexible, and robust, covering most of the activities in the evaluation, comparison, and ranking process of websites. However, this methodology is related to the final interface and cannot be used to measure usability in Conceptual Modelling.

In the next section we introduce OOWS, a web engineering method based on transformation models.

3. OOWS: a Development Method for Web Environment

OO-Method [23] is an object oriented software production method that generates information systems automatically. OO-Method models the system in different abstraction levels, distinguishing between problem space (the most abstract level) and solution space (the lowest abstract level). The system is represented by a set of Conceptual Models: 1) A Class Diagram that represents the static structure; and 2) The State and Functional Diagrams that represents the behaviour. OOWS [9] is the OO-Method extension in order to model and generate Web Applications. The OOWS conceptual schema is defined from two models that describe the different concerns of a web application: the Navigational Model and the Presentation Model.

- **Navigational Model**: This model defines the system navigational structure. It describes the navigation allowed for each type of user by means of a Navigational Map (Fig. 2. right). This map is depicted by means of a directed graph whose nodes represent navigational contexts and their arcs represent navigational links that define the valid navigational paths over the system. Navigational contexts are made up of a set of **Abstract Information Units** (AIU) (Fig. 2. left), which represent the requirement of retrieving a chunk of related information. AIUs are views defined over the underlying OO-Method Class Diagram. These views are represented graphically as UML classes that are stereotyped with the *«view»* keyword and that contain the set of attributes and operations that will be available to the user. Basically, an AIU represents a web page of the Web Application at conceptual level. Figure 2 shows an example of AIU and a Navigational map.
  Sometimes, the retrieved information is difficult to manage due to its size. To simplify the information browsing, two search mechanisms are provided: indexes and filters. Indexes create a list of summarized information using an attribute from the AIU. For instance, an index by year shows a list of years to the user. When the user selects a concrete year, only the books from that year are shown. Additionally, filters define a population condition to restrict the object instances to be retrieved depending on a value provided by the user. Thanks to filters, the user can retrieve only the books from a year inserted by him. These two search mechanisms are valuable from a usability point of view.
- **Presentation Model**: Thanks to this model we are able to specify the visual properties of the information to be shown. To achieve this goal, a set of presentation patterns are proposed to be applied over our conceptual primitives. Examples of presentation patterns are: (1) Information Layout: to arrange information, OOWS provide three basic layout patterns, register, tabular, and tree, and one layout operator, master-detail; (2) Pagination: this mechanism allows us to specify information by "scrolling". All the retrieved instances are broken into logical blocks of a given cardinality; (3) Ordering Criteria: this pattern defines a class population ordering (ASCendant or DESCendant) using the value of one or more attributes.

Both models are complemented by OO-Method models that represents functional and persistence layers. OOWS generates the code corresponding to the user interaction layer, and OlivaNOVA [3], the industrial OO-Method implementation, generates the business logic and the persistence layers.

OOWS development process is compliant with Model Driven Architecture (MDA) principles [17], where a Model Compiler transforms a Platform Independent Model

(PIM) into its corresponding Software Product. This MDA transformation process has been implemented by means of a case tool and a model compiler. Further details can be found in [26].

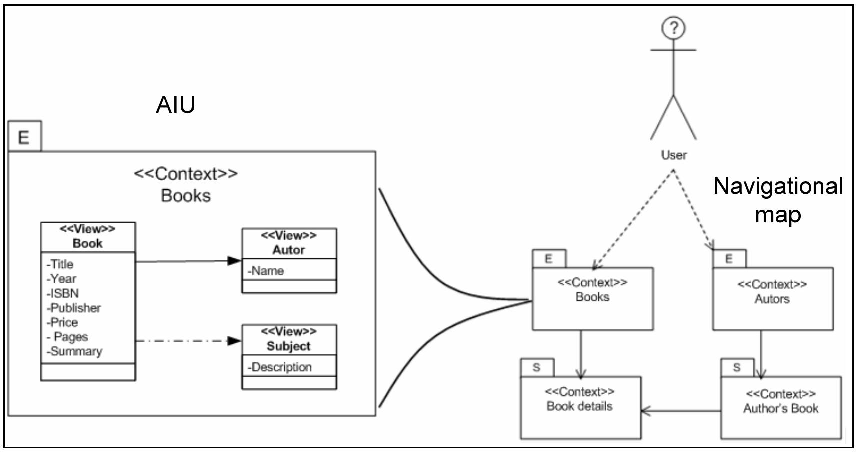

Fig. 2. Example of AIU and Navigational map

4. A Method for Measuring Usability in Software Systems

In previous work, a usability model based on the ISO 9126-1 [12] was defined to establish the aspects that make up system usability [7][1]. This usability model has two primitives: sub-characteristics and attributes. A s*ub-characteristic* represents a general aspect related to usability while an *attribute* is a measurable aspect of the usability. Sub-characteristics are composed of several attributes. Therefore, the usability model has a hierarchical decomposition. This decomposition starts with sub-characteristics and ends with measurable attributes. Sub-characteristics were extracted from the sub-characteristic of the ISO 9126-1. Attributes were extracted from several usability taxonomies as Bastien and Scapin's work [2]. The proposed method for measuring usability is based on these measurable attributes of the usability model.

The process of defining an early evaluation includes: a set of *metrics* to assign a value to each usability attribute; a number of *indicators* to classify the generated metrics; and the usability *results* extracted from the indicator of each attribute.

Each metric assigns a quantitative value for each attribute from the usability model. The metrics for each attribute are the following:

1. Action determination: This attribute is used to help the user to understand the meaning of his/her future actions. This attribute is related to the learnability sub-characteristic. In OOWS, the specification of an alias (a large description) for a Filter, a Context and a Service corresponds to this attribute. Filter, Context and Services with alias helps to obtain a more usable system. In case of the analyst does not define an alias, the Model Compiler uses the identifier of the Filter, the Context or the Service as alias.

2. Labelling significance: This attribute is used to measure whether or not a label has significance and is related to the learnability sub-characteristic. In OOWS, this attribute can be measured counting the number of attributes with alias defined in the classes of the OO-Method Class Diagram. For the attributes without alias, the Model Compiler assigns the identifier of the attribute as alias.
3. Brevity: This attribute, which belongs to the understandability sub-characteristic, measures the cognitive effort of the user. Some studies have demonstrated that the human memory has the capacity to retain three different scenarios [15]. Therefore, each reachable context through more than three contexts decreases the usability.
4. Information density: This attribute, which is related to understandability, measures whether too much information is shown in an interface. It takes several measures into account:
   - The more attributes there are in a context view, the more information density there will be. Some Web guidelines [16] recommend writings with a line of characters between 75 and 100 characters in order to avoid using scroll. In other words, the context should include at most ten attributes (taking an average of seven characters per attribute).
   - The same criterion can be used for the number of services. If a context has more than seven services, the user must use the scroll [16] because they do not fit in the interface. Therefore, when the service number in a view context exceeds seven, usability decreases.
   - Seven or less navigations from the same context provide plenty of information [18]. Therefore, contexts with more than seven navigations decrease usability because they provide information in exceed and not useful.
   - Definition of Filters and Indexes help readability, but they should not contain more than three variables [16] to avoid confusing the user. If the analyst has to provide different alternatives for filtering or indexing, he/she should define several Filters or Indexes that only contains three variables at most.
   - Pagination is another mechanism to make the list of instances readable. Information should be arranged so that the user does not need to use the scroll [16]. Twenty instances fit in an interface at most, therefore more than twenty instances by page decreases usability.
5. Message concision: This attribute indicates whether a message shown to the user is clear. This attribute, which belongs to the understandability sub-characteristic, is significantly related to semantic aspects, which cannot be measured in the Conceptual Model. However, a criterion to automatically measure this attribute could be the number of words used in the message. Error messages should not contain more than 20 words, according to [16].
6. Navigability: This attribute indicates whether navigation is easy and it belongs to the understandability sub-characteristic. Two measures are used for this attribute:
   - The maximum number of navigations to reach a context. According to [16], more than three navigations decrease usability.
   - System functionality should be organized into submenus so that no more than seven menus [18] are shown to the user.

Table 1 shows the metrics defined for each attribute in a formal notation. The attributes are grouped by the sub-characteristics to which they belong.

Table 1. Summary of metrics for each attribute. Notation used: (S)Service; (F)Filter; (C)Context; (A)Attribute

Sub-characteristic	Attribute	Measure
Learnability	Action determination (AD)	$\forall x \in Alias(S,F,C): \frac{\sum_{i=1}^{n} x_i}{\sum S + \sum F + \sum C} = AD$
	Labelling significance (LS)	$\forall x \in Alias(A): \frac{\sum_{i=1}^{n} x_i}{\sum A} = LS$
Understandability	Brevity (BR)	$\forall x, y \in Context: MinPath[x, y] = BR$
	Information density (ID)	$\forall x \in ContextAttribute: \sum_{i=1}^{n} x_i = ID1$
		$\forall x \in ContextService: \sum_{i=1}^{n} x_i = ID2$
		$\forall x \in Context: \sum OutDegree(x) = ID3$
		$\forall x \in Context\,(FilterVariable \vee Index): \sum_{i=1}^{n} x_i = ID4$
		$Pagination\_Cardinality = ID5$
	Message concision (MC)	$\forall x \in PreconditionText(Word): \sum_{i=1}^{n} x_i = MC$
	Navigability (NV)	$\forall x, y \in Context: MaxPath[x, y] = NV1$
		$\forall x \in Context: Path[Root, X] = 1 \rightarrow \sum_{i=1}^{n} x_i = NV2$

These metrics provide a number to measure the usability, but this number needs a meaning. Therefore, we must define **indicators** in order to establish how useful the measure obtained by means of the metrics is. For this purpose, we have defined five ranks for each metric. Each one of these ranks represents a qualitative value to show the degree of learnability or understandability of the measured attribute. Depending on the rank in which the numeric value of the metric is, a qualitative value is assigned to the attribute. The qualitative values used are: Very Good (VG), Good (G), Medium (M), Bad (B), and Very Bad (VB). The determination of the ranks was based on certain usability criteria, such us [15],[16],[18] Table 2 shows the indicators for each attribute with a defined metric:

Table 2. Qualitative values for each obtained measured

Measure	VG	G	M	B	VB
AD	AD$\geq$.95	.95>AD$\geq$.85	.85>AD$\geq$.75	.75>AD$\geq$.65	AD<.65
LS	LS$\geq$.95	.95>LS$\geq$.85	.85>LS$\geq$.75	.75>LS$\geq$.65	LS<.65
BR	BR$\leq$2	2<BR$\leq$4	4<BR$\leq$5	5<BR$\leq$6	BR>6
ID1	ID1$\leq$8	8< ID1$\leq$10	10< ID1$\leq$12	12< ID1$\leq$16	ID1>16

Measure	VG	G	M	B	VB
ID2	ID2$\leq$5	5< ID2$\leq$7	7< ID2$\leq$10	10< ID2$\leq$13	ID2>13
ID3	ID3$\leq$5	5< ID3$\leq$7	7< ID3$\leq$10	10< ID3$\leq$13	ID3>13
ID4	ID4$\leq$2	2< ID4$\leq$4	4< ID4$\leq$6	6< ID4$\leq$8	ID4>8
ID5	ID5$\leq$20	20< ID5$\leq$30	30< ID5$\leq$40	40< ID5$\leq$50	ID5>50
MC	MC$\leq$15	15<MC$\leq$20	20<MC$\leq$25	25<MC$\leq$30	MC>30
NV1	NV1$\leq$3	4< NV1$\leq$6	6< NV1$\leq$8	8< NV1$\leq$9	NV1>9
NV2	NV2$\geq$.95	.95>NV2$\geq$.85	.85>NV2$\geq$.75	.75>NV2$\geq$.65	NV2<.65

The final step is to establish the usability of the sub-characteristics from the indicators defined for the attributes. Since a sub-characteristic is composed of several attributes, a method to group all the attributes is needed. The method chosen for this is rewriting terms, a method used in Quispe et al work [24]. This method builds a tree with the usability model: the leaves are the attributes, and the branches are the sub-characteristics. Transformations used in the rewriting terms collect leaves two by two to obtain a qualitative value applying a set of rules. The application of these rules is carried out down to top, until the root of the tree, which corresponds to the usability of the system. These rules are the following:

- If we have the Combination of two parameters, one with the Medium (M) value and the other with an X value, the result of this combination is X.

$$\forall x \in Indicators : Combination(x, M) \rightarrow x$$

- The rest of combinations are in Figure 3a:

$$\begin{aligned}
&Combination(VG,VB) \rightarrow M\\
&Combination(G,B) \rightarrow M\\
&Combination(VB,VB) \rightarrow VB\\
&Combination(VB,B) \rightarrow VB\\
&Combination(VB,G) \rightarrow VB\\
&Combination(B,B) \rightarrow VB\\
&Combination(G,G) \rightarrow VG\\
&Combination(VG,B) \rightarrow VG\\
&Combination(VG,G) \rightarrow VG\\
&Combination(VG,VG) \rightarrow VG
\end{aligned}$$

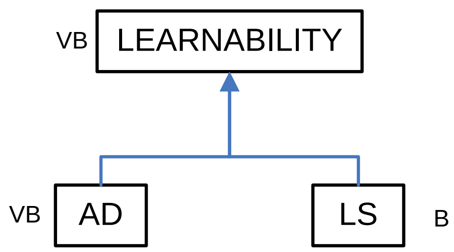

Fig. 3. (a) Combination rules for qualitative values. (b) Example of grouping attributes

For instance, if the measure AD has the indicator VB and the measure LS has the indicator B, the result of the sub-characteristic learnability is VB following the rewriting terms (Figure 3b). The value extracted using the transformation rules for each sub-characteristic comprises the results of the analysis.

All this evaluation method can be carried out automatically. Once the analyst has built the Conceptual Model, he/she can obtain the results of the usability evaluation and change the models in order to improve the usability of the represented system.

5. A priori Validation of the Early Usability Model

According to Morris and Desharnais [19], a priori validation includes two types of verification related to reliability and coherence of the documentation and the training process in order to assure the understandability of the method to be evaluated. In this section, we present the verification of the reliability of the instruments to be used in the evaluation process of the early usability model. Figure 4 summarizes this process, which consists of a comparison between two sets of values: 1) a first set obtained by an analyst applying an early Usability Model within OOWS conceptual schemas, and, 2) a set of values of perceived learnability and understandability for end users using web applications generated from previously evaluated conceptual schemas..

For the gathering of these values from analyst and end users, we prepare various experimental instruments, whose reliability are verified and described below.

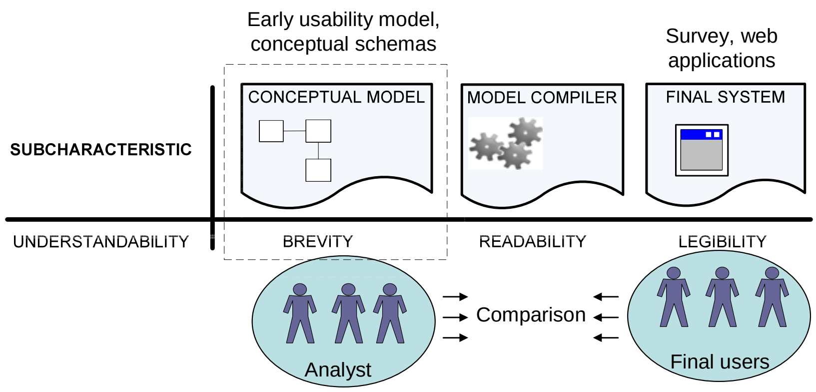

Fig. 4. General process for verifying an early usability model

5.1 Verifying the reliability of the instruments

With the aim of verifying the clarity of the metrics definition proposed in the early usability model; we selected five senior analysts with extensive experience in the use of the OOWS approach. These analysts measured the conceptual schemas of two case studies. The code from both examples (except aesthetic properties such as colour, font-family, background etc.) was automatically generated by OlivaNOVA and the OOWS Model Compiler:

- *Rent Car:* This web application has been generated from a conceptual model that represents an on-line car rental service. Its application domain is related to e-commerce, as a consequence the rental service must be attractive to potential customers. To simplify the interaction, users can only make car rents and introduce new vehicles into the system. This web application is in *http://oomethod.dsic.upv.es/rentcar*

- *IMDB Lite***:** This case study is based on an online repository of information related to the movie world: The Internet Movie Database (IMDB). This case study is a good example of massive information retrieval web. From usability point of view, is interesting to provide accurate mechanisms to find the required information. Only some functionality from the Website, have been implemented. For instance to see information about movies and actors, etc. and to make comments about them. This web application is in *http://oomethod.dsic.upv.es/imdblite*

Analysing the set of metrics (Table 1), we note a low reproducibility for the NV2, MC, and LS metrics. Other metrics showed a high reproducibility.

With respect to the survey instrument, it was based on existing surveys. These surveys are typically used to obtain information about user perceptions related to achieved satisfaction using prototypes or final software systems, such as WAMMI (Web site Analysis and MeasureMent Inventory) [13], ISOMETRICS [10], SUMI (Software Usability Measuring Inventory) [14], and QUIS (Questionnaire for User Satisfaction)[4]. We adapted these surveys in order to focus solely on the attributes that can be measured within the Conceptual Model (learnability and understandability sub-characteristics).

Our survey instrument included thirteen closed questions: four questions relating to learnability (I1,I5,I7,I11), and nine questions relating to understandability (I2,I3,I4,I6,I8,I9,I10,I12,I13). The items used were formulated using a 5-point Likert scale with the opposing-statement question format. The order of the items was randomized and half the questions negated to avoid monotonous responses. Figure 5 shows two items of the survey.

I3	It is easy to see at a glance the information and options available in a page	O	O	O	O	O	Typically, scroll bars are needed to be able to visualize all the information of a page
I4	Error messages clearly explain the causes of the problems	O	O	O	O	O	Error messages do not clearly explain the causes of the problems

Fig. 5. Two items of the survey

With the aim of verifying the reliability of the survey, eight users were first requested to accomplish several tasks using two generated Web Applications (*Rent Car* and *IMDB Lite*). *Rent Car* Web Applications had some usability problems while *IMDB Lite* was theoretically usable. Then, in order to capture the users' perceptions using these applications, users completed the specially designed survey.

We used an inter-item correlation analysis to evaluate the construct validity of the learnability and understandability variables. Tables 3 and 4 show the analysis made with data from the surveys of *Rent Car* and *IMDB Lite* applications, respectively. We employed two criteria, Convergent Validity (CV) (first yellow column) and Discriminant Validity (DV) (second yellow column), for each item; if convergent validity was higher than discriminant validity, the item would be validated (last column).

Table 3. Inter-item correlation analysis – *Rent Car* application

		AD			LS		BR		ID		MC			NV					
		I1	I7	I11	I5	I11	I2	I8	I3	I9	I4	I10	I13	I6	I8	I12	CV	DV	VALID
AD	I1	1,00	0,36	0,50	0,52	0,50	0,50	0,22	0,21	0,12	0,13	0,09	0,19	0,45	0,22	0,70	0,62	0,32	Yes
	I7	0,36	1,00	0,69	0,04	0,69	0,03	0,95	0,81	0,15	0,10	0,25	0,18	0,23	0,95	0,73	0,68	0,43	Yes
	I11	0,50	0,69	1,00	0,46	1,00	0,68	0,65	0,68	0,63	0,35	0,40	0,66	0,69	0,65	0,63	0,73	0,62	Yes
LS	I5	0,52	0,04	0,46	1,00	0,46	0,48	0,00	-0,06	0,49	0,70	0,50	0,20	0,17	0,00	0,35	0,73	0,30	Yes
	I11	0,50	0,69	1,00	0,46	1,00	0,68	0,65	0,68	0,63	0,35	0,40	0,66	0,69	0,65	0,63	0,73	0,63	Yes
BR	I2	0,50	0,03	0,68	0,48	0,68	1,00	0,00	0,35	0,53	0,42	0,39	0,65	0,73	0,00	0,11	0,50	0,43	Yes
	I8	0,22	0,95	0,65	0,00	0,65	0,00	1,00	0,81	0,27	0,12	0,19	0,25	0,19	1,00	0,65	0,50	0,46	Yes
ID	I3	0,21	0,81	0,68	-0,06	0,68	0,35	0,81	1,00	0,12	0,34	0,50	0,25	0,26	0,81	0,30	0,56	0,46	Yes
	I9	0,12	0,15	0,63	0,49	0,63	0,53	0,27	0,12	1,00	0,22	0,07	0,88	0,66	0,27	0,34	0,56	0,41	Yes
MC	I4	0,13	0,10	0,35	0,70	0,35	0,42	0,12	0,34	0,22	1,00	0,88	-0,03	-0,16	0,12	-0,12	0,62	0,21	Yes
	I10	0,09	0,25	0,40	0,50	0,40	0,39	0,19	0,50	0,07	0,88	1,00	-0,02	-0,06	0,19	-0,14	0,62	0,23	Yes
	I13	0,19	0,18	0,66	0,20	0,66	0,65	0,25	0,25	0,88	-0,03	-0,02	1,00	0,91	0,25	0,31	0,32	0,45	No
NV	I6	0,45	0,23	0,69	0,17	0,69	0,73	0,19	0,26	0,66	-0,16	-0,06	0,91	1,00	0,19	0,43	0,54	0,40	Yes
	I8	0,22	0,95	0,65	0,00	0,65	0,00	1,00	0,81	0,27	0,12	0,19	0,25	0,19	1,00	0,65	0,61	0,43	Yes
	I12	0,70	0,73	0,63	0,35	0,63	0,11	0,65	0,30	0,34	-0,12	-0,14	0,31	0,43	0,65	1,00	0,69	0,37	Yes

Table 4. Inter-item correlation analysis – *IMDB Lite* application

		AD			LS		BR		ID		MC			NV					
		I1	I7	I11	I5	I11	I2	I8	I3	I9	I4	I10	I13	I6	I8	I12	CV	DV	VALID
AD	I1	1,00	0,65	0,58	0,63	0,58	0,29	0,34	0,61	-0,44	0,27	0,33	0,48	0,61	0,34	0,85	0,74	0,40	Yes
	I7	0,65	1,00	0,18	0,69	0,18	0,44	0,92	0,18	0,10	0,41	0,50	0,73	0,66	0,92	0,55	0,61	0,52	Yes
	I11	0,58	0,18	1,00	0,31	1,00	0,06	-0,07	0,95	0,05	0,34	0,30	0,28	0,38	-0,07	0,76	0,58	0,36	Yes
LS	I5	0,63	0,69	0,31	1,00	0,31	0,31	0,47	0,32	0,08	0,28	0,42	0,27	0,84	0,47	0,63	0,65	0,44	Yes
	I11	0,58	0,18	1,00	0,31	1,00	0,06	-0,07	0,95	0,05	0,34	0,30	0,28	0,38	-0,07	0,76	0,65	0,36	Yes
BR	I2	0,29	0,44	0,06	0,31	0,06	1,00	0,44	0,06	0,00	0,10	0,29	-0,09	0,26	0,44	0,54	0,72	0,21	Yes
	I8	0,34	0,92	-0,07	0,47	-0,07	0,44	1,00	-0,07	0,30	0,45	0,51	0,72	0,43	1,00	0,27	0,72	0,40	Yes
ID	I3	0,61	0,18	0,95	0,32	0,95	0,06	-0,07	1,00	0,11	0,41	0,43	0,24	0,33	-0,07	0,79	0,55	0,40	Yes
	I9	-0,44	0,10	0,05	0,08	0,05	0,00	0,30	0,11	1,00	0,41	0,46	0,11	0,00	0,30	-0,11	0,55	0,10	Yes
MC	I4	0,27	0,41	0,34	0,28	0,34	0,10	0,45	0,41	0,41	1,00	0,95	0,63	-0,10	0,45	0,30	0,86	0,31	Yes
	I10	0,33	0,50	0,30	0,42	0,30	0,29	0,51	0,43	0,46	0,95	1,00	0,51	0,02	0,51	0,43	0,82	0,37	Yes
	I13	0,48	0,73	0,28	0,27	0,28	-0,09	0,72	0,24	0,11	0,63	0,51	1,00	0,24	0,72	0,24	0,71	0,35	Yes
NV	I6	0,61	0,66	0,38	0,84	0,38	0,26	0,43	0,33	0,00	-0,10	0,02	0,24	1,00	0,43	0,62	0,68	0,34	Yes
	I8	0,34	0,92	-0,07	0,47	-0,07	0,44	1,00	-0,07	0,30	0,45	0,51	0,72	0,43	1,00	0,27	0,57	0,41	Yes
	I12	0,85	0,55	0,76	0,63	0,76	0,54	0,27	0,79	-0,11	0,30	0,43	0,24	0,62	0,27	1,00	0,63	0,50	Yes

Item I8 appears in two rows and columns of the tables because corresponds to a question used to perceived two attributes of the usability model: *brevity (BR)* and *navigability (NV)*. In a similar way, I11 corresponds to *action determination (AD)* and *labelling significance (LS)*. The remaining items perceive just one attribute each one.

In the *Rent Car* analysis (see Table 3), the CV value was higher than the DV value for twelve items. So, the twelve items were validated. However, as item I13 had a CV value lower than the corresponding DV value, this item was not validated.

In the *IMDB Lite* analysis (see Table 4), the CV value was higher than the DV value for the thirteen items, all were validated, including item I13.

In addition, we also conducted a reliability analysis on the items to calculate the degree to which the values of the constructs are free of measurement error. The reliability analysis was carried out using the Chronbach alpha technique; the corresponding alpha value for learnability and understandability sub-characteristics is shown in Table 5.

Table 5. Reliability analysis for survey items

Sub-characteristic	Cronbach alpha	Number of Items
Learnability	0.732	4
Understandability	0.804	9
General	0.873	13

These values indicate that the items included in the survey are reliable, alphas of 0.7 or above being acceptable according to Nunally [21].

6. Conclusions and Future Work

This paper presents a usability evaluation method applicable to the analysis phase of the OOWS development process. This method is of particular interest within an industrial context because it allows easy improvement of systems usability. After a rapid analysis of usability, the analyst can change the Conceptual Models to improve the usability value. To carry out this process, a set of attributes that can be measured in the Conceptual Model are identified on the basis of a generic Usability Model [7][1], including also attributes that are measured in the Model Compiler or in the final system.

From all the set of attributes, this work is centred in the attributes that can be measured in a Conceptual Model. We define a set of metrics for each one of the usability attributes that can be measured in an early phase (learnability and understandability). A qualitative value is then assigned to each numeric value provided by these metrics in order to provide a meaning to them. This qualitative value is expressed on an ordinal scale with five levels; intervals for each level are defined according to usability criteria revised in the literature [15],[16],[18].

All the qualitative values obtained for each attribute are then grouped in order to obtain the degree of learnability and understandability. This value aggregation is carried out applying a set of rules proposed by Quispe et al.[24].

It is important to note that early usability forms only a part of the measurable usability in that there are many subjective attributes that can be measured only in the final system or in the model compiler. Nevertheless, this early usability evaluation is useful because it helps the analyst to predict the usability of applications generated with OOWS.

Finally, in this paper, we have verified the instruments to be used to evaluate an early usability model. We have found that the items included in the designed survey are reliable. However, we note that three metrics needed to be revised due to the low reproducibility obtained.

As a next step we plan to carry out an experimental study with Computer Science students in order to ensure that the metrics, indicators and aggregation rules have been correctly defined in order to evaluate the usability of OOWS conceptual schemas.

Acknowledgments

We would like to thank Alain Abran and the anonymous reviewers for his helpful comments and suggestions that contributed to the improvement of this paper.

References

[1] Abrahao, S., Insfrán, E. (2006). Early Usability Evaluation in Model Driven Architecture Environments. Sixth International Conference on Quality Software (QSIC'06).pp. 287-294.
[2] Bastien, J. M., Scapin, D. (1993). "Ergonomic Criteria for the Evaluation of Human-Computer Interfaces." Rapport technique de l'INRIA: 79
[3] CARE Technologies S.A. www.care-t.com
[4] Chin, J. P., Diehl, V. A, Norman, K. (1987). Development of an instrument measuring user satisfaction of the human-computer interface, Proc. ACM CHI '88 (Washington, DC) 213-218.
[5] Dix, A., Abowd, G., Beale, R. and Finlay, J. (1998), Human-Computer Interaction, Prentice Hall Europe.
[6] Dromey R.G. (1998) Software Product Quality: theory, Model and Practice, Technical report, Software Quality Institute, Griffith University, Nathan, Brisbane, Australia.
[7] España, S., Pederiva, I., Panach, I., Abrahão, S., Pastor, O. (2006). Evaluación de la Usabilidad en un Entorno de Arquitecturas Orientadas a Modelos. IDEAS 2006, Argentina.pp. 331-344.
[8] Fitzpatrick, R. and C. Higgins (1998). Usable Software and Its Attributes: A Synthesis of Software Quality, European Community Law and Human-Computer Interaction Proceedings of HCI on People and Computers XIII.Springer-Verlag: 3-21.
[9] Fons J., P. V., Albert M., and Pastor O (2003). Development of Web Applications from Web Enhanced Conceptual Schemas. ER 2003, LNCS. Springer.
[10] Gediga, G., Hamborg, K.-C., and Duntsch, I. (1999). "The IsoMetrics Usability Inventory: An Operationalisation of ISO 9241-10," Behaviour and Information Technology, Vol. 18, pp. 151-164.
[11] ISO 9241-11 (1998) Ergonomic requirements for office work with visual display terminals - Part 11: Guidance on Usability.
[12] ISO/IEC 9126-1 (2001), Software engineering - Product quality - 1: Quality model.
[13] Kirakowski, J.; Claridge, N. (1998). Human centered measures of success in web site design. In Proceedings of the Fourth Conference on Human Factors & the Web (Basking Ridge, NJ, June).
[14] Kirakowski, J.; Corbett, M. (1993). SUMI: The software usability measurement inventory. British Journal of Educational Technology 24, 3, 210-212.
[15] Lacob, M.-E. (2003). Readability and Usability Guidelines.Web:https://doc.telin.nl/dscgi/ds.py/ Get/File-35439
[16] Leavit, M., Shneiderman, B. (2006). Research-Based Web Design & Usability Guidelines, U.S. Government Printing Office. Web: http://www.usability.gov/pdfs/guidelines.html
[17] MDA: http://www.omg.org/mda Last visited: May-2007.

[18] Miller, G. A. (1956). "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." The Psychological Review 63: 81-97.

[19] Morris P., Desharnais J.M., Function Point Analysis. Validating the Results. in IFPUG Spring Conference, Atlanta, USA, 1996.

[20] Nielsen, J. (1993). Usability Engineering. Morgan Kaufmann Publishers Inc.

[21] Nunally J., Psychometric Theory, McGraw-Hill, 2nd ed., New York, NY1978.

[22] Olsina, L. (1999). Quantitative Methodology for Valuation and Comparison of Web Site Quality. Argentina, Phd. Facultad de Ciencias Exactas Universidad Nacional de La Plata.

[23] Pastor, Ó., Molina, J.C. Model Driven Architecture in Practice: A Software Production Environment base don Conceptual Modeling. Springer-Verlag. 2007.

[24] Quispe Cruz, M., Condori Fernández, N. (2007). Requisitos No Funcionales: Evaluando el Impacto de Decisiones. VI Jornada Iberoamericana de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC07), Universidad Pontificia Católica del Perú, Lima (Peru).

[25] Shneiderman, B. (1998), Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley, Reading, Massachusetts, USA.

[26] Valverde, F., P. Valderas, et al. (2007). A MDA-Based Environment for Web Applications Development: From Conceptual Models to Code. 6th International Workshop on Web-Oriented Software Technologies (IWWOST), Como (Italy).

[27] Welie, M. v., G. C. v. d. Veer and A. Eliëns (1999). Breaking Down Usability. NTERACT 99, Edinburgh. IOS Press, pp. 613-620.