An Empirical Evaluation of a Usability Measurement Method in a Model Driven Framework

Lassaad Ben Ammar[1] Adel Mahfoudhi[1]

University of Sfax, ENIS, CES Laboratory
Soukra Road km 3,5, B.P: 1173-3000 Sfax TUNISIA
benammar_lassad@hotmail.com
adel.mahfoudhi@ceslab.org

Abstract. Usability is increasingly considered as a basic determinant of the Interactive Systems (IS) success. An IS that satisfies all the functional requirements can be rejected by end-users if it presents usability problems. Unusable User Interface (UI) is probably the main reason that may lead to the failure in the actual use of an IS. Therefore, several approaches dealing with the evaluation of the user interface usability have been proposed in literature. However, these approaches are focused on the final system and require a large amount of resources to perform the evaluation (end-users, video cameras, questionnaires, etc.). The ability to go back and makes major changes to the design is greatly reduced. It is widely accepted that the evaluation performed at the beginning of the development process is a critical part of ensuring that the product will be used and effective for its intended purpose. In addition, an early usability evaluation would be a significant advantage with regard to saving time and resources.
The purpose of the present paper is to investigate the integration of the usability issues at an early stage of the development process. A model based approach is presented and empirically evaluated.

Keywords: Plastic User Interface, Usability Model, Empirical Evaluation, Model Driven Engineering.

1 Introduction

Usability denotes the ease of use of a system for a particular class of users carrying out specific tasks in a specific environment. It is widely considered as a basic determinant of the acceptance of an interactive system [1]. Unusable User Interface (UI) is probably the main reason that may lead to the failure in the actual use of an interactive system [2]. For that reason, a variety of approaches have been adopted in literature in order to evaluate the usability of user interfaces ([3], [4], [5], among them). However, these methods involve activities that require a huge amount of resources (usability experts, questionnaire, several end users, usability laboratory, etc). They are focused in the final product in order to

carry out the evaluation. Consequently, changes to the user interface are costly and difficult to implement.
In recent years, the emergency of a wide spectrum of interactive devices has raised new issues for user interface designers and developers. They are facing the challenge of producing not only a highly usable user interfaces but also that can be adapted in order to enable different classes of users to access information regardless of the interactive devices they are using even when the environment changes dynamically. [6] used the term *Plastic* to indicate this kind of user interfaces. Within this context, several approaches have been proposed. Those following the Model Driven Engineering[1] (MDE) [7] principles proved quite appropriate [8]. A renowned work in this context is the Cameleon projet [9] which provides a unifying reference framework for the user interface development taking into consideration the context of use wherein the interaction takes place. The main limitation of the Cameleon framework is that research efforts have focused only on the functional aspect of the user interface adaptation while neglecting usability. Usability is considered as a natural by-product of whatever approach was being used. Therefore, there is a need to expand the Cameleon framework in order to change perspective and make usability a first class entity in its development process.
The main objective of this paper is to promote the usability issue as a first class entity in the Cameleon framework. To do so, we propose to expand such framework by considering usability engineering as a part of the development process. We propose to carry out the evaluation from the conceptual models.The objective is to make usability evaluation independent from the system implementation and to reduce the development costs involved by measuring the usability late in the development process. The proposed usability evaluation method is based on a usability model which decomposes usability on measurable attributes and metrics. It is intended to isolate potential usability problems so as to determine a figure of merit of the overall interface.
The remainder of this paper is structured as follows. While Section 2 presents an outline of the usability methods quoted in the literature, section 3 provides a brief description of our proposed to integrate usability issue into an MDE method. The proposed usability model is described in Section 4, and the empirical study to evaluate the proposed usability model is illustrated in section 5. Finally, Section 6 presents the conclusion and provides perspectives for future research work.

2 Related Works

The usability evaluation is often defined in [10] as methodologies for measuring the usability aspects of a user interface and identifying specific problems. There exist several methods addressing the usability evaluation of the user interfaces.

[1] Model Driven Engineering MDE: a software approach which promotes a new form of building software systems based on the construction and maintenance of models at different levels of abstraction to drive the development process

In this section, the focal point is on the analysis of model-based methods since our main motivation is to integrate usability issues into a model driven development approach.
Model-based usability evaluation methods specify usability attributes and metrics required to assess the user interface in order to identify potential usability problems. Usability models are usually based on existing standards such as ISO/IEC 9126-1 [11] and ISO/IEC 9241-11 [12]. In fact, both standards are useful in providing principles and recommendations. However, they are abstract and need to be extended and / or decomposed for their use on different kinds of systems.
[13] proposed a usability model that extends the ISO/IEC 9126-1 model, which is intended to evaluate the usability of a user interface from the beginning of an MDE approach. The main limitation of this proposal is the lack of guidelines about how usability attributes are measured and how to interpret their scores. An extension of this model is proposed in [14] in order to assess web applications.
[2] reviewed the existing usability standards and models to detect their limitation and complementarities. As a result, a consolidated model (QUIM) based on the ISO/IEC 9241-11 standard is proposed. Other relevant characteristics such as Learnability and Security are extracted from ISO/IEC 9126-1 and other resources to enrich the model. The QUIM model includes metrics that are based on the system code as well as on the generated interface. This makes the application of the QUIM to a model driven development process difficult.
[15] evaluated the usability of multi-devices user interfaces in terms of effectiveness, efficiency and satisfaction. The usability evaluation is based on the experiments with end-users. This dependency is incompatible with an early usability evaluation.
[16] proposes an early usability measurement method. The usability evaluation is carried out early in the development process since the conceptual model. The main limitation of this proposal is that metrics are specific to the OO-method [17]. Therefore, they cannot be applied to other method, which is a disadvantage. They need some adaptation in order to be used (adopted) in other similar methods.
Considering the research works just mentioned, three main limitations are underlined. The first problem is the lack of measurement details. The proposals (except the proposition of Panach) specify usability attributes and metrics without defining how these metrics should be measured and how to interpret their scores. The second problem is the need for the system implementation. Most proposals carry out the evaluation at the last step of the development process which is incompatible with an automatic early evaluation. Regardless of the approach, none of them takes into account the variation of context elements during their process activities and the influence it brings to the selection of the most relevant attributes and metrics.
It becomes clear that integrating usability issues into an MDE method for plastic user interface generation is still an immature area. Therefore many more research works are needed. However, it should be noted that the aforementioned models

are useful in providing guidelines about the most relevant attributes required to measure the user interface usability. They can be a useful resource from which we draw our proposal model.

3 Proposed Method to integrate Usability into a Model Driven Development process

3.1 Overview

As already mentioned, the main motivation of this paper is to integrate usability issues as a part of the development process of the Cameleon framework. In fact, the choice of the Cameleon framework is motivated by the fact that such framework unifies models, methods and tools for the generation of user interfaces for multiple contexts of use. The Cameleon framework structures the development process into four levels of abstraction, starting from task specification to a running interface.

- The Task and Concepts: brings together the concepts and the tasks descriptions produced by the designers for that particular interactive system and that particular context of use.
- The Abstract User Interface (AUI): this level represents the user interface in terms of interaction spaces (or presentation units), independently of which interactors are available on the targets.
- The Concrete User Interface (CUI): this level turns an Abstract UI into an interactor-dependent expression.
- The Final User Interface (FUI): this level consists of source code, in any programming or mark-up language (e.g., Java, HTML, etc.).

In the Cameleon framework, conceptual models are a primary artifact in the analysis and design of an interactive system. They are used to define the user requirements and as a basis for developing interactive systems to meet these requirements. In the software engineering field, the quality of the conceptual models is usually neglected. Research efforts have focused on the quality of the final product. However, more that half of the errors which occur during the systems development are requirement errors [18]. The correction of a post-implemented error is more than 100 times more costly to correct it during the requirement analysis [19]. Therefore, it is more effective to concentrate on the quality assurance from the conceptual models. In the present paper, we focus our interest in the usability characteristic which is largely considered as one, among other, of the most important quality characteristic. We argue that ensuring the usability of a user interface, generated according to the Cameleon framework, from the conceptual model can be an appealing way to ensure the usability of this user interface.
The conceptual models covers the abstract user interface level and the concrete one. The concrete user interface is the most affected by usability. For that reason, we opted to perform the evaluation from the concrete user interface (Fig.1).

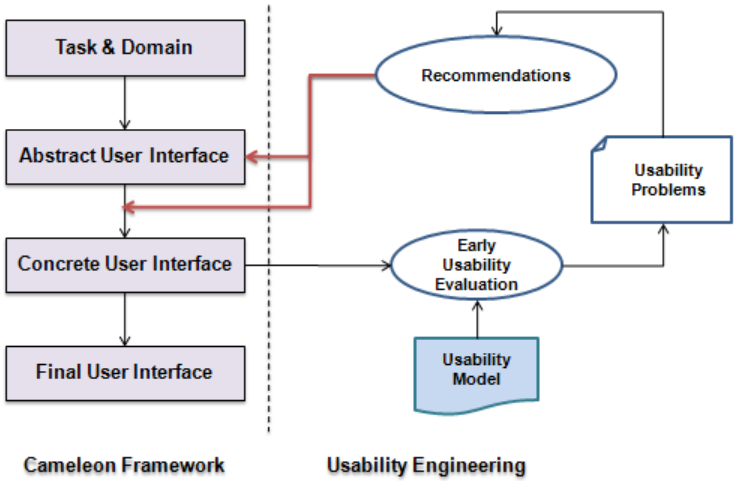

Fig. 1. Proposed Method to Early Usability Evaluation in the Cameleon Framework

Conceptual model As far as the early usability measurement method is concerned, it extends the one presented in [16]. In such proposition, the usability is evaluated from the conceptual model. The usability metrics presented in this proposition are adapted since they are specific for the OO-method. The applicability of our proposed method is shown in the present paper through Cameleon-Compliant method presented in [20]. Such method follows the MDE principles and use the BPMN [21] notion to define the user interface models. The BPMN notation is built on the Petri networks which allows the validation of the user interface models.

The extension covers three steps. The first is the add of some other usability attributes and metrics which are relevant in the context of plastic user interface (e.g., *Prompting* and *Informative Feedback*). These attributes are not only closely related to the user features, but also crucial to better guide a user with a low level of experience in interacting with computer. This explains the importance of their measurement in the context of plastic user interface. We also add the Attractiveness attributes. We argue that this sub-characteristic is crucial since it presents attributes that are related to the user preferences such as the color and the font style presented in the user interfaces. The second step is to adapt the concept of some usability metrics in order to be compatible with the underlying method. The last step is association of a priority index (weight) with each usability attribute in the grouping function. This step aims at promoting the most relevant usability properties with respect to the population characteristics and/or the task requirements.

Generally speaking, our proposed early usability measurement method is made up on four stages: 1) attributes specification, 2) metrics definition, 3) indicators definition, and 4) grouping function establishment.

3.2 Early Usability Measurement

The proposed early usability measurement method is a model-based approach. This category of usability evaluation method employs a usability model and a user interface model to generate a user interface prediction. The usability model is intended to be used in a plastic user interface development process from the conceptual models. For that reason, we take into account two key factors when we propose the usability attributes. The first one is the possibility to be measured quantitatively from the conceptual model. This allows the early usability evaluation and the automation of the evaluation process. The second is the relation between each usability attributes and the features of the context of use. This make the attribute relevant in the context of plastic user interface.

Attribute Specification. With regard to the Learnability sub-characteristic, we propose to associate the following attributes: *Prompting*, *Predictability* and *Informative Feedback*. The *Prompting* refers to the information provided to the user about the system status, possible or expected actions. The *Predictability* focuses on the available means that help the user to predict his future action. In fact, knowing the possible actions and their consequences may decrease the probability of errors. The *Informative Feedback* concerns the system's response to the user action. With respect to the user characteristics (expert, novice, etc.), learnability attributes will be considered as essential or optional in order to guarantee a high level of user satisfaction. The user without experience in similar application should be guided at all times.
The Understandability sub-characteristic can be decomposed into many attributes. The first attribute is the *Information Density* which is the user's workload from a perceptual and cognitive point of view pertaining to a set of elements. Next, *Brevity* focus on the reduction of the user's cognitive efforts (number of action steps). The short-term memory capacity is limited. Consequently, shorter entries reduce considerably the probability of making errors. Besides, *Navigability* pertains to the ease with which a user can move around in the application. Finally, *Message Concision* concerns the use of few words while keeping expressiveness in the error message. The majority of understandability attributes are related to the platform features. For example, the screen size has strong influences to the information density, the navigability and the brevity attributes.
Operability includes attributes that facilitate the user's control and operation of the system. We propose the following attributes: *User Operation Cancellability*, the possibility to cancel actions without harmful effect to the normal operation; *User Operation Undoability*, the proportion of actions that can be undone without harmful effect to the normal operation; *Explicit User Action*, the system should perform only actions requested by the user; *Error Prevention*, available means to detect and prevent data entry errors, command errors, or actions with destructive consequences. Interactive systems should allow a high level of control to users especially those with a low level of experience. Hence, the user interface is obliged to present interface components allowing such control. The screen size

of the platform being used can affect this control when it does not allow displaying button like undo, cancel, validate, etc.
The Attractiveness sub-characteristic includes the attributes of software products, which are related to the aesthetic design to make it attractive to the user. We argue that some aspects of attractiveness can be measured with regard to the *Font Style Uniformity* and *Color Uniformity*. The *Consistency* measures the maintaining of the design choice to similar contexts. The user preferences in terms of color or font style are related to the attractiveness attributes. It should be noted that some environment features (e.g. indoor/outdoor, luminosity level) affect the choice of the color in order to obtain a good contrast that gives clearer information.
Fig. 2 shows an overview of our proposal for attributes specification. The added attributes are colored red.

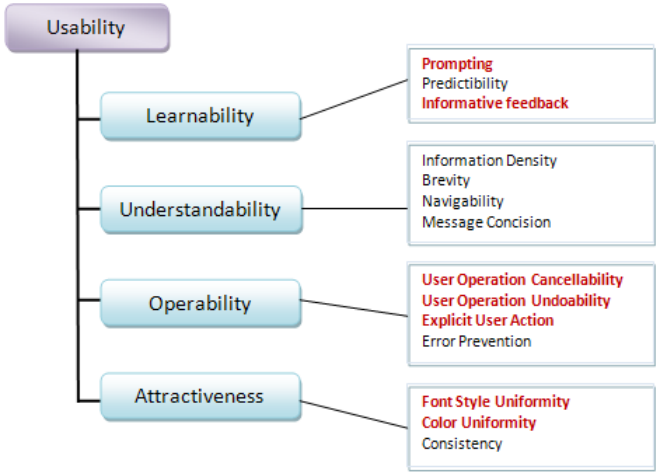

Fig. 2. Proposed Usability Model

Metric Definition. Metrics definition is crucial in order to be able to measure the usability of internal attributes. Metrics are intended to be used since the conceptual models. Therefore, they should be founded on the basis of the conceptual primitives of the underlying method. It should be noted that even though the metrics are specific to the method presented in [20], the concept of each one can be applied to any MDE method with similar conceptual primitives. As already mentioned, our proposed method is intended to evaluate the usability of plastic user interface from the conceptual models. We opted for metrics that are closely related to some context features and can be affected when these features changes. However, it should be noted that the user capacity and preferences, the screen size of the platform being used and the luminosity of the environment

are the most considered features when defining the metrics. In what follows, we list the definition of some examples of these metrics. It is recommended to look at the description of the underlying method presented in [20] in order to better understand some specific terminologies.

Prompting. One way to guide the user to enter correct data is to display the required data format when it is necessary. Hence, user interface should display as maximum as possible the supplementary information to better guide the user. We propose to calculate the average of label that display supplementary information as a metric to measure the prompting attribute (PR).

$$PR = \sum_{i=1}^{n} StaticField()/n. \tag{1}$$

StaticField() returns the number of labels (UIStaticField in the proposition of [20]) that display additional information.

Information Density. It is usually recommended to have user interface which are not too dense. The information density can be measured using the number of elements per interface to keep a good equilibrium between information and white space. We propose four metrics: the number of input elements (ID1), the number of action elements (ID2), the number of navigation element (ID3) and the total number of elements per user interface (ID4).
The average of field edit per user interface (UIWindow in [20]).

$$ID1 = \sum_{i=1}^{n} xi/\sum_{i=1}^{n} yi. \tag{2}$$

x ∈ (UIFieldEdit), y ∈ (UIWindow).
The average of action elements per user interface.

$$ID2 = \sum_{i=1}^{n} xi/\sum_{i=1}^{n} yi. \tag{3}$$

x ∈ (UIFieldAction), y ∈ (UIWindow).

Brevity. Due to the capacity of the human memory which cannot retain more than three scenarios, we propose the number of step (counted in keystrokes) required to accomplish a goal or a task from the source screen (UIWindow) to the target screen.

$$MA = distance(x, y). \tag{4}$$

x,y ∈(UIWindow), distance(x,y) returns the distance between x and y.

Navigability. The navigability can be measured by counting the number of navigation elements per user interface (Navigation Breadth).

$$NB = \sum_{i=1}^{n} xi / \sum_{i=1}^{n} yi. \tag{5}$$

x ∈ (UIFieldNavigation), y ∈ (UIWindow).

Message Concision. Since the quality of the message is a subjective measure, we propose the number of word as an internal metric to measure the quality of the message.
The number of word in a message

$$WN = \sum_{i=1}^{n} xi. \tag{6}$$

x ∈ (word in UIDialogBox).

Error Prevention. To prevent the user from error while entering data, we propose to use a conceptual primitive which represents a list (dropdown list, radio button, etc) when the input element have a set of limited possible values.

$$ERP = \sum_{i=1}^{n} list(x)/n. \tag{7}$$

x ∈ (UIFieldIn with limited possible values), *list* return the number of primitive which represents a list (UIFieldList).

Indicator Definition. The previously defined metrics provides a numerical value that needs to have a meaning in order to be interpreted. The mechanism of indicator is restored in order to reach such a goal. It consists in the attribution of qualitative values to each numerical one. Such qualitative values can be summarized in: Very Good (VG), Good (G), Medium (M), Bad (B) and Very Bad (VB). For each qualitative value, we assign a numerical range. The ranges are defined on the basis of some usability guidelines and heuristics described in the literature. Next, we detail the numeric ranges associated with some metrics in order to be considered as a Very Good value.

– Prompting: Some usability guidelines recommend the use of additional information (e.g. the required data format) in order to better guide the user during entering data [22]. At least 95% of the input element labels should display information (Prompting PR).

- Predictability: usability guidelines recommend action label that should be clear, descriptive and meaningful. 95% of action labels should be non-default labels and display supplementary information in order to increase usability [16] (Action Determination AD).
- Information Density: several usability guidelines recommend minimizing the density of a user interface [23]. We define a maximum number of elements per user interface to keep a good equilibrium between information density and white space: 15 input elements (ID1), 10 action elements (ID2), 7 navigation elements (ID3), and 20 elements in total (ID4) [16].
- Brevity: some research studies have demonstrated that the human memory has the capacity to retain a maximum number of 3 scenarios [24]. Each task or goals requiring more than 3 steps (counted in keystrokes) to be reached decreases usability (Minimal Action MA).
- Navigability: some studies have demonstrated that the first level navigational target (Navigation Breadth NB) should not exceed 7 [25].
- Message Concision: since the quality of the message can be evaluated only by the end-user, the number of the word in a message is proposed as an internal metrics to assess message quality (Word Number WN). A maximum of 15 words is recommended in a message [16].
- Error Prevention: The system must provide mechanisms to keep the user from making mistakes [22]. One way to avoid mistakes is the use of ListBoxes for enumerated values. [16] recommend at least 90% of enumerated values must be shown in a ListBox to improve usability (ERP).

It should be noted that ranges are established with two different ways. For metrics which are extracted and adapted from the proposition of [16], only the concept of metrics is adapted. We opted for the same ranges of values since they are empirically validated. This is the case of Predictability, Information Density, Error Prevention, etc. For the other metrics, the value to be considered as Very Bad is estimated taken into account the value recommended as Very Good. The discussion with some usability experts is crucial in order to benefit from their experience and help us to estimate the value to be considered as a Very Bad. After that, we equitably distribute the values for the Good, Medium and Bad categories since we have the two extremes. May be some slightly adjustments are necessary. The Table 1 shows the list of indicators that we have been defined.

Grouping Definition. The grouping function aims at putting metrics and attributes together in order to obtain a single usability measure. We adapt the grouping function proposed by [16] by the ad of weight to each element of the usability model tree. The attribution of the priority index requires usability expert and domain expert. This step aims at promoting the most relevant usability properties with respect to the population characteristics and/or the task requirements. As an outcome, the usability model will be annotated by the priority index. The model annotation is performed at the usability requirements establishment phase which precedes each evaluation.

Table 1. Proposed indicators

Metric	VG	G	M	B	VB
AD	>0.95	0.95≤AD≺0.85	0.85≤AD≺0.75	0.75≤AD≺0.65	AD≤0.65
ID1	≺15	15≤ID1≺20	20≤ID1≺25	25≤ID1≺30	ID1≥30
ID2	≺10	10≤ID2≺13	13≤ID2≺16	16≤ID2≺19	ID2≥19
ID3	≺7	7≤ID3≺10	10≤ID3≺13	13≤ID3≺16	ID3≥16
ID4	≺20	20≤ID4≺30	30≤ID4≺40	40≤ID4≺50	ID4≥50
MA	≺2	2≤MA≺4	4≤MA≺5	5≤MA≺6	MA≥6
NB	≺7	5≤NB≺10	10≥NB≺13	13≤NB≺16	NB≥16
WN	≺15	15≤WN≺20	20≤WN≺25	25≤WN≺30	WN≥30
ERP	>0.90	0.90≤ERP≺0.80	0.80≤ERP≺0.70	0.70≤ERP≺0.60	ERP≤0.60

While executing the evaluation, metrics are applied. The obtained numerical values are converted into their corresponding qualitative one. Next, each categorical value is converted to numerical values with respect to the following hypothesis. VG ⇒ 5, G ⇒ 4, M ⇒ 3, B ⇒ 2, and VB ⇒ 1. The final result is calculated with the formula presented by equation 8:

$$AVG = \sum_{i=1}^{n} pivi. \tag{8}$$

Where pi represents the weight of the item and vi represents the numerical values of the item.
The last step in the aggregation function is to convert the obtained numerical value into an ordinal value. We assign VB to the value between 1 and 1.5, B to the value between 1.6 and 2.5, M to the value between 2.6 and 3.5, G to the value between 3.6 and 4.5 and VG to value lower or equal to 5.
The three steps are applied for each grouping. Groupings are performed bottom-up from indicators until the overall user interface usability is reached.

4 An Experiment to Evaluate our Proposal

We argue that measuring the internal usability is an appealing way to predict the external usability of software product. It is recommended that internal metrics have a relationship that is as strong as possible with the external metrics. The experimentation elaborated in this Section aims at investigating the relationship between the proposed metrics and those perceived by end-user.

4.1 Objectives

With respect to the GQM template [26], the goal of the experiment is to **analyze** internal measures of usability **for the purpose** of evaluating it **with regard** to their relationship with those perceived by end-users **from the viewpoint** of the researcher **in the context** of end-users evaluating the interactive system

that is automatically generated from conceptual models.
The main research question conveyed through this study is:
RQ: is there a significant coherence between user's perception about the usability of the final product and value obtained with the proposed usability method?
We identify two hypotheses related to RQ:

- H0: There is not a significant difference between the usability obtained with our proposed method (EUE) and that perceived by the end-user (PU).
  H0: μ EUE= μ PU.
- H1: There is a significant difference between the usability obtained with our proposed method (EUE) and that perceived by the end-user (PU).
  H1: μ EUE $\neq$ μ PU.

4.2 Designing the Experimentation

Identification of variables. We identify two types of variables:

- **Response variable**: variable that corresponds to the outcome of the experimentation [27]. We identify each usability sub-characteristic as a response variable.
- **Factors**: variable that affects the response variable. We identify the Evaluation Method. This factor has two alternatives: 1) early usability evaluation without end-user, 2) usability evaluation with end-user.

Objects. The object is **rent a car system.**. It is specified using the conceptual models presented in [20] and the final system is (semi-) automatically generated from these models.

Subjects. The subjects were Thirteen undergraduate students from the Sfax National School of Engineering. Their age ranged between 27 and 35 years. Although the subjects did not have any experience in conceptual modeling, they had high level of knowledge in Human Computer Interaction. The number of the subjects is chosen according to the recommendation presented in [28].

4.3 Executing the Experimentation

Instruments. The instruments used to carry out the experiment were:

- *A demographic questionnaire*: A set of questions to know the level of experience of each user in interactive applications similar to the rent a car system.
- *Tasks*: A list of tasks that the user must carry out. The definition of tasks is intended to guarantee that all the users interact with the same contexts that are the most significant.

– *Survey*: A list of fourteen questions defined to capture the user's perception in a 5-point Likert scale format. Each question refers to one of the defined metrics to measure the usability of the internal attribute. PR and AD use the same question and the same thing for ID3 and NG. Using this survey, the user's impressions for each metrics is obtained. Hence, the usability values obtained by means of the proposed method can be compared with those perceived by the end-user. It should be noted that we need a specific question for each attribute that is why we could not use any existing questionnaire. Fig. 3 shows an example of questions from the survey.

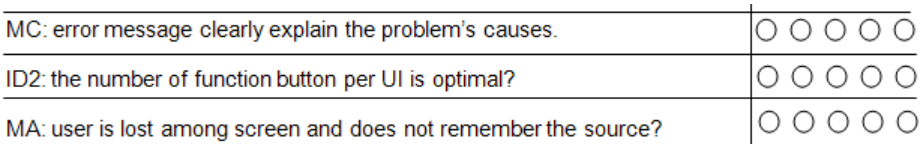

Fig. 3. Example of question from the survey

– *Spreadsheet*: It was used to accelerate the metric calculation based on the conceptual models.

4.4 Validity Evaluation

The validity evaluation is an important concept which is intended to ensure that the experimental results are valid for the target population.

Conclusion Validity. This type of validity refers to the the degree to which the conclusions made about the null hypothesis are reasonable or correct. Our evaluation was threatened by random heterogeneity of subjects. This threat appears when some users are more experienced than others. In our experiment, the experience is related to the use of interactive applications. This threat was resolved with a demographic questionnaire that allowed us to evaluate the knowledge and experience of each participant beforehand. The demographic questionnaire revealed that most users had experience with this kind of systems.

Construct Validity. One of the most basic issues in validity is the *construct validity*. It provides an answer to the following question: are we measuring what we intended to measure? We have used an inter-item correlation analysis to evaluate the construct validity of the response variable. For each item, we made use of two criteria: *Convergent validity*, which refers to the convergence among different indicators used to measure a particular construct; and the *Discriminant validity*, which refers to the divergence of indicators used to measure different constructs [29]. If the convergent validity was higher than the discriminant validity, the item is validated. The results of the validity analysis show that the

Convergent validity value was higher than the *Discriminant validity* value (see Fig.4) for each item, except for ID3, ID4 and MA.

			PR	AD	IF	ID1	ID2	ID3	ID4	MA	NB	MC	UOC	UOU	EUA	ERP	FSU	CU	CS	CV	DV	Valid
Learnability	prompting	PR	1	0.202	0	-0.12	-0.29	-0.1	1	0.2	0	-0.12	-0.29	-0.099	1	0.202	0	-0.1	-0.2855	0.40054	0.07	yes
	Predictability	AD	0.20161946	1	0.312	-0.07	0.253	0.346	0.202	1	0.312	-0.07	0.25	0.3463	0.202	1	0.31	-0.1	0.2533	0.50466	0.3	yes
	Informative Feedback	IF	0	0.312	1	0.28	-0.29	0.382	0	0.31	1	0.28	-0.29	0.3824	0	0.312	1	0.28	-0.2949	0.43745	0.24	yes
Understandability	Information Density	ID1	-0.12038585	-0.073	0.28	1	1	-0.11	-0.12	-0.1	0.28	1	-0.41	-0.107	-0.12	-0.07	0.28	1	-0.4125	0.42565	0.02	yes
		ID2	-0.28552012	0.253	-0.295	1	1	0.271	-0.29	0.25	-0.29	-0.41	1	0.2706	-0.29	0.253	-0.3	-0.4	1	0.21872	0.12	yes
		ID3	-0.03672481	0.346	0.382	-0.11	0.271	1	-0.1	0.35	0.382	-0.11	0.27	1	-0.1	0.346	0.38	-0.1	0.2706	0.24095	0.27	no
		ID4	1	0.202	0	-0.12	-0.29	-0.1	1	0.2	0	-0.12	-0.29	-0.099	1	0.202	0	-0.1	-0.2855	0.08237	0.16	no
	Brevity	MA	0.20161946	1	0.312	-0.07	0.253	0.346	0.202	1	0.312	-0.07	0.25	0.3463	0.202	1	0.31	-0.1	0.2533	0.28114	0.38	no
	Navigability	NB	0	0.312	1	0.28	-0.29	0.382	0	0.31	1	0.28	-0.29	0.3824	0	0.312	1	0.28	-0.2949	0.2799	0.27	yes
	Message Concision	MC	-0.12038585	-0.073	0.28	1	-0.41	-0.11	-0.12	-0.1	0.28	1	-0.41	-0.107	-0.12	-0.07	0.28	1	-0.4125	0.22387	0.02	yes
Operability	User operation Cancellability	UOC	-0.28552012	0.253	-0.295	-0.41	1	0.271	-0.29	0.25	-0.29	-0.41	1	0.2706	-0.29	0.253	-0.3	-0.4	1	0.30959	0.01	yes
	User Operation Undoability	UOU	-0.03672481	0.346	0.382	-0.11	0.271	1	-0.1	0.35	0.382	-0.11	0.27	1	-0.1	0.346	0.38	-0.1	0.2706	0.37956	0.22	yes
	Explicit User Action	EUA	1	0.202	0	-0.12	-0.29	-0.1	1	0.2	0	-0.12	-0.29	-0.099	1	0.202	0	-0.1	-0.2855	0.20434	0.11	yes
	Error Prevention	ERP	0.20161946	1	0.312	-0.07	0.253	0.346	0.202	1	0.312	-0.07	0.25	0.3463	0.202	1	0.31	-0.1	0.2533	0.45031	0.31	yes
Attractiveness	Font Style Uniformity	FSU	0	0.312	1	0.28	-0.29	0.382	0	0.31	1	0.28	-0.29	0.3824	0	0.312	1	0.28	-0.2949	0.32829	0.26	yes
	Color Uniformity	CU	-0.12038585	-0.073	0.28	1	-0.41	-0.11	-0.12	-0.1	0.28	1	-0.41	-0.107	-0.12	-0.07	0.28	1	-0.4125	0.28909	0.07	yes
	Consistency	CS	-0.28552012	0.253	-0.295	-0.41	1	0.271	-0.29	0.25	-0.29	-0.41	1	0.2706	-0.29	0.253	-0.3	-0.4	1	0.09755	0.01	yes

Fig. 4. Inter-Item Correlation Analysis

In order to conduct the reliability analysis on the survey, we calculate the Chronbach alpha for every question of the survey. The value obtained for the whole questionnaire was 0.966, which is a very good value for reliability. The reliability for the response variables were: 0.584 for learnability, 0.848 for understandability, 0.671 for operability, and 0.756 for the attractiveness. We argue that these values are also very good for an academic experiment.

4.5 Data Analysis

In this Sub-Section, we compare the usability perceived by end-user with the outcomes of our proposed usability evaluation method. Fig. 5 represents the comparison for the **rent a car system**.

We can state that the trend is similar for most metrics. For example, values are similar for PR, AD, ID1, ID2, ID4, MA, UOU, UOC, ERP, CU, and CS.
On the basis of the observed values, we can state that there is a relation between the values obtained from our proposal and the user's perception. The significant difference observed for some attributes can be explained by the range of values used as indicators. We may adjust some of them in order to better improve their accuracy since they are extracted from other research works that focus on other domains.
In order to study the comparison of factors in depth, we performed a statistical study called standard deviation. This test is a statistical procedure that is used to determine the mean difference between a sample and the value of a population mean. In our case, the sample was composed of the evaluation performed

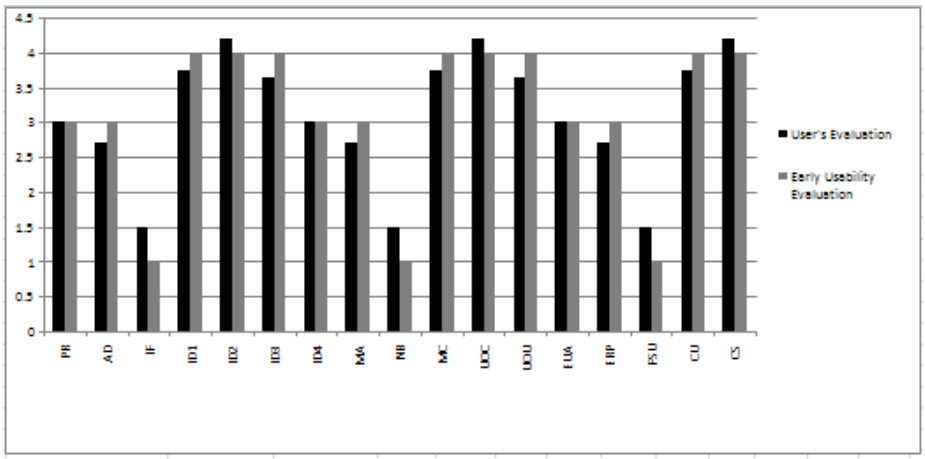

Fig. 5. Comparison of user's perception with the early usability method

with thirteen subjects (the experiment) and the population mean was the level of usability obtained from metrics and indicators (early usability evaluation). The results of the standard deviation test show that there are no significant differences between the population mean and the sample for most of metrics. Only three metrics have a significant difference (IF, NB and FSU). This can be explained by the sources from which indicators are extracted. Some adjustments are recommended in order to improve the results.
The application of the grouping function to the two methods (User's Perception UP, Early Usability Evaluation EUE) provides the results shown in Table 2.

Table 2. Usability level of the underlying system.

	Learnability	Understandability	Operability	Attractiveness
UP	M	M	M	G
EUE	M	G	M	VG

Considering the results, we can state that the null hypothesis is true for 52% of the proposed metrics. We argue that this results can be considered as encouraging as to build on it and conduct many more experiments to improve our study. The first experience with the empirical evaluation of our proposed method allows us to learn more about the potentialities of our method in the the prediction of the usability perceived by end-user. However, the accuracy of results and their acceptance rate is the main questions to be resolved in further experiments. The considerable contribution to the development costs of the final application is the main advantage of the early usability measurement method presented in this paper. The experimentation emphasizes the importance of the empirical evalua-

tion to improve our proposed measurement methods. It is not enough to provide an apparently adequate set of metrics and indicators. An empirical evaluation is usually recommended to assess the accuracy of the measurements.
However, it is essential to conduct many more experiments with other types of system and try to verify if the results will be the same. It is also crucial to more adjust the indicators estimated based on the value defined on the usability guidelines and surveys.

5 Conclusions and Future Work

In the present paper, an extension of the Cameleon framework is presented. The extension aims at integrating the usability issues during the development process of user interface where features of the context of use are taken into account. To reach this objective, we propose an early usability measurement method. The objective of this paper is to show the importance of early evaluating the usability from the conceptual model. To do this we propose a model based approach that evaluate the concrete user interface. The usability model used to perform the evaluation gather usability attributes that can be quantified using usability metrics which are based on the conceptual primitives of the underlying method. However, the concept of these metrics can be adapted in order to be used in other similar approaches.
The proposed method have two objective. The first one is to detect usability problems from the design phase which allow their correction early and with a low cost than correcting them after implementation. The second objective is to predict the usability level of the generated user interface. The present paper focus on the second objective.
To better consolidate our method, we conduct an empirical evaluation. to do that, an experiment was conducted with thirteen participant in order to investigate the relationship between the value obtained by our method and the value perceived by end-users. Results show the usefulness of the contribution in the prediction of the usability perceived by end-users. It also allows us to learn more about the potentialities and limitations of our proposed method. Results show the importance of the empirical validating a proposal rather than be justified it by logical or theoretical arguments alone.
Further research works are intended to investigate the implementation of an automatic usability evaluation process. The implementation of the usability driven model transformation process is a crucial target for further research work. The accuracy of the results requires a slightly adjustment of the indicators and to conduct many more experiments in order to validate the new proposed values.

References

1. Alain Abran, Adel Khelifi, Witold Suryn, and Ahmed Seffah. Usability meanings and interpretations in iso standards. *Software Quality Control*, 11(4):325–338, November 2003.

2. Ahmed Seffah, Mohammad Donyaee, Rex B. Kline, and Harkirat K. Padda. Usability measurement and metrics: A consolidated model. *Software Quality Control*, 14:159–178, June 2006.
3. Nigel Bevan and Miles Macleod. Usability measurement in context. *Behaviour and Information Technology*, 13:132–145, 1994.
4. Jurek Kirakowski and Mary Corbett. Sumi: the software usability measurement inventory. *British Journal of Educational Technology*, 24(3):210–212, 1993.
5. Jeffrey Rubin. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Inc., New York, NY, USA, 1994.
6. David Thevenin and Joëlle Coutaz. Plasticity of user interfaces: Framework and research agenda. In *Proc. Interact99, Edinburgh, , A. Sasse & C. Johnson Eds, IFIP IOS Press Publ*, pages 110–117, 1999.
7. Douglas C. Schmidt. Model-driven engineering. *IEEE Computer*, 39(2), February 2006.
8. Joëlle Coutaz. User interface plasticity: model driven engineering to the limit! In *EICS*, pages 1–8, 2010.
9. Gaelle Calvary, Joëlle Coutaz, and David Thevenin. A unifying reference framework for the development of plastic user interfaces. In *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, EHCI '01, pages 173–192, London, UK, UK, 2001. Springer-Verlag.
10. Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90, pages 249–256, New York, NY, USA, 1990. ACM.
11. ISO/IEC. *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC, 2001.
12. *ISO/IEC 9241. Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)*. ISO/IEC, 1998.
13. Silvia Abrahao and Emilio Insfran. Early usability evaluation in model driven architecture environments. In *Proceedings of the Sixth International Conference on Quality Software*, pages 287–294, Washington, DC, USA, 2006. IEEE Computer Society.
14. Adrian Fernandez, Emilio Insfran, and Silvia Abrahão. Integrating a usability model into model-driven web development processes. In *Proceedings of the 10th International Conference on Web Information Systems Engineering*, WISE '09, pages 497–510, Berlin, Heidelberg, 2009. Springer-Verlag.
15. Nathalie Aquino, Jean Vanderdonckt, Nelly Condori-Fernández, Óscar Dieste, and Óscar Pastor. Usability evaluation of multi-device/platform user interfaces generated by model-driven engineering. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '10, pages 30:1–30:10, New York, NY, USA, 2010. ACM.
16. José Ignacio Panach, Nelly Condori-Fernandez, Tanja E. J. Vos, Nathalie Aquino, and Francisco Valverde. Early usability measurement in model-driven development: Definition and empirical evaluation. *International Journal of Software Engineering and Knowledge Engineering*, 21(3):339–365, 2011.
17. Jaime Gómez, Cristina Cachero, and Oscar Pastor. Conceptual modeling of device-independent web applications. *IEEE MultiMedia*, 8(2):26–39, April 2001.
18. Albert Endres and Dieter Rombach. *A handbook of software and systems engineering : empirical observations, laws and theories*. The Fraunhofer IESE series on software engineering. Pearson/Addison Wesley, Harlow, England, London, Paris, 2003.
19. Barry W. Boehm. *Software Engineering Economics*. Prentice Hall, Englewood Cliffs, NJ, 1981.

20. Wided Bouchelligua, Adel Mahfoudhi, Nesrine Mezhoudi, Olfa Daassi, and Mourad Abed. User interfaces modelling of workflow information systems. In *EOMAS*, pages 143–163, 2010.
21. Bpmn: Business process modeling notation version 1.0. (2004). availabe: http://www.bpmn.org.
22. J.M. Christian Bastien and Dominique L. Scapin. Ergonomic criteria for the evaluation of human-computer interfaces. Technical Report RT-0156, INRIA, June 1993.
23. B. Shneiderman M. Leavit. *Research Based Web Design & Usability Guidelines.* 2006.
24. Maria Eugenia Lacob. *Readability and Usability Guidelines.* 2003.
25. Masaki Murata, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara. Magical number seven plus or minus two: Syntactic structure recognition in japanese and english sentences. In *Proceedings of the Second International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '01, pages 43–52, London, UK, UK, 2001. Springer-Verlag.
26. Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. The goal question metric approach. In *Encyclopedia of Software Engineering.* Wiley, 1994.
27. Juristo and Moreno. *Basics of Software Engineering Experimentation.* Kluwer Academic Publishers, Norwell, MA, USA, 2001.
28. Andreas Holzinger. Usability engineering methods for software developers. *Commun. ACM*, 48(1):71–74, January 2005.
29. D. T. Campbell and D. W. Fiske. Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin*, 56:81–105, 1959.