Early Usability Evaluation in Model Driven Framework

Lassaad Ben Ammar and Adel Mahfoudhi
University of Sfax, ENIS, CES Laboratory
Soukra km 3,5, B.P: 1173-3000
Sfax TUNISIA
benammar_lassaad@hotmail.com, adel.mahfoudhi@ceslab.org

Keywords: MDE, User Interface, Early Usability Evaluation, Metrics

Abstract: Usability evaluation play a key role in the user interfaces development process. It is a crucial part that constrains the success of an interactive application. Usability evaluation is usually conducted by end users or experts after the generation of the user interfaces. Therefore, the ability to go back and makes major changes to the design is greatly reduced.

Recently, user interfaces engineering is moving towards Model Driven Development (MDD) process. The conceptual models have become a primary artifact in the development process. Therefore, evaluating the usability from the conceptual models would be a significant advantage with regard to saving time and resources. The present paper proposes a model-based usability evaluation method which allows designers to focus on the usability engineering from the conceptual models. The evaluation can be automated taking as input the conceptual model that represent the system abstractly.

1 INTRODUCTION

The evaluation of user interfaces is a recognized problem and well explored in literature (Grislin and Kolski, 1996). Several methods, techniques, tools and criteria have been proposed to ensure the usability of the user interfaces. However, usability evaluation is usually conducted at the last stage of the development process by end users or experts. At this stage, the ability to go back and makes major changes in the design is greatly reduced.

In the last decade, the user interfaces engineering is moving towards Model Driven Development (MDD) process. In this context, the Model Driven Engineering (MDE) (Favre, 2004) proved quite appropriate. This approach tends to develop user interfaces through the definition of models and their transformations to a less abstract level to the code in the target platform. A renowned work in this context is the Cameleon project (Calvary et al., 2001). It provides a unifying reference framework for the user interface development taken into consideration the context of use. Such user interfaces are namely *multi-target*. As drawback, this framework ignore usability engineering and consider usability as a natural by-product property of whatever approach being used. Therefore, there is a need to extend the Cameleon framework in order to promotes usability as a first class entity in the development process.

The main objective of this paper is to proposes an extension of the Cameleon framework by considering the usability engineering as a part of the development process. We opted for the Cameleon framework since it presents a unifying framework for the development of multi-target user interface. The evaluation of the usability can be conducted from the conceptual models.

We structure the remainder of this paper as follows. While Section 2 presents an outline of the usability models quoted in the literature, Section 3 provides a description of our proposed usability evaluation method. A case study is presented in Section 4 in order to show the usefulness of our proposal to the uncovering of potential usability problems. Finally, Section 5 presents the conclusion and provides perspectives for future research work.

2 Related Works

Usability evaluation is often defined as methodologies for measuring the usability aspects of a user interface and identifying specific problems (Nielsen, 1993). There exist several methods targeting the usability evaluation of user interfaces. In this section, we focus our interest in the analysis of model-based methods since our main motivation is to integrate usability issues into a model driven development approach.

The usability evaluation has attracted the attention of both Human Computer Interaction (HCI) community and Software Engineering (SE) communities. The SE community proposed a quality model in the ISO/IEC 9126-1 standard (ISO, 2001). In this model, usability is decomposed into *Learnability*, *Understandability*, *Operability*, *Attractiveness* and *Compliance*. However, the HCI community has shown in the ISO/IEC 9241-11 standard (ISO, 1998) how usability can be measured in term of *Efficiency*, *Effectiveness* and *User Satisfaction*. Although both standards are useful, they are too abstract and need to be extended or adapted in order to be applied in a specific domain.

Some initiatives to extend both standards are proposed over the last few years. (Seffah et al., 2006) analyzed existing standards and surveys in order to detect their limits and complementarities. Moreover, the authors unify all these standards into a single consolidated model called Quality in Use Integrated Measurement QUIM. 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.

(Abrahão and Insfrán, 2006) proposed an extension of the ISO/IEC 9126-1 usability model. The added feature is intended to measure the user interface usability at an early stage of a model driven development process. The model contains subjective measurement which raises the question about its applicability at the intermediate artifacts. Besides, it lacks of any detail about how various attributes are measured and interpreted. An extension of this model is presented in (Fernandez et al., 2009).

The usability of a multi-platform user interface generated with an MDE approach is evaluated in (Aquino et al., 2010). The evaluation is conducted in term of effectiveness, efficiency and user satisfaction. The usability evaluation is based on the experiments with end-users. This dependency is incompatible with an early usability evaluation.

(Panach et al., 2011) 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 (Gómez et al., 2001). 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.

The analysis of the related works allows us to underline some limitations. The system implementation is always a requirement to perform the evaluation. Besides, the majority of the existing proposals lack of guidelines about how usability attributes and metrics are measured and interpreted. Regardless of the approach, none 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. Considering these limitations, it becomes clear that usability evaluation in a multi-target user interface development process is still an immature area and many more research works are needed. In order to covers this need, we propose to integrate usability issues into the Cameleon framework. The goals of our proposal are: 1) the evaluation process must be carried out quickly in the development process and independently of the system implementation, 2) the evaluation must be done in an automation way. The proposed method is intended to evaluate he usability from the conceptual model. For that reason, we propose a usability model wherein usability metrics are based on the conceptual primitives. Metrics are extracted from existing usability guidelines such as (Bastien and Scapin, 1993), (M. Leavit, 2006) and (Panach et al., 2011) with respect to the following requirements: 1) possibility to be quantified based on conceptual primitives and 2) relation with one of the context of use elements (user, platform, environment).

3 Proposed Usability Evaluation Method

3.1 Overview

The Cameleon framework provides a user interface development process which defines four essential levels of abstraction: Task & Domain, Abstract User Interface (AUI), Concrete User Interface (CUI) and Final User Interface (FUI). The development process takes as input the conceptual models in order to generate the final executable user interface. In this framework, the conceptual models covers the AUI and the CUI levels. The CUI model is the most affected by usability. Therefore, we opted to perform the evaluation from this level. To do that, we proposes a set of usability attributes which can be quantified from by means of metrics which are based on the conceptual primitives of this model. The usability evaluation module take as input the CUI model and the usability model. As outcome, it provides a set of specific usability problems. Problems are related to the conceptual primitives that are affected by it. These problems are used to suggest some recommendations in order to correct the previous stages or the transformation rules (see Fig.1).

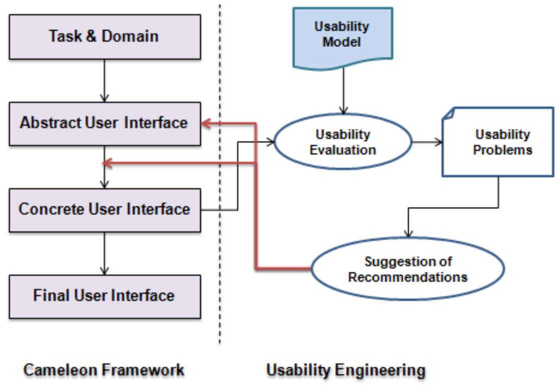


Figure 1: The proposed Usability Measurement Method.

The proposed usability model extend that presented in the ISO/IEC 9126-1 standard. In such model usability is decomposed into five sub-characteristics that are defined as follows:

- Learnability: the capability of the software product to enable the user to learn its application.
- Understandability: the capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use.
- Operability: the capability of the software product to enable the user to operate and control it.
- Attractiveness: the capability of the software product to be attractive to the user.
- Compliance: The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability.

Since the sub-characteristics have been described abstractly, we have analyzed some usability guidelines presented in the literature ((Bastien and Scapin, 1993), (M. Leavit, 2006), (Panach et al., 2011)) in order to extract and adapt more detailed usability attributes. Next Sub-Section shows our proposal to decompose the former sub-characteristics into measurable attributes.

3.2 Attribute Specification

The Learnability can be measured in terms of *Prompting*, *Predictability* and *Informative Feedback*. The *Prompting* refers to the means available to advise, orient, inform, instruct, and guide the users throughout their interactions with a computer. The *Predictability* refers to the ease with which a user can predict his future action. The *Informative Feedback* concerns the response of the system to the user action. Learnability attributes are closely related to the user characteristics. They can be considered as essential in order to guarantee a high level of user satisfaction.

In order to be able to measure the Understandability sub-characteristic, we propose four measurable attributes. The first attribute is the *Information Density* which is the degree in which the system will display/demand the information to/from the user in each interface. The *Brevity* focus on the reduction of the level of cognitive efforts of the user (number of action steps). The short term memory capacity is limited. Consequently, shorter entries reduce considerably the probability of making errors. Besides, the *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 understandability attributes are closely 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 action 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 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, 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 product that are related to the aesthetic design to make it attractive to user. We argue that some aspect of attractiveness can be measured with regard to the *Font Style Uniformity* and *Color Uniformity*. The *Consistency* measure the maintaining of the design choice to similar contexts. The user preferences in term 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 which give more clear infor-

mation.
Fig.2 shows an overview of our proposal for attributes specification.

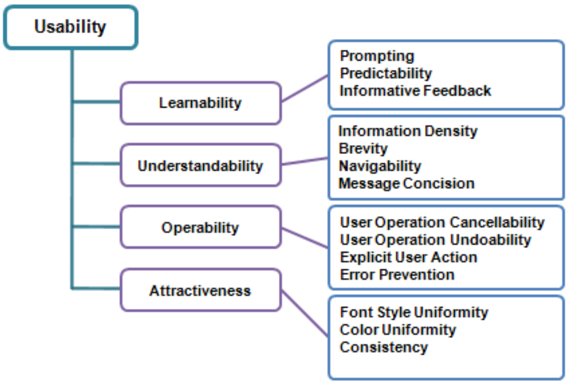


Figure 2: The Proposed Usability Model.

3.3 Metric Definition

In order to be able to measure the internal attributes proposed in the previous Section, we need to define the metrics required to measure each one. It should be noted that metrics are intended to measure the internal usability from the conceptual models that is why they are founded based on the conceptual primitives of the method presented in (Bouchelligua et al., 2010). Even though the metrics are specified to this method, the concept of each one can be applied to any MDE method with similar conceptual primitives. The main reason of the choice of the method presented in (Bouchelligua et al., 2010) is that this method is compliant to the Cameleon framework and use the BPMN notation to describe the user interface models. The BPMN notation is based on the Petri networks which allows the validation of metrics. In what follows, we list the definition of some examples of these metrics.
Information Density
The average of field edit per UI.

$$ID1 = \sum_{i=1}^{n} xi / \sum_{i=1}^{n} yi. \quad (1)$$

x ∈ (UIFieldEdit), y ∈ (UIWindow).
The maximum number of elements per UI.

$$ID2 = \sum_{i=1}^{n} xi. \quad (2)$$

x ∈ (UIField).
Brevity
We propose the number of step required to accomplish a goal or a task from a well designated context.

$$BR = distance(x,y). \quad (3)$$

x,y ∈(UIWindow), distance(x,y) returns the distance between x and y.
Navigability
The average of navigation elements per UI.

$$NB = \sum_{i=1}^{n} xi / \sum_{i=1}^{n} yi. \quad (4)$$

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

$$MC = \sum_{i=1}^{n} xi. \quad (5)$$

x ∈ (word in UIDialogBox).
Error Prevention To prevent user against error while entering data, we propose to use a drop down list instead of text field when the input element have a set of accepted values.

$$ERP = \sum_{i=1}^{n} dropdownlist(x)/n. \quad (6)$$

x ∈ (UIFieldIn with limited values), dropdownlist return the number of UIDropDownList.

3.4 Indicator Definition

The metrics defined previously provides a numerical value that need to have a meaning in order to be interpreted. The mechanism of indicator is restored in order to reach such 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 build on 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.

- Information Density: several usability guidelines recommend minimizing the density of a user interface (M. Leavit, 2006). 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) (Panach et al., 2011).
- Brevity: some research studies have demonstrated that the human memory has the capacity to retain a maximum number of 3 scenarios (Lacob, 2003).

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 (Murata et al., 2001).
- 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 (Panach et al., 2011).
- Error Prevention: The system must provide mechanisms to keep the user from making mistakes (Bastien and Scapin, 1993). One way to avoid mistakes is the use of ListBoxes for enumerated values. (Panach et al., 2011) recommend at least 90% of enumerated values must be shown in a ListBox to improve usability (ERP).

Metrics which are extracted from the proposition of (Panach et al., 2011), they are extracted with their ranges of values. in fact, this ranges are empirically validated. For the others metrics, the ranges of values to consider the numeric value as Very Good are taken into consideration in order to estimate the value to be considered as Very Bad. The Medium, Bad and Good values are equitably distributed once we have the two extremes. The Table 1 shows the list of indicators that we have been defined.

3.5 Automatic Usability Evaluation Process

Conducting the usability measurement manually is a tedious task. That is why we propose to automatize this process by implementing it as a model transformation process. The model transformation process requires two model as input (the user interface model and the usability model) and provides as outcome a usability report which contains the detected usability problem. In the model transformation literature, the definition of the meta-model[1] is a prerequisite in order to use a model.

Concerning the usability model, the proposed meta-model is composed of hierarchy with four levels:

- Sub-characteristic: A set of abstract concept used to define usability.
- Attribute: An entity which can be ensured during the model transformation process.
- Metric: A set of metric used to quantify an attribute.
- Indicator: qualitative value assigned to each set of values used to rank metric to give meaning.

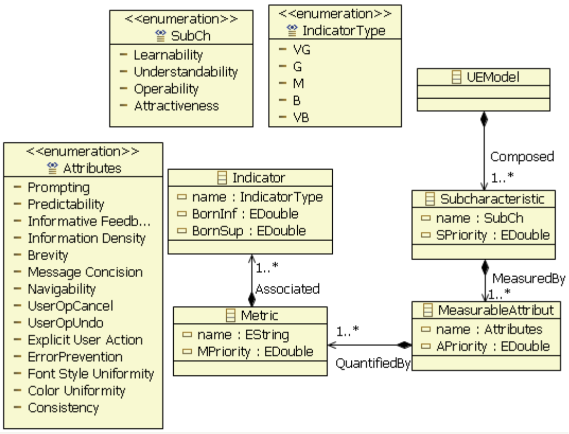


Figure 3: Usability Meta-Model.

With regard to the usability report, we propose a simple meta-model which explain the usability problem using the following scheme: the *description* of the usability problem, the *related attribute* is the sub-characteristic and attribute in the model that are affected by the usability problem, the *level* of the detected problem and the *recommendation* to solve such problem.

Fig. 4 shows the proposed usability report meta-model.

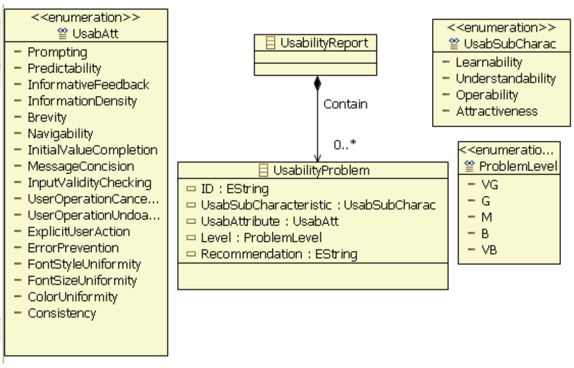


Figure 4: Usability Report Meta-Model.

It should be noted that the use of internal usability attributes and metrics which are based on the conceptual models is recommended as an appealing way to predict the usability perceived by end-users (Panach et al., 2011). However, the validity of the proposed method need to be tested empirically.

[1] A meta-model is a language that can express models. It defines the concepts and relationships between concepts required for the expression of the model.

Metric	VG	G	M	B	VB
ID1	$\prec 15$	$15 \leq ID1 \prec 20$	$20 \leq ID1 \prec 25$	$25 \leq ID1 \prec 30$	$ID1 \geq 30$
ID2	$\prec 10$	$10 \leq ID2 \prec 13$	$13 \leq ID2 \prec 16$	$16 \leq ID2 \prec 19$	$ID2 \geq 19$
MA	$\prec 2$	$2 \leq MA \prec 4$	$4 \leq MA \prec 5$	$5 \leq MA \prec 6$	$MA \geq 6$
NB	$\prec 7$	$5 \leq NB \prec 10$	$10 \geq NB \prec 13$	$13 \leq NB \prec 16$	$NB \geq 16$
WN	$\prec 15$	$15 \leq WN \prec 20$	$20 \leq WN \prec 25$	$25 \leq WN \prec 30$	$WN \geq 30$

Table 1: Proposed indicators.

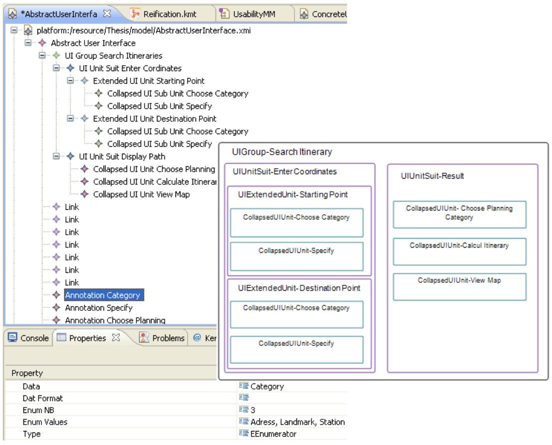
4 An Illustrative Case Study

This section investigates a case study in order to illustrate the applicability of our proposal. The purpose is to show the usefulness of our proposal in the assessment of the user interface usability. The research question addressed by this case study is: *Does the proposal contribute to uncover usability problem since the conceptual model?*

The object of the case study is a Tourist Guide System (TGS). The scenario is adapted from (Hariri, 2008). The mayor's office of a touristic town decides to provide visitors a tourist guide system. The system allows the visitors to choose the visit type (tourism, shopping, work, etc.). During the visit, the TGS offers tourists several choices of visit traverses, indicate the paths to follow and provides information about the nearby points of interests. Tourists can use the system to find places (restaurant, hotel, etc.) and know the itineraries of visits. The system will run on terminals of visitors (laptop, PDA, mobile phone, etc.). Therefore, the user interface must adapt to the context of use. For example, the computing devices being used, the tourist language, preference, etc. It should be able also to bring a feeling of comfort and ease of use in order to increase the satisfaction degree.

Since the tourist guide system is large, we focus our interests in the generation of the concrete user interface for the «Search itinerary» task. We suppose to have the abstract user interface from Fig 5 as a result of the transformation of the task model «Search itinerary» following the model transformation explained in details in (Bouchelligua et al., 2010). The result of the transformation is an XML file which is in accordance with the AUI metamodel (left part of Fig. 5). To better clear up the user interface layout, we develop an editor with the Graphical Modeling framework (GMF) of eclipse. The sketch of the user interface presented by the editor is shown in the right part of Fig. 5.

The abstract user interface contains a *UIGroup* called «Search itinerary» which gives access to two *UIUnitSuit* called «Enter Coordinates» and «Result». The «Enter Cordinates» container gives access to specify the starting point and the destination point. The tourist should choose the category (Address, Landmark, Station) before specifying the starting or the destination point. The validation of the coordinates allows tourist to choice the planning (Pedestrian, Cyclable, Vehicule, Metro, Train, Bus). After that, the TGS system shows the list of possible itineraries. The TGS system can shows the list in a map.


*AbstractUserInterfa
Reification.kmt
UsabilityMM
Concretel
platform:/resource/Thesis/model/AbstractUserInterface.xmi
Abstract User Interface
UI Group Search Itineraries
UI Unit Suit Enter Cordinates
Extended UI Unit Starting Point
Collapsed UI Sub Unit Choose Category
Collapsed UI Sub Unit Specify
Extended UI Unit Destination Point
Collapsed UI Sub Unit Choose Category
Collapsed UI Sub Unit Specify
UI Unit Suit Display Path
Collapsed UI Unit Choose Planning
Collapsed UI Unit Calculate Itinerar
Collapsed UI Unit View Map
Link
Link
Link
Link
Link
Link
Link
Annotation Category
Annotation Specify
Annotation Choose Planning
Console
Properties
Problems
Ker
UIGroup-Search Itinerary
UIUnitSuit-Enter Coordinates
UIExtendedUnit- Starting Point
CollapsedUIUnit-Choose Category
CollapsedUIUnit-Specify
UIExtendedUnit- Destination Point
CollapsedUIUnit-Choose Category
CollapsedUIUnit-Specify
UIUnitSuit-Result
CollapsedUIUnit- Choose Planning Category
CollapsedUIUnit-Calcul Itinerary
CollapsedUIUnit-View Map

Property	
Data	Category
Dat Format	
Enum NB	3
Enum Values	Adress, Landmark, Station
Type	EEnumerator

Figure 5: Abstract User Interface.

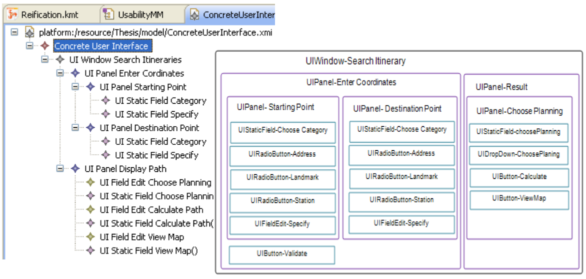
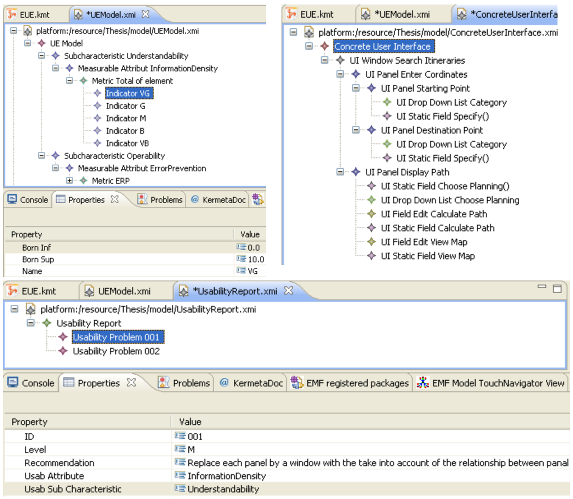


Figure 6: Concrete User Interface.

An ordinary transformation which takes as input the abstract user interface model allows producing the concrete user interface model of Fig. 6. It should be noted that this transformation was done taken into account a context of use defined by the analyst. The context is the following: a laptop as an interactive device (normal screen size), an Englishman as a tourist with a low level of experience.

In order to evaluate the concrete user interface, we pursue a reduced version of the usability evaluation process presented in (Ammar et al., 2012). The *purpose of the evaluation* is to evaluate the usefulness of the proposed model to discover the usability problems presented in the evaluated artifact. The *product part to be evaluated* is the concrete user interface model. The *selected attributes* are the Information Density and the Error Prevention. The *metrics* selected to evaluate the former attributes are ID2 and ERP. The *indicators* are those presented in Tab. 1.

The result of the evaluation is a usability report model which contains the detected problems (see Fig. 7).



```
platform:/resource/Thesis/model/UEModel.xmi
  UE Model
    Subcharacteristic Understandability
      Measurable Attribut InformationDensity
        Metric Total of element
          Indicator VG
          Indicator G
          Indicator M
          Indicator B
          Indicator VB
    Subcharacteristic Operability
      Measurable Attribut ErrorPrevention
        Metric ERP
```

Property	Value
Born Inf	0.0
Born Sup	10.0
Name	VG

```
platform:/resource/Thesis/model/ConcreteUserInterface.xmi
  Concrete User Interface
    UI Window Search Itineraries
      UI Panel Enter Cordinates
        UI Panel Starting Point
          UI Drop Down List Category
          UI Static Field Specify()
        UI Panel Destination Point
          UI Drop Down List Category
          UI Static Field Specify()
      UI Panel Display Path
        UI Static Field Choose Planning()
        UI Drop Down List Choose Planning
        UI Field Edit Calculate Path
        UI Static Field Calculate Path
        UI Field Edit View Map
        UI Static Field View Map
```

```
platform:/resource/Thesis/model/UsabilityReport.xmi
  Usability Report
    Usability Problem 001
    Usability Problem 002
```

Property	Value
ID	001
Level	M
Recommendation	Replace each panel by a window with the take into account of the relationship between panal
Usab Attribute	InformationDensity
Usab Sub Characteristic	Understandability

Figure 7: Usability Evaluation Process.

- Usability problem N1: There is no means which prevents the user against error while entering data.
  Related attribute: Operability / Error Prevention.
  Level: VB
  Recommendation: Each input element with limited values will be displayed in a dropdown list to protect user against error while entering values (e.g. typos).
- Usability problem N2: There are enough elements in the user interface which increase the information density.
  Related attribute: Understandability / Information Density
  Level:B
  Recommendation: It is recommended to replace panels with a window.

The second transformation to be conducted takes an «iPAQ Hx2490 Pocket PC» as platform. The migration to such platform raises a new redistribution of the user interface elements. The small screen size (240x320) is not sufficient to display all information. The number of the concrete component to be grouped is limited to the maximum number of concepts that can be manipulated (5 in the case of «iPAQ Hx2490 Pocket PC»). Therefore, the user interface elements are redistributed on several windows. The redistribution of interface elements on several windows will bring more steps to reach the goal. It should be noted that with a small screen size the *Information Density* and the *Brevity* are the most relevant usability attributes. The problem is that these two attributes have a contradictory impact. It is recommended to distribute the concrete components on several screens in order to obtain better *Information Density*. However, redistribute elements from one screen to several will influence negatively the *Brevity* attributes.

Learned Lesson.
The case study allow us to learn more about the potentialities and limitations of our proposal and how it can be improved. The proposed method allows the detection of several usability problems since the early stage of the development process. The evaluation process may be a means to discover which usability attributes are directly supported by the modeling primitives or to discover limitations in the expressiveness of these artifacts. The ranks of indicators are extracted from existing studies which do not consider the context variation. Therefore, many more experimentations are needed in order to propose a repository of indicators in several cases (medium screen size, small screen size, large screen size). The same things for other metrics which are influenced by the context variation. Another important aspect which must be studied is the contradictory affect of usability attributes. For example, for computing platform with small screen size the information density and the brevity has a contradictor affect. Increasing the information density will decrease certainly the brevity attribute. Finally, the case study was very useful for us. We can state that the method presented in this paper can be a building block of an MDE method that generate a user interface taken into account the context variation of use while respecting human factors.

5 Conclusion and Future research Works

This paper presents a method for integrating usability issue as a part of a plastic user interface development process. The proposed method extends the Cameleon reference framework by integrating usability issues to the development process. The early usability measurement has the objective to discover the usability problems presented in the intermediate artifact. Therefore, the present paper proposes a usability model which decomposes the usability on measurable attributes and metrics that are based on the conceptual primitives. Metrics are extracted from existing usability guidelines with respect to their relation with context features (user characteristics, platform features, etc.). Many details about how to measure and interprets attributes are presented.

If compared to the existing proposals, our framework presents three main advantages: 1) costs are very low: internal usability evaluation reduce considerably the development cost, 2) system does not have to be implemented, 3) it provides a proper details about how to measure attributes and interpret their scores.

The continuity of our research work leads directly to the implementation of the usability driven model transformation. We have to investigate the relationship between usability attributes and their contradictory influence to the whole usability of the user interface. An empirical evaluation of the early usability measurement is recommended to clearly demonstrate the coherence between values obtained by our proposal and those perceived by end-user.

REFERENCES

(1998). *ISO/IEC 9241. Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)*. ISO/IEC.

(2001). *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC.

Abrahão, S. M. and Insfrán, E. (2006). Early usability evaluation in model driven architecture environments. In *QSIC*, pages 287–294.

Ammar, L. B., Mahfoudhi, A., and Abid, M. (2012). A usability evaluation process for plastic user interface generated with an mde approach. In *Software Engineering Research and Practice*, pages 323–329. CSREA Press.

Aquino, N., Vanderdonckt, J., Condori-Fernández, N., Dieste, O., and Pastor, O. (2010). 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. ACM.

Bastien, J. C. and Scapin, D. L. (1993). Ergonomic criteria for the evaluation of human-computer interfaces. Technical Report RT-0156, INRIA.

Bouchelligua, W., Mahfoudhi, A., Mezhoudi, N., Dâassi, O., and Abed, M. (2010). User interfaces modelling of workflow information systems. In *EOMAS*, pages 143–163.

Calvary, G., Coutaz, J., and Thevenin, D. (2001). 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. Springer-Verlag.

Favre, J. M. (2004). Toward a Basic Theory to Model Driven Engineering.

Fernandez, A., Insfran, E., and Abrahão, S. (2009). 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. Springer-Verlag.

Gómez, J., Cachero, C., and Pastor, O. (2001). Conceptual modeling of device-independent web applications. *IEEE MultiMedia*, 8(2):26–39.

Grislin, M. and Kolski, C. (1996). Human-machine interface evaluation during the development of interactives systems. *TSI. Technique et science informatiques ISSN 0752-4072 CODEN TTSIDJ*, (3):265–296.

Hariri, M. (2008). *Contribution à une méthode de conception et génération d'interface homme-machine plastique*.

Lacob, M. E. (2003). *Readability and Usability Guidelines*.

M. Leavit, B. S. (2006). *Research Based Web Design & Usability Guidelines*.

Murata, M., Uchimoto, K., Ma, Q., and Isahara, H. (2001). Magical number seven plus or minus two. In *Proceedings of the Second International Conference on Computational Linguistics and Intelligent Text Processing*, pages 43–52, London, UK, UK. Springer-Verlag.

Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Panach, J. I., Condori-Fernández, N., Vos, T. E. J., Aquino, N., and Valverde, F. (2011). Early usability measurement in model-driven development: Definition and empirical evaluation. *International Journal of Software Engineering and Knowledge Engineering*, 21(3):339–365.

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