# Graphical Controls Based Environment for User Interface Evaluation

Selem Charfi, Abdelwaheb Trabelsi, Houcine Ezzedine, Christophe Kolski

# Graphical Controls Based Environment for User Interface Evaluation

Selem Charfi[1], Abdelwaheb Trabelsi[2], Houcine Ezzedine[1] and Christophe Kolski[1]
[1]UVHC, LAMIH, CNRS, UMR 8201, F-59313 Valenciennes, France
{Selem.charfi, Houcine.Ezzedine, Christophe.Kolski}@univ-valenciennes.fr
[2]University of Sfax, GIAD, 3018, Sfax, Tunisia
Abdelwaheb.Trablesi@fss.rnu.tn

**Abstract.** For more than two decades, the HCI community has elaborated numerous tools for user interface evaluation. Although the related tools are wide, the evaluation remains a difficult task. This paper presents a new approach for user interface evaluation. The proposed evaluation process focuses on utility and usability as software quality factors. It is based on the UI ergonomic quality inspection as well as the analysis and the study of the Human-Computer interaction. The proposed approach is mainly based on graphic controls dedicated to the user interface evaluation. These controls have, on the one hand, the role to compose graphically the interfaces. On the other hand, they contribute to the UI evaluation through integrated mechanisms. The evaluation is structured into two phases. The first consists of a local self-evaluation of the graphical controls according to a set of ergonomic guidelines. This set is specified by the evaluator. The second allows an electronic informer to estimate the interaction between the user interface (graphically composed by the evaluation based controls) and the user.

**Keywords:** User interface (UI), UI evaluation; utility and usability inspection.

## 1 Introduction and background

The user interface (UI) evaluation is essential for interactive systems validation and test [10] [15] [16]. It is defined as the detection of UI aspects leading to use difficulties and errors. The HCI community has proposed many tools to evaluate UI for more than two decades. Although these tools are numerous, there are some difficulties related to this theme. As follows, an attempt is made to briefly summarize some of the main shortcomings in UI evaluation.

First, tools based on ergonomic guidelines (EG) raise difficulties for their exploitation for UI evaluation. Indeed EG are generally expressed in natural language [7]. Therefore, they are independent from any context of use [16]. Thus, their exploitation and interpretation are rather difficult [7]. Second, the evaluation results are hard to analyze [4]. Indeed, the evaluator is often confronted with a huge amount of data set. Their analysis turns out a costly task in time and resources [8]. Third, the UI evaluation remains a neglected task by many designers. This negligence is essentially due to its high cost and its complexity. Another potential shortcoming is that the evaluation results can vary from a method to another one and from an evaluator to the other for the same UI [10]. Indeed, the evaluation often is based on the evaluator's quantitative judgments. Therefore, many works aim at automating the evaluation task to promote a subjective evaluation [1] [6] [15]. These shortcomings constitute motivations for the implementation of newer kind of UI evaluation tools. This paper falls within this line of work and aims at contributing by improvement of the existing tools. In this context, the proposed work consists in a framework for UI evaluation.

The remainder of this paper is organized as follows: the next section presents recent works related to UI evaluation tools. This section covers representative tools, their purposes and limits. Thus it provides the motivation for the presented work. Then, the proposed evaluation tool is presented: its general architecture, the proposed functionalities and the covered aspects of the evaluation process. The paper concludes with a summary, a brief overview of the proposed approach experimental validation and a discussion of the outcomes of the presented work.

## 2 Related work

In order to contribute to UI evaluation, many tools were proposed [6]. These tools have essentially as an objective, automating the evaluation. In addition, they aim at detecting the problems engendering use difficulties. The proposed tools diverge on the aspects they cover, their efficiency, the degree of efficiency and the UI type that they evaluate.

### 2.1 User interface evaluation tools

Many tools exist for UI evaluation [6]. These tools are mainly based on the utility, the usability and/or the accessibility as quality factors. The existing tools are so numerous that we consider it useful to classify them into categories. The first includes tools based on the usability inspection. The usability is defined as the ease of use and of learning UI system degree [8] [15]. The usability based tool focuses generally on the information display on the UI. The related tools are generally based on ergonomic guidelines and usability metrics, as in [14]. As example of usability inspection tool, we can mention MAGENTA [9]. The second set inspects the UI utility. The utility determines if the system allows the user realizing his/her task and if the system satisfies or not the needs for which it was elaborated. It corresponds to the functional capacities, the system performance and the provided technical support quality [10]. The related tools are generally based on the UI interaction study and analysis. Representative example of this class of tools is the electronic informer (EI) such as the case of EISEval [13]. The third category is based on the accessibility as a quality factor. The accessibility is defined as the UI capacity to be used and exploited by the largest possible users; as related tool we can cite EvalAccess [1].

### 2.2 Motivation of the present work

The present work is motivated essentially by the perspective that UI evaluation tools should be easier to establish and take into account the maximum of evaluation features. In addition, the provided evaluation process should be automated in order to provide better results. In [10], the authors recommend combining between several evaluation methods to get more reliable results. Then, we propose to adopt the ergonomic quality inspection and the EI for evaluation process. We intend to extend the range of functionality and the scope of existing tools. Through this environment, we intend to facilitate the evaluation process for the evaluators in order to make it more practical and easier. In fact, evaluation process is embedded into graphical controls[1]. Besides, the evaluation is established when conceiving the UI by composing it graphically.

In addition to that, our contribution major advantage is to support the evaluation process since the early stages (UI design) of the software development cycle. Indeed as expressed previously it is often neglected by software designers; more the evaluation process is too often located only at the end of software development cycle.

## 3 Graphical control based environment for user interface evaluation

### 3.1. Global overview of the proposed environment

The proposed environment is constituted mainly of two parts. The first one inspects the UI usability. It consists of the UI static display evaluation. The evaluation is elaborated while designing the UI. The graphical control inspects its conformity according to a set of ergonomic guidelines (EG). This self-evaluation process is established while adding a graphical control to the interface via drag and drop. The designer will be notified about detecting ergonomic inconstancies. The second inspects the UI utility. It analyses the interactions between the user and the interface. It is based mostly on an electronic informer (EI) for the capture and the analysis of the interaction sequences. Note that there is no mechanism implemented in the UI for the interaction capture. The UI is graphically composed by the evaluation based controls. These controls are similar to the graphical control proposed by the IDE. In fact, they ensure the same features and are exposed on the toolbar in the environments operating with the drag and drop principle, Figure 1. The prototype of the environment is developed using the “MS Visual Studio 2010”, Framework 4.0. The applications were implemented in C# language.

The proposed approach allows only identifying UI use problems. It does not correct them or identify clearly the utility problems. Then, it assists the evaluator for UI utility inspection. The problems identified depend on the EG selected for the

[1] A control is an element used for graphical user interface composition. It displays information to the user and allows him/her to interact with the functional kernel. As examples, we can cite: combo box, button and text box.

evaluation process. This process is not totally automated. It only captures and analyses data automatically. The suggestion and the improvement phases are elaborated manually.

### 3.2. Usability inspection phase

This phase inspects the UI usability. The major advantage is to provide an early evaluation. It may be 100 times more costly to proceed for UI improvement at a late stage that early one [10]. This phase is based on graphical controls that evaluate themselves according to a set of EG. The guidelines are defined into XML files. Once added to the UI (and even modified in the UI), each evaluation based control loads the associated EG into the memory and then it inspects itself according to these guidelines. The inspection is done as mentioned in figure 2 by a comparison between the control value and the recommended value stored in the EG. For each EG is associated possible design errors and corresponding recommendations. At the end of the auto-evaluation process, the control notifies the designer by a message mentioning the design errors and recommendations, Fig 2.

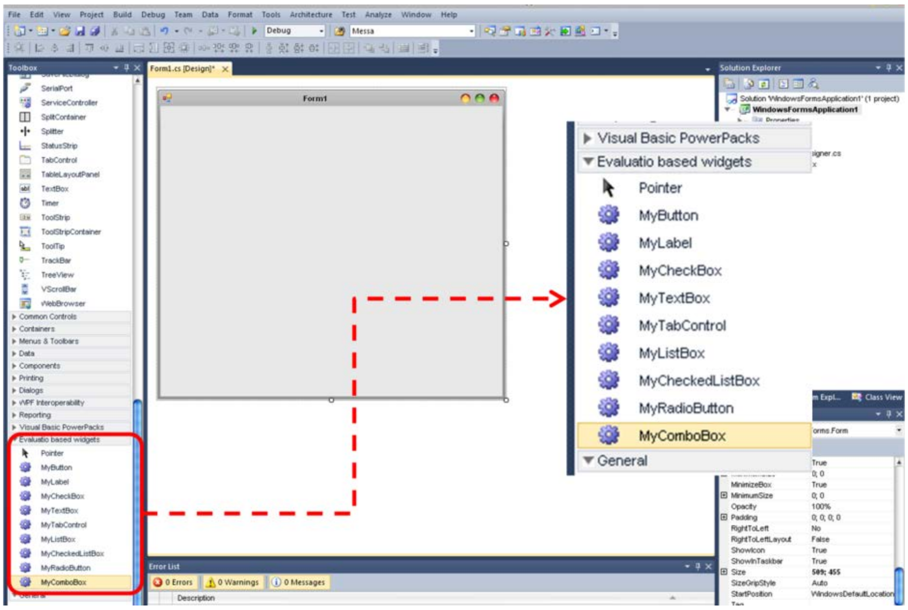


**Fig. 1.** Evaluation based graphical controls

### 3.3. Utility inspection phase

This phase is based on both an EI and evaluation based controls. The controls communicate the interaction data to the EI: elementary action execution time, action type (button click, check-list select, etc.), associated form (the interface containing the graphical control), graphical control text, control type (button, text-box, label, combo-box, etc.) and the machine IP address. The EI analyzes these data in order to detect UI utility problems. The proposed EI has a modular architecture; it is articulated around four modules: (1) Referential model generator, (2) Evaluated object model generator, (3) Confrontation and (4) Statistics generator.

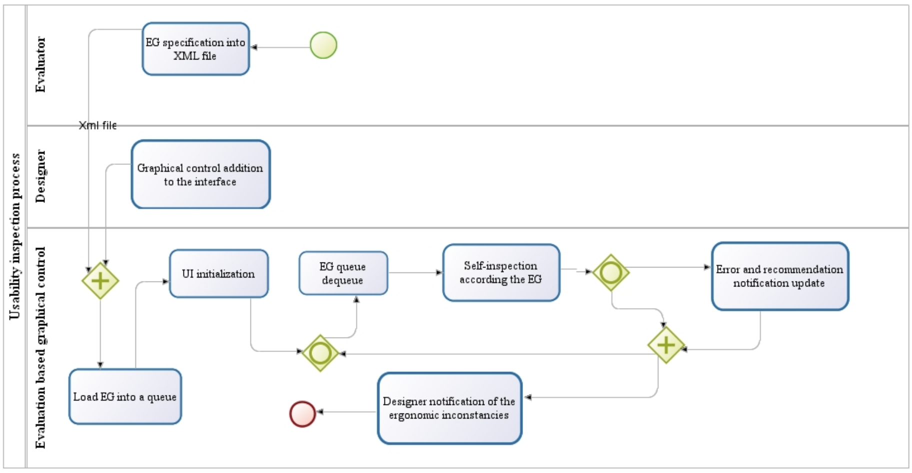


**Fig. 2.** Graphical control auto-evaluation process (modeled by BPMN notation)

The first module sets up the referential model. This model contains a description of the tasks required to be executed from the user during the test phase. The task is expressed through its sub-tasks in CTT notation [11], Figure 3. The task trees are specified by the evaluator (assisted if necessary by the UI designer). These sequences are determined by the UI designer. The designer elementary tasks are associated to the referential model (expressed through CTT notations). The second module captures the elementary actions executed by the user. This capture is done by the reception of information from the proposed evaluation based controls. Once the user ends the tasks execution, this module stores the elementary action sequences into an XML file. The interaction sequence is realized separately for each task. The third module is the EI central module. It insures the comparison between referential and evaluated object models. This comparison aims at detecting repetitive, useless and erroneous actions. Furthermore, it detects usability problems. The confrontation is based on Finite State Automaton [2]. They allow modeling the subtasks' various alternatives (often we have more than one possibility to perform a task). The CTT model is converted into a finished automaton. Then, the confrontation model inspects the elementary actions sequence (the evaluated object model) to detect inconsistencies and actions that are repetitive, useless, erroneous or missing. The fourth module generates statistics to simplify the evaluation process and to minimize the evaluator analysis and interpretation workload. These statistics concern the task execution rate, the tasks and sub-tasks execution average and the comparison between the evaluated object and the referential models through a graphic way with a colored legend, Figure 6. They cannot be described by lack of space.

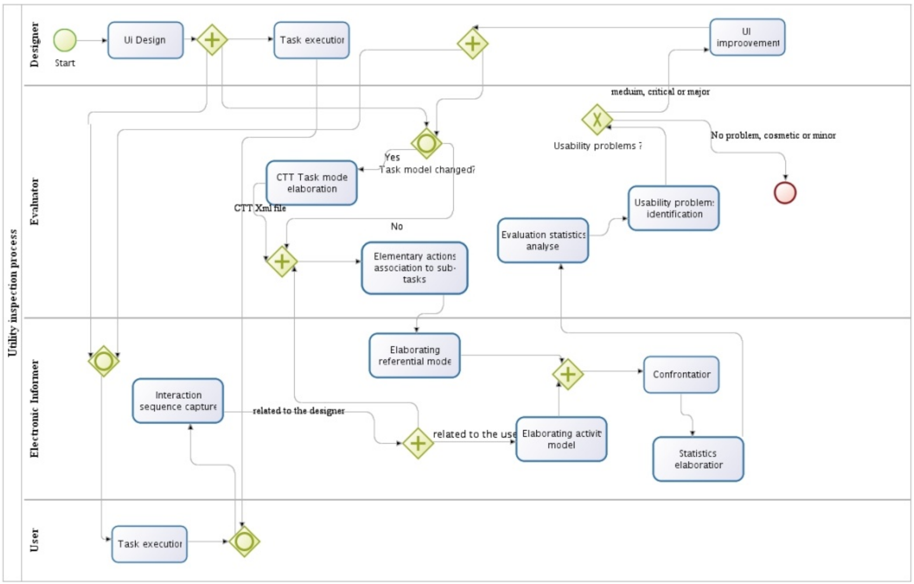


**Fig. 3.** The utility inspection process

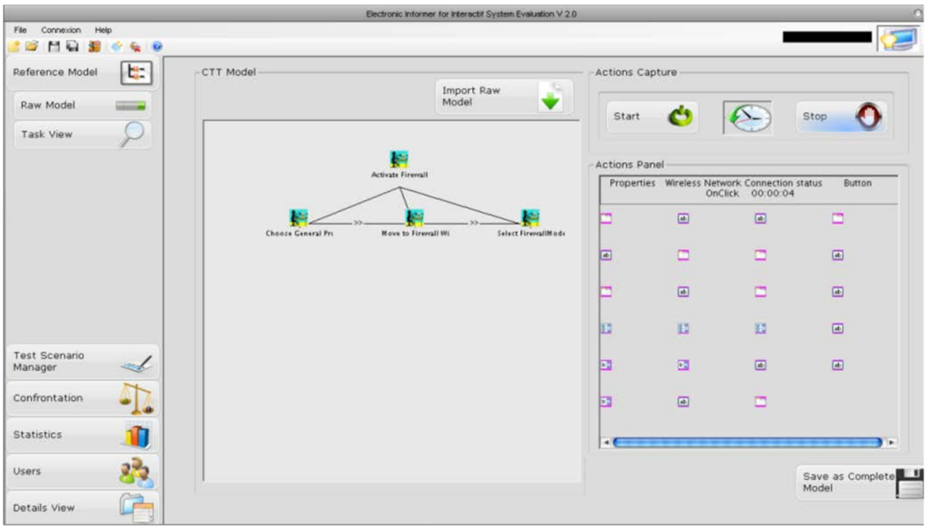


**Fig. 4.** The electronic informer for user interface evaluation

## 5 Conclusion

UI evaluation is the object of numerous researches for the last two decades. However, the evaluation remains a difficult task. Among the difficulties of the evaluation, we can underline the choice of the method and the tool to be used. In this paper we introduced several categories of UI evaluation tools. Then, we presented an approach for evaluating UI. Its goal is to contribute to obtaining useful and usable UI. It is constituted by two phases. The first one allows usability inspection by a self-evaluation process established by graphical controls. The second one inspects the UI usability through the evaluation controls and the EI. This approach aims at automating the UI evaluation during the information capture, analysis and criticism. Furthermore, it considers the HCI usability test since early interactive system design phase. The proposed approach is an attempt to combine two evaluation methods in order to contribute to UI evaluation. Different preliminary and deep evaluations may confirm the effectiveness and the easiness of the proposed approach.

In order to validate our approach, a first case study was conducted with eight novice users (undergraduate students in software engineering), one designer and one evaluator; their respective tasks are visible in Fig. 3. None of the users had deep knowledge at HCI. The study aimed at: identifying UI utility and usability problems, checking the evaluator acceptability of the proposed approach, gathering evaluators' remarks and improvement suggestions about the proposed evaluation environment, verifying that the proposed environment runs correctly. This case study concerned the evaluation of an interactive system evaluation in transport domain (the system is described in [5]). The result consists mainly on that 6 users among 8 executed successfully the required tasks. The evaluator signaled that the electronic informer is difficult to use: in fact, the guidance aspect (in the sense of Bastien & Scapin ergonomic criteria [3]) has to be improved. The inspected guidelines deal basically with the presented information clarity and readability (such as font color and size).

As research perspectives, we intend to deploy the proposed approach as service oriented architecture to provide better operability for evaluators. In addition, we intend to save evaluation result into a specific format to allow comparison between different evaluation process results to compare between prototypes alternatives.

**Acknowledgments.** The present research work is partially supported by the International Campus on Safety and Intermodality in Transportation (CISIT), the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the Ministry of Higher Education and Research, and the CNRS.

## References

1. Abascal, J., Arrue, M., Fajardo, I., Garay, N.: An expert-based usability evaluation of the EvalAccess web service. In R. Navarro-Prieto, J. Lorés (Eds.): HCI related papers of Interacción 2004, Springer, Netherlands, 1-17 (2006)
2. Anderson, J.A., Head, T.J.: Automata theory with modern applications. Cambridge University Press, ISBN 9780521848879, 105-108 (2006)
3. Bastien, J. M. C., Scapin, D. L.: Evaluating a user interface with ergonomic criteria. International Journal of Human-Computer Interaction, 7, 105-121 (1995)
4. Charfi, S., Ezzedine, H., Kolski, C., Moussa, F.: Towards an automatic analysis of interaction data for HCI evaluation, application to a transport network supervision system.

J.A. Jacko (Ed.), Human-Computer Interaction: Design and Development Approaches, HCI International 2011, Proc. LNCS 6761 (1), Springer 175-184 (2011)
5. Ezzedine, H., Bonte, T., Kolski, C., Tahon, C.: Integration of traffic management and traveller information systems: basic principles and case study in intermodal transport system management. International Journal of Computers, Communications & Control, 3, pp. 281-294 (2008)
6. Ivory, M., Hearst, M.: The State of the Art in Automated Usability Evaluation of User Interfaces, in ACM Computing Surveys, 33 (4), pp. 173-197 (2001)
7. Keith, V., Using Guidelines to assist in the visualization design process. Australian Computer Society, Inc. Asia Pacific Symposium on Information Visualization, Sydney, Australia. Conferences in Research and Practice in Information Technology, SeokHee Hong, Ed (2005)
8. Lazar, J.: Universal Usability: Designing Computer Interfaces for Diverse User Populations. Chichester, UK: John Wiley and Sons (2009)
9. Leporini, B., Paterno, F., Scorcia, A.: Flexible tool support for accessibility evaluation. Interacting with Computers, 18(5), pp. 869-890 (2006)
10. Nielsen, J.: Usability Engineering. Boston, MA: Academic Press, (1994)
11. Paterno, F., Mancini, C., Meniconi, S.: ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Proceedings Interact'97, pp. 362-369, Sydney, Chapman&Hall (1997)
12. Thatcher, J., Burks, M.R., Lauke, P.H., Heilmann, C., Kirkpatrick, A., Lawson, B., Henry, S., Regan, B., Rutter, R., Urban, M.: Web Accessibility: Web Standards and Regulatory Compliance. Friends of Ed. ISBN-13: 978-1590596388 (2006)
13. Tran, C.D., Ezzedine, H., Kolski, C.: Evaluation of Agent-based IS: Proposal of an Electronic Informer Using Petri Nets. Journal of Universal Computer Science, 14 (19), pp. 3202-3216 (2008)
14. Rauterberg, M. How to Measure the Ergonomic Quality of User Interfaces in a Task Independent Way. Advances in Occupational Ergonomics and Safety I (vol. 1; A. Mital, H. Krueger, S.Kumar, M. Menozzi & J.E. Fernandez, eds.), Cincinnati (Ohio USA): International Society for Occupational Ergonomics and Safety, 1996, pp. 154-157 (1996)
15. Rubin, J., Chisnell, D., Spool, J.: Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, 2nd Edition, Wiley, ISBN: 978-0-470-18548-388 (2008)
16. Vanderdonckt, J.: Development Milestones towards a Tool for Working with Guidelines. Interacting with Computers, 12 (2), p. 81-118 (1999)