



HAL
open science

A User-Oriented Test environment based on User-Interface evaluation graphical controls

Selem Charfi, Abdelwaheb Trabelsi, Houcine Ezzedine, Christophe Kolski

► **To cite this version:**

Selem Charfi, Abdelwaheb Trabelsi, Houcine Ezzedine, Christophe Kolski. A User-Oriented Test environment based on User-Interface evaluation graphical controls. 12th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Aug 2013, Las Vegas, United States. pp.494-504, 10.3182/20130811-5-US-2037.00060 . hal-03384691

HAL Id: hal-03384691

<https://uphf.hal.science/hal-03384691>

Submitted on 14 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A User-Oriented Test environment based on User-Interface evaluation graphical controls

Selem Charfi*, Abdelwaheb Trabelsi**, Houcine Ezzedine*, Christophe Kolski*

* LAMIH – UMR CNRS 8201, UVHC, Le Mont Houy,
59313 Valenciennes Cedex 9, France.

e-mail: {Selem.Charfi, Houcine.Ezzedine, Christophe.Kolski}@univ-valenciennes.fr

** University of Sfax, Sfax, Tunisia.

e-mail: Abdelwaheb.Trabelsi@fss.rnu.tn

Abstract: Several user-oriented techniques have been proposed for Graphical User Interface evaluation. In this paper, we focus on User-Oriented Test concepts, advantages and related problems. Among the User-Oriented Test techniques, the electronic informer is a powerful technique for interactive system testing. Thus, this research paper contributes in the improvement of existing tools for interactive system evaluation. This environment is achieved using (1) an electronic informer dedicated to the interactive system evaluation and (2) evaluation based graphical controls. The proposed evaluation environment is structured following client-server architecture. It aims to automate the interactive system evaluation process. Besides it takes the evaluation process into account since early interactive system design phases. This environment is validated by an experimental test.

Keywords: User-Oriented Test (UOT), usability test, electronic informer, HCI evaluation.

1. INTRODUCTION

Interactive systems have become ubiquitous in all areas and sectors. Thus, obtaining usable systems has become a growing challenge to which software engineering community is often challenged to. Thus, interactive system usability is a real need in the interactive systems design. Indeed, usability testing is based on performance and user experience (Stone et al., 2005; Rubin et al., 2008; Yvonne et al., 2011). To support interactive system usability, various evaluation techniques have been proposed for usability testing (Rubin et al., 2008; Stone et al., 2005; Yvonne et al., 2011). Among them, we focus in this paper on User-Oriented Test techniques. These techniques can mainly be classified following two categories. The first one includes formal tests which are performed as a simulation of the real system running (they are computed under conditions similar to those used for the system use context). The second category contains less formal tests that consist generally in an iterative testing cycles. These cycles are designed in order to detect different failures related to usability (Rubin et al., 2008).

In this paper, we will present an environment for user-oriented testing. This environment is based on an electronic informer and evaluation based graphical controls. The evaluation through the proposed environment consists of comparing actions, performed by users, to planned ones by the evaluator. In this environment, the graphical controls are used to compose user interfaces. The evaluation process through the proposed environment aims to detect the user interface usability problems. The environment, proposed in this research, represents a contribution for the techniques of User Oriented Test. It intends to detect usability problems since software development cycle early stages.

2. USER-ORIENTED TEST (UOT)

The HCI domain is rich in evaluation techniques, concepts, approaches and tools (Rubin et al., 2008; Stone et al., 2005). Among the related techniques, we find the User-Oriented tests. According to Nielsen, the UOTs are the most effective for usability testing. They enable the direct interaction between users and the interactive system. The problems of usability can be identified in real terms.

Generally, the UOT consists, at the first step, of creating a scenario (similar to those of the evaluated system use context). This scenario includes a set of tasks required for users to execute using the evaluated system. During the task execution, an observer records different interaction data either manually or automatically. The aim of such tests consists of investigating the user reaction together with the interactive system. This study enables the detection of the usability problems in the evaluated graphical user interface. Many UOT techniques exist such as “*Think aloud protocol*” (Kuusela and Pau, 2000), eye tracking (Chynal and Szymanski, 2011) and the electronic informer (Trabelsi et al., 2013). Following the usability evaluation, the UOT evaluates interactive system efficiency, effectiveness and accessibility (Yvonne et al., 2011). In addition, they measure also user performance, productivity and satisfaction (Rubin et al., 2008). UOT offers the possibility to compare different prototypes alternatives and versions. It establishes benchmarks over a longer period of time to avoid user problems, to understand competitors, and to spark guidelines for feature, user interfaces and interaction design. The benchmarks can display different goals such as relevance, efficiency, learnability, attitude, and feature availability. Ivory and Hearst identifies UOT techniques as five categories (2001):

- **Testing:** consists in observing the users interacting with the interactive system to determine usability problems;
- **Inspection:** consists in inspecting a set of criteria and aspects in order to identify the user interface (UI) usability problems;
- **Inquiry:** enables gathering the user opinion and feedback essentially through questionnaires and interviews to identify the evaluated system use problems;
- **Analytical modeling:** enables predicting usability problems via the user and the interface models' generation and,
- **Simulation:** consists in simulating the interaction between the user and the interactive system to cover the evaluated system usability errors and problems.

The UOT identifies usability problems and provides evaluation results. As the UOT are based on empirical data (Rubin et al., 2008) and needs the system user in the evaluation process, the provided results are reliable. Despite the UOT advantages, we found some drawbacks. On one hand, although the UOTs are based on the system real use simulation and representation, generally the user test does not reproduce the identical interactive system usage scenario of the interactive systems use. On the other hand, the UOT offers to the evaluator important quantitative measures. Then, these data can be difficult to be processed and to be analyzed. They can require many resources: time, cost, operators, etc. In this paper, we are particularly interested in the electronic informer as a UOT technique. The electronic informer will be the object of the following section.

3. THE ELECTRONIC INFORMER AS A USER-ORIENTED TEST TECHNIQUE

3.1. What is an electronic informer?

An electronic informer is defined by Ezzedine et al. (2008) as a technique able to ensure the automatic collection, in a real situation, of users' actions and their repercussions on the system. The collection of information is achieved in a discreet and transparent way for the user. Users of evaluated system should not be disturbed by the presence of informers. This evaluation technique is structured following two closely dependent phases. The first phase consists on interaction data acquisition between the user and the interactive system. In the second one, the electronic informer classifies and analyzes the recovered data (Ezzedine et al., 2008). The electronic informer is a useful and a powerful technique for interactive system evaluation. It gathers automatically exhaustive interaction information. These data are processed to provide an empirical evaluation. The proposed evaluation is related to quantitative measures that ensure the evaluation process result reliability. In other word, the electronic informer tracks the interaction between the user and the interface in order to detect eventual usability problems in the evaluated user interface.

3.2. Electronic Informer vs usability tracking tools

Many tools for tracking user interaction are proposed for User-Oriented Tests. Such tools capture generally the interaction sequence as a video. They are mostly focused on the mouse target during the interaction session. For instance, we can cite "*Simple Mouse Tracking*" tool (Pan et al., 2004). It stores the mouse target into a database. Thus, such tools provide the evaluator by an important amount of information as an evaluation result. These data are difficult to be processed in the evaluation process. Contrary to tracking tools, the electronic informer is based on different interaction devices information. It is not focused on the mouse target. In addition, it provides more detailed and exploitable information for the usability test. Furthermore, most of tracking tools are generally indented for Web user interface usability tests such as the case of "*clickheat*"¹.

3.3. Existing electronic informers

There are many electronic informers for interactive system evaluation. In this paper we follow an illustrative way rather than the exhaustive one. The most recent electronic informer can be illustrated as the following:

- *MESIA* (Trabelsi et al., 2013): is an electronic informer dedicated for the agent based interactive system evaluation. The strong idea behind this electronic informer is its architecture coupled with the system to evaluate. *MESIA* is structured by several informer agents resulting from the evaluated system architecture (and more particularly from the presentation layer agents). *MESIA* is based on the Petri nets (PN) for the interaction modeling between the user and the system to evaluate. It simulates the PN related to the interaction. It also ensures PN confrontation and specification. This electronic informer is specific to a transportation supervision system. *MESIA* supports only interaction abstract levels. This can lead to important data number. These data are costly to be analyzed (in terms of time and resources).
- *MultiDevice RemUsine* (Paterno et al., 2007): is based on *CTT*² task modelling to analyze interaction captured data. It is intended for mobiles interactive system's Remote evaluation. *MultiDevice RemUsine* considers the external environment different elements (the noise, the lighting, etc.). The *CTT* task model enables detecting the divergences between the actions performed by the user and those planned by the interactive system designer. *MultiDevice RemUsine* proposes, at the end the evaluation process end, a graphical report that presenting the collected information.

¹ <http://www.labsmedia.com/clickheat/index.html>

² Concur Task Trees is a notation for user task modelling. Its major advantage is its task hierarchic structure and the provided graphical syntax. It focuses on activities that users intend to perform. It is rich of temporal operators (Paterno, 2000).

- *EISEval* (Environment for Interactive System Evaluation) (Tran et al., 2013): is a generic and configurable electronic informer. It is used especially for agent based interactive systems. It captures Human-Computer interaction. Like *MESIA*, it models the interaction sequences using the PN. Using its confrontation module, it analyzes the various captured interaction information. The major disadvantage of this informer is the proposed PN as an evaluation result. They are difficult to be explored for user interface evaluation. Moreover, the configuration process requires injecting a code into the evaluated interactive system code source.

4. ISUTE: INTERACTIVE SYSTEM USER-TEST ENVIRONMENT

ISUTE is an Interactive System User-Test Environment. It is intended to detect usability problems in the evaluated user interface. It is structured following client-server architecture. The client is the evaluation based graphical user interface control. On the server part, we find an electronic informer dedicated for the evaluation of interactive systems. This environment supports the UOT at early system design phases: it supports interactive system early evaluation.

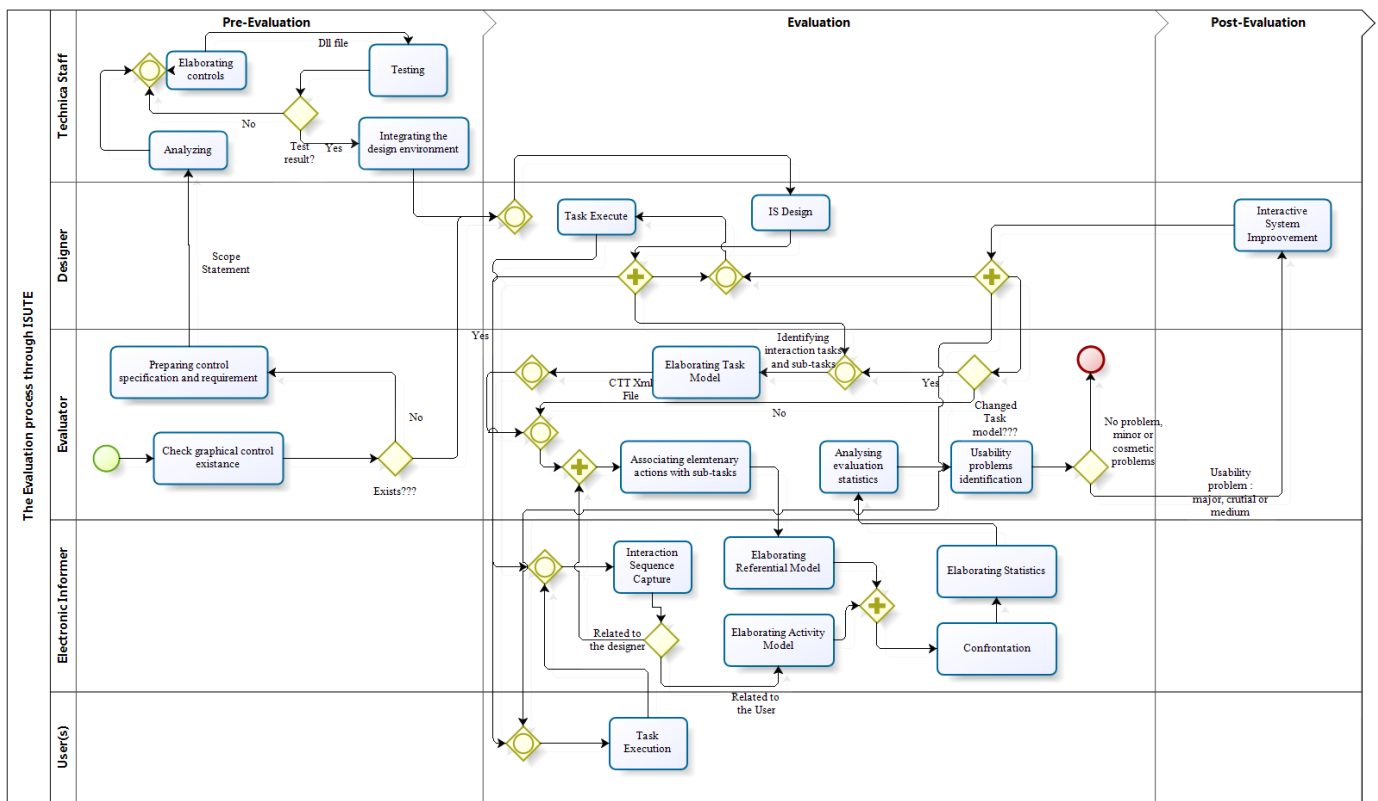


Fig. 1. The User-Oriented Test process through ISUTE (expressed through the BPMN notations).

The early evaluation consists on an explicit coupling between design and test phases (Nielsen, 1994; Tarby et al., 2008). Such evaluations reduce test cost (time and resources). Indeed, according to Nielsen, correcting final system usability problems is 100 times more costly than correcting these design errors at first design phases (Nielsen, 1994). *ISUTE* operates as follows: the graphical controls are endowed with mechanisms supporting interaction information sending through sockets via the *TCP/IP* network protocol to the informer. Using the *CTT*, the informer enables the confrontation between the executed action sequence (“the observed task model”) and the planned action sequence (“the referential model”). This confrontation aims at detecting user interface usability problems. The evaluation process requires users, designers and an evaluator. In order to model user tasks, we opt for the *CTT* (Paterno, 2000). In deep, this modeling notation structures well the interaction between the

user and the interactive system. It models interaction into tasks and sub-tasks. Besides, *CTT* proposes some operators as the one offering the choice between different action sequences (Paterno, 2000).

The evaluation process is illustrated below with the *BPMN*³ notations (Wohead et al., 2006), Fig. 1. During the pre-evaluation phase, a programmer team implements the evaluation based controls. The proposed controls (client part in the proposed evaluation environment) are similar to those proposed by development environments. Indeed, they ensure the same features and are exposed to the toolbar in the integrated development environments operating with the “*Drag and Drop*” principle, Fig. 2. There is not any specificity of usability test that is visible to the designer. The specificity of graphical component for evaluation consists on

³ Business Process Model and Notation.

informing the electronic informer (the server part in the evaluation environment) via the executed elementary task messages. These controls allow graphical user interface composition through the “WYSIWYG”⁴ environments. Once the interface is elaborated, the designer executes the possible tasks with the graphical user interface. During the interaction session, the controls send information to the informer. The evaluator establishes tasks and sub-tasks diagram using *CTT* notations.

The evaluator exploits the electronic informer to associate the various elementary actions (executed by the designer) with sub-tasks using “*drag and drop*”. Once this association is established, the electronic informer generates the user task referential model. Then, the user executes the various tasks. The evaluation based controls informs the electronic informer of the executed actions. The graphical control sends to the electronic informer:

- The execution time of elementary actions;
- The type of the executed actions (button click, checklist select, etc.);
- The associated form (the screen interface to which the graphical control is added);
- The text content of the graphical control;
- The graphical control name;
- The type of the graphical control (button, textbox, label, combo-box, etc.) and,
- The IP address of the used machine.

At the user task execution end, the confrontation module compares between the action sequence specified by the designer (the referential model) and the executed action sequence by the user (the observed task model). This confrontation generates the UOT results. This result articulates essentially around the usability problems in the evaluated interface. More details are given later (in *Result and analyzes* section).

4.1. Graphical Evaluation Based Custom Control

ISUTE proposes a set of graphical controls for the design and the test of user interfaces. These controls are inherited from graphical controls proposed by “*MS Visual Studio 2010*”. These controls are integrated into the toolbox so the designer can use these controls to compose the interactive system user interface.

The proposed controls are: combo-box, radio-button, label, button, tab control, textbox, list-box, check-box and check-box list. These controls are encapsulated into a DLL file to ensure their integration into the toolbox, Fig. 2.



Fig. 2. The custom evaluation based controls integrated into the integrated development environment Toolbox.

4.2. Electronic Informer

The proposed electronic informer has a modular architecture. It is articulated around four modules.

4.2.1. Referential Model Generator Module

This module sets up the referential model. This model contains the instructions and elementary action sequences. These sequences are determined by the interactive system designer. The reference model is established by associating the executed elementary tasks and task modeling, Fig. 3.

4.2.2. Observed Task Model Generator Module

The observed task model generator module captures of the elementary actions executed by the user. This capture is done via the reception of information from the evaluation based controls. Once the user ends the executed of the tasks, this module piles the elementary action sequences into an XML file. The interaction sequence is realized separately for each task. Besides, the electronic informer supports the same task execution by several users. This module provided result will be exploited by the confrontation module.

4.2.3. Confrontation Module

The confrontation module is the electronic informer central module. It establishes the confrontation between the referential and the observed task models. This confrontation aims to detect the repetitive, useless and erroneous user actions. Furthermore, this module detects the usability various problems. The confrontation is based on Finite State Automaton⁵ (FSA). We opt for the FSA because they model the sub-tasks' various alternatives (often we have more than one possibility to execute a task). The CTT model is converted into a finished automaton. Then, the confrontation

⁴ Notation for « *What You See Is What You Get* ». It designs programming environments that compose visually the GUI.

⁵ Automata are constituted by states and tasks. They model process, cycles and protocols. The automata are frequently used at formal language and compilation level.

module inspects the sequence of elementary actions (the observed task model) to detect inconsistencies. This inspection aims to compare the observed task model to the referential one.

4.2.4. Statistics Generator Module

The statistics generator module provides the evaluation process result. In order to simplify the UOT process and to minimize the evaluator analysis and interpretation workload, this module proposes statistics to the evaluator. These statistics concern the user that succeeded the execution of the required tasks, the tasks and sub-tasks execution average and the comparison between the observed task and referential models in a graphic way with a colored legend, Fig. 6 and Fig. 9.

5. APPLICATION TO A NETWORK TRANSPORTATION SYSTEM

ISUTE proposed electronic informer collects the interactions between the user and the evaluated interactive system. The captured data are very important as far as they enable reconstituting referential model and compare it to user models. The tests process participants are familiarized with the interactive system during five minutes before beginning the experimental tests. The evaluator has to specify the task to be realized by the users. To validate the proposed environment, we apply the proposed graphical component for evaluation to design and then evaluate a network supervision system user interface: The IAS.

5.1. The IAS (Information Assistant System)

The IAS is a system proposed within the project “SART”. It intends to assist regulators in supervision rooms for the management and the supervision of urban transport networks. It aims to ensure that regulators do their jobs optimally during normal or abnormal transport network situations (Ezzedine et al., 2008). Thus, it aims to minimize the waiting time of passengers and their continuity of displacements in multimodal networks (Tran et al., 2008). The IAS presents the different network information to passengers and regulators. It basically consists of informing supervisors among the different network vehicles position in the transportation network. In addition, it also enables the supervisor to communicate with vehicles drivers and passengers via sending messages.

Note that the IAS architecture is an agent based one. Interested readers can consult (Ezzedine et al., 2008; Trabelsi et al., 2009; Tran et al., 2008).

To proceed to the evaluation of the IAS, we propose a prototype designed with the proposed evaluation controls, Fig. 3. The used controls are: button, label, picture box, text box and Combo-Box.

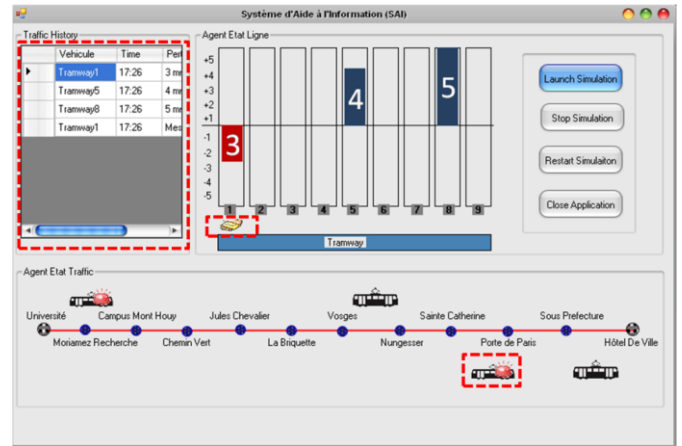


Fig. 3. A prototype of the IAS user interface

5.2. The evaluation process

The evaluation consists on an interaction session. During this session, the user is required to execute three tasks with the evaluated user interface. For every task, a referential model is defined. It is modeled through the CTT task modeling notation.

As mentioned previously, the comparison consists on comparing between referential and observed task model. Then, we proceed to interaction session to define the observed task model to proceed to the evaluation.

5.3. Interaction session

The interaction session consists on enquiring the user to manage the perturbations occurring on the network using the IAS. As mentioned previously, the proposed environment has client-server architecture, Figure 4.

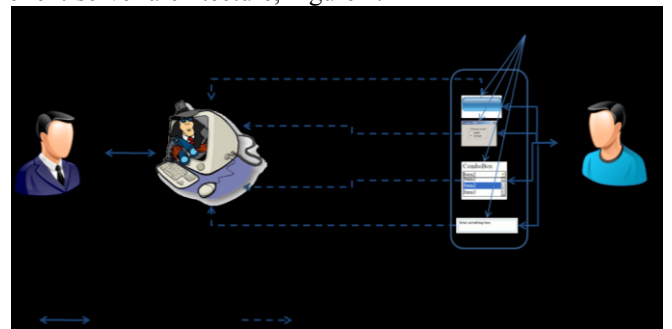


Fig. 4. Interaction session devices

In the proposed evaluation process, the evaluator interacts with the electronic informer and the user with the user interface designed with the proposed evaluation based controls. These controls communicate with the informer through the sockets to send the different user interaction information. The tasks to execute by the user are, Fig. 5:

- Send a message to the driver of the vehicle “Tramway 5” due to an advance of four minutes;
- Send a message to the driver or the vehicle “Tramway 1” due to a delay of three minutes and,

- Send a message to the driver of the vehicle "Tramway 8" due to a delay of five minutes.

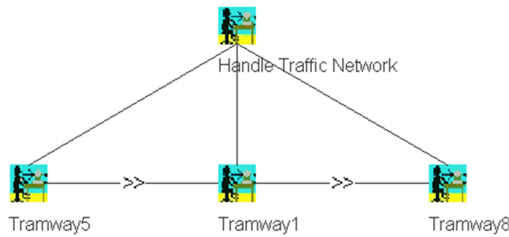


Fig. 5. Interaction session tasks to be executed by the user

Each task is decomposed of four sub-tasks, Fig. 7. The first sub-task consists on click the icon for the delay. This icon is a picture box control. This click will display an interface entitled "Agent_message", Fig. 6. It enables user to enter the message to be sent. Then, the second sub-task consists on specifying the user who will receive the message (a vehicle driver, waiting passengers or vehicle's passengers; in our case the message receiver is a vehicle driver). The message receiver selection is done through a selection using a combo box. Then, the third sub-task consists of entering the message to send in a textbox. The fourth sub-task is the validation of the message to be sent via the button ("Send").

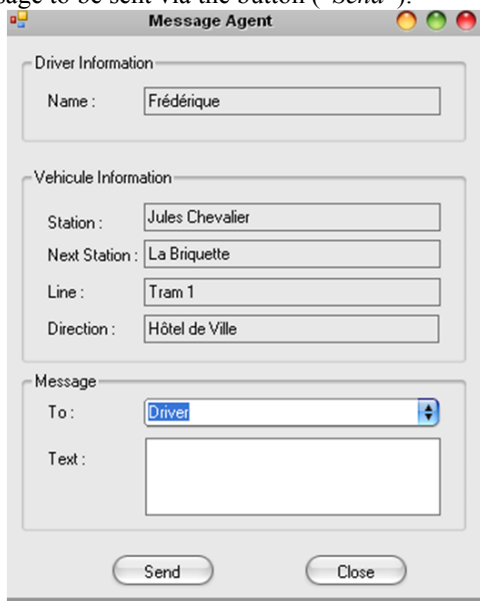


Fig. 6. A screen shot of the message sending user interface

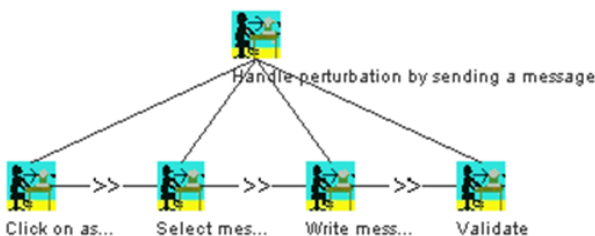


Fig. 7. The "handling of the perturbation" task modelling through CTT notations

5.4. Choice of users for interaction session

For this experimentation, we are based on the works of (Hwang and Salvendy, 2008; Whiting et al., 2008). We choose twelve users. These users are classified into two categories. The first one is constituted by six computer science student (experimented users). The second category includes the other six sport science students (novice users). Before proceeding with the interaction session, the evaluator explains to the users the experimentation context as well as to tasks that they have to execute with the interactive system.

5.5. Result and analyses

At the experimental test end, the electronic informer proposes to the evaluator a list of performed tasks by the user. The tasks are colored according to the task execution success (green: totally executed, red: unfilled, pink: delay in the execution and yellow: incomplete), Fig. 8. Furthermore, the executed task sequence is proposed to the evaluator. They contain essentially the task execution success rate.

In our experimentation, the users did not find any problems to execute the three required tasks. Indeed, the electronic informer shows 100 % success result. One of major advantages of the proposed environment is that it offers to the evaluator the erroneous manipulation and/or its locations in the action sequence.

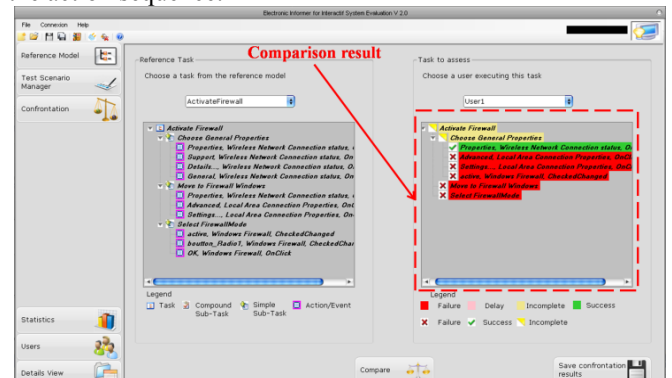


Fig. 8. A view of experimentation results

First, the evaluation showed the ease of the proposed graphical controls for the evaluation of user interface. Remember that the purpose of this evaluation is not to evaluate the IAS user interface but to test the functionality of the proposed evaluation controls. The evaluation process does not detect any usability problem when using the evaluated IAS. The users were able to easily perform the required three tasks. As an evaluation result, the electronic informer provides the evaluator by the success rate of the tasks performed by the different users and the performed actions' sequences. An evaluation of the IAS was elaborated by Tran et al. (2008). The provided evaluation by ISUTE are simpler to be exploited than those provided by EISEval.

One of the limits of the proposed electronic informer consists of the fact that the evaluator is not able to save actions performed by the user. In addition to that, it does not provide evaluation results under a form that can be exploitable for further studies and analysis. The informer cannot precisely

specify the location of usability problems and provide improvement suggestion to the evaluator to get better user interface. In addition, the proposed statistics are poorly developed and require to be worked more.

6. RELATED WORK

The proposed environment identifies the problems of usability at user interfaces. It informs mainly about usability problems frequency, impact and persistence. The detected problems are related to the following usability problem categories: minor, major and catastrophic usability problems (in the sense of usability problem rating scale of Nielsen (1994)).

Most usability evaluation tools and techniques do not consider the evaluation since the first interactive system life-cycles. Thus, evaluation is elaborated at the last stage: the test phase in the sense of the waterfall life-cycle (e.g. MESIA (Trabelsi et al., 2009), EISEval (Tran et al., 2008)). In fact, the evaluation process is more complex. One major advantage of the proposed environment is the fact that the evaluation has been considered since early stage. In fact it, the designer uses evaluation dedicated controls to compose graphically the user interface. Indeed, he/she has not to integrate code lines into the source code to capture interaction by the electronic informer or to implement specific mechanisms for the evaluation process.

In addition, most existing tools provide to the evaluator a huge amount of information. Then, it requires an important amount of time and effort to process it (Charfi et al., 2011a). In fact, the most difficult task to process while evaluating interactive system is to analyze the captured data in order to provide critics as an evaluation report to the evaluator. This difficulty is due to the lack of conformity of the information to be processed. Then, it will be hard to be exploited in the UOT process. The usability problems detection process, through ISUTE, is rather effective and easy to set up; the electronic informer proceeds of an exhaustive information collection for the evaluation process. Indeed, ISUTE covers a wide information spectre. Furthermore, the interaction data capture is made in a discreet way that does not influence the user-test context. However, ISUTE does not interpreter automatically the detected usability problems. Thus, the interpretation is done manually by the evaluator. Then, ISUTE does not propose any solution for the detected usability problems. ISUTE detects only automatically the interactive system usability problems.

In addition, the evaluators are often faced with a difficulty for evaluating an interactive system. It consists of setting up mechanism for information capture. The existing informers need to integrate code lines into the evaluated system code source to ensure data capture. Furthermore, the electronic informer setting up is generally a complex process (especially for building and configuring). For instance for MESIA, some instructions need to be injected into the interactive system source code that the electronic informer can detect the interaction information). Thus, the pre-evaluation process is complex to establish. However, ISUTE does not require code integration into the evaluated system source. It does not affect the interactive system source to evaluate.

7. CONCLUSION

This article presented an environment proposition for the interactive system User-Oriented tests. This Environment is composed of two parts. The first one is interested in the evaluation based graphical controls. The second part is an electronic informer. The proposed UOT process is automated during the information capture, and analysis. Furthermore, it considers the HCI usability test since early interactive system design phase. In this article we present this environment, entitled ISUTE. It evaluates the user interfaces basing on the interaction sequence comparison. The interaction is modeled through CTT notations. The main advantage of this environment is the fact that it does not require specific mechanism to integrate in the evaluated system source code. Comparing this finding to the existing environment, ISUTE has the specificity to incorporate the graphical user interface control to the evaluation process. The proposed environment was applied for the evaluation of a system for transport network supervision.

As research perspectives, we propose to combine these graphical controls with those proposed in (Charfi et al., 2011b). Indeed, we suggest integrating the ergonomic quality inspection in the proposed evaluation based controls to cover the maximum of the interactive system usability problems. On the other hand, we suggest improving the informer so that it can capture the elementary actions of various users simultaneously. Besides, we suggest using the task planning automatic concept (Gabilon et al., 2011). In the proposed UOT environment, the number of the proposed graphical controls for evaluation is limited to only ten graphical controls. Besides, the current Environment evaluates only the WIMP (Windows, Icons, Menu, and Pointing device) interfaces; it is not possible to evaluate WUI (Web User Interfaces). On the other hand, the provided evaluation results are not well normalized and formalized.

ACKNOWLEDGMENTS

The present research work is partially supported by the International Campus on Safety and Inter-modality 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

- Charfi, S., Ezzedine, H., Kolski, C., and Moussa, F. (2011) 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, 14th International Conference, HCI International 2011, Proc. Lecture Notes in Computer Science 6761, Part 1*, Springer, 2011, 175-184.
- Charfi, S., Trabelsi, A., Ezzedine, H., and Kolski, C. (2011) Towards ergonomic guidelines integration within graphical interface controls for the evaluation of the interactive system. *Proc. MSLT 2011, First IEEE International Conference on Mobility, Security and Logistics in Transport*, 2011, 76-82.

- Chynal, P., and Szymanski, J.M. (2011) Remote usability testing using eyetracking. In Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part I (INTERACT'11), Pedro Campos, Nuno Nunes, Nicholas Graham, Joaquim Jorge, and Philippe Palanque (Eds.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, 356-361
- Ezzedine, H., Bonte T., Kolski, C., and Tahon C. (2008) Integration of traffic management and traveler information systems: basic principles and case study in intermodal transport system management. *International Journal of Computers, Communications & Control (IJCCC)*, 3, 281-294.
- Gabillon, Y., Petit, M., Calvary, G., and Fiorino, H. (2011) Automated planning for user interface composition. In Proc. *International Workshop on Semantic Models for Adaptive Interactive Systems: SEMAIS'11 at IUI conference*.
- Hwang, W., and Salvendy, G. (2010) Number of people required for usability evaluation: the 10±2 rule. *Commun. Volume 53, no. 5, ACM*, 130-133.
- Kuusela, H., and Paul, P. (2000) A comparison of concurrent and retrospective verbal protocol analysis. *American Journal of Psychology*, volume 113(3), University of Illinois Press, 387-404.
- Ivory, Y.M., and Hearst, M.A. (2001) The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, volume 33 (4), 470-516.
- Nielsen, N. (1994) Usability Engineering, *Morgan Kaufmann*.
- Pan, B., Hembrooke, H., Gay, G., Granka, L., Feusner, M. and Newman, J. (2004) The Determinants of Web Page Viewing Behavior: An Eye Tracking Study. In *Proceedings of the 2004 symposium on Eye tracking research & applications (ETRA '04)*. ACM, New York, NY, USA, 147-154.
- Paternò, F. (2000) ConcurTaskTrees: An Engineered Notation for Task Models. In D. Diaper, Stanton, N. (Eds.), *The Handbook of Task Analysis for Human-Computer Interaction*, Lawrence Erlbaum Associates Inc, 483-501.
- Paternò, F., Russino, A., and Santoro, C. (2007) Remote evaluation of Mobile Applications. *Task Models and Diagrams for User Interface Design 6th International Workshop, TAMODIA 2007, Lecture Notes in Computer Science*, Vol. 4849, Winckler, Marco; Johnson, Hilary; Palanque, Philippe (Eds.) ISBN: 978-3-540-77221-7, 155-169.
- Rubin, J., Chisnell, D., and Spool, J. (2000) Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, 2nd Edition, *Wiley*, ISBN: 978-0-470-18548-3, 2008.
- Schaffer, E. (2004) Institutionalization of Usability: A Step-by-Step Guide. *Addison-Wesley Professional*, ISBN-13: 978-0321179340.
- Stone, D., Jarrett, C., and Woodroffe, M. (2005) User interface design and evaluation. San Francisco: *Morgan Kaufmann/Elsevier*.
- Tarby J.C., Ezzedine H., and Kolski C. (2008) Prevision of evaluation by traces during the software design of interactive systems: two approaches compared In Ahmed Seffah, Jean Vanderdonckt, and Michel Desmarais (Ed.), *Human-Centered Software Engineering: Architectures and Models-Driven Integration*, Springer HCI Series, 257-276.
- Trabelsi, A., and Ezzedine, H. (2013) Evaluation of an Information Assistance System based on an agent-based architecture in transportation domain: first results. *International Journal of Computers, Communications and Control*, 8 (2), pp. 320-333.
- Tran, C-D., Ezzedine, H., and Kolski, C. (2013) EISEval, a Generic Reconfigurable Environment for Evaluating Agent-based Interactive Systems. *International Journal of Human-Computer Studies*, 71 (6), pp. 725-761.
- Whiting, M. A., Haack, J., and Varley, C. (2008) Creating realistic, scenario-based synthetic data for test and evaluation of information analytics software. Proc. *Conference on Beyond Time and Errors: Novel Evaluation Techniques For information Visualization*, Florence, Italy, 1-9.
- Winckler, M. A., Palanque, P., and Freitas, C. M. (2004) Tasks and scenario-based evaluation of information visualization techniques. Proc. *Annual Conference on Task Models and Diagrams. TAMODIA '04*, Prague, Czech Republic, vol. 86, 165-172.
- Wohed, P., van der Aalst, W.M.P, Dumas, M. ter Hofstede, A.H.M., and Russell, N. (2006) On the Suitability of BPMN for Business Process Modelling. In S. Dustdar, J.L. Faideiro, and A. Sheth, editors, *International Conference on Business Process Management (BPM 2006)*, volume 4102 of Lecture Notes in Computer Science, Springer-Verlag, 161-176.
- Yvonne R., Sharp, H., and Preece, J. (2011) *Interaction Design: Beyond Human-Computer Interaction* (3rd ed). Chichester: *Wiley*.
- Zhang, Z., Basili, V.R., and Shneiderman B. (1999) Perspective-Based Usability Inspection: An Empirical Validation of Efficacy. *Empirical Software Engineering: An International Journal*, volume 4 (1), 43-70.