Towards ergonomic guidelines integration within graphical interface controls for the evaluation of the interactive system

Application to a transport network Information Support System

Selem Charfi[1], Abdelwaheb Trabelsi[2], Houcine Ezzedine[1], Christophe Kolski[1]

1 - Univ Lille Nord de France, F-59000 Lille, UVHC, F-59313 Valenciennes,
LAMIH, CNRS, Fre 3304, F-59313 Valenciennes, France.
charfiselem@gmail.com, {houcine.ezzedine, christophe.kolski}@univ-valenciennes.fr

2 - University of Sfax. Road of Aeroport Km 0.5 BP 1169 .3029 Sfax
GIAD, ISIG, 3018, Sfax, Tunisia
Abdelwaheb.Trabelsi@fss.rnu.tn

***Abstract*—In this paper, a new approach for interactive system evaluation is presented. It is based on the ergonomic guidelines integration within Graphical User Interface (GUI) controls. These controls provide assistance to the evaluator for the interfaces design according to a set of ergonomic guidelines. The controls use is proposed for both the evaluators and the developers. The proposed approach is based on the fact that when creating a graphical component (when adding it to the interface), it inspects a set of ergonomic criteria and verifies the GUI compliance with these criteria. The advantage of the suggested proposal consists essentially on its easiness to provide ergonomic guidelines validation when designing a GUI. To validate our approach, we propose in this paper, an application on a transport network supervision system and more particularly to an Information Assistance System for passenger.**

Keywords-component : Human-Computer Interaction, Ergonomic Guidelines, Graphical User Interface, Guidelines validation, Evaluation based controls, transport network information support system.

I. Introduction

The evaluation process is extremely important to obtain good ergonomic quality interactive system [8] [15] [23] [25] [29]. It is important in the case of failures non-tolerant interactive system such as monitoring system. Indeed, in these systems, information must be presented in the best possible way for the user to protect him from errors and wrong manipulation. In this context, evaluation methods are various. They vary according to the interactive system life cycle phase (specification, design, development, testing, validating...). They also vary according to their cost (in material resources and evaluation time) [36]. There are different quality factors over which the evaluation process can be based. The most commonly used one for the transportation system evaluation are essentially the utility and the usability [17] [22] [25] [29]. The utility determines if the system allows the user to perform its task and if he is able to achieve what is necessary to meet the user expectation from the system. It corresponds to the functional capabilities, system performance and the technical assistance quality given to the user by the system [2] [22] [27]. According to ISO 9241-18, an interactive system is said to be usable if it enables the user to perform its task with effectiveness, efficiency and satisfaction in a specified use context [13].

Moreover, each Human-Computer Interaction (HCI) evaluation method operates on a given principle. Some are based on user actions analysis such is the case of the electronic informer: MESIA [32], EMA [4], Sherlock [12], Usine [18] ... Other consist on conversing with users or with the designers in order to identify usability problems, for instance we found dedicated questionnaires, usability focused interviews and the Cognitive Walkthrough method [21] . There are others tools that are based on traces to conduct an early evaluation based on oriented use traces [9] [31]. Another category consists on validating ergonomic guidelines in the HCI: ERGOVAL [10], Sierra [35], EvalAccess [1], Destine [6]... In the following section, we will focus especially on this set of methods.

II. The interactive systems evaluation based on ergonomic guidelines validation

The Human-Computer interfaces evaluation can be implemented as ergonomic guidelines validation, in conjunction with the interactive system to evaluate. In this case, it is to ensure compliance of the user interfaces with the ergonomic guidelines and recommendations issued from the software usability [1] [6] [10]. This evaluation may take into account the static presentation, but not considering the interaction between the user and the interface (for instance: user performed actions sequence, action duration, unnecessary actions, and repetitive actions [4] [12] [33]). The advantage of such evaluation is that it can be easily implemented. Indeed, its principle is simple. It is based on Boolean and arithmetic operators in the comparison between the recommended parameters by software usability and the user interface controls attributes values [24] [30] [37]. Moreover, it does not require a large participant number. Generally the ergonomic guidelines validation tools require only the presence of an evaluator during the evaluation process.

In the ergonomic guidelines validation process, we have to dispose of all graphical controls attributes values such as writing color, graphical component dimensions, size and content... There are various methods for collecting evaluation data. Some of them parse the application source code to detect these values. For instance, the web service "A-Prompt" parses the HTML page code to verify its compliance with a set of ergonomic guidelines [3]. So the web pages evaluation seems to be very simple to evaluate but once the system to evaluate is software, the evaluation data acquisition is more difficult to perform. This is due to the following facts:

- The source code is not accessible from the application. It must be provided to the evaluator to identify the attributes of the interface.
- Each programming language codes its attributes in a very specific format, so even if it has the source code of the application, the determination of the different required data for evaluation is a tedious task. For instance, Borland C++ save the interface graphical controls attributes in a "dfm" file and Visual Studio code the controls attributes in a file called "Designer.cs".

Other tools are based on the interface descriptions established by the evaluator. These description files are realized manually. For example, the Sherlock tool is based on a client-server solution. The client machine sends the interface description to the server machine. This server contains the various ergonomic guidelines inspection algorithms. Moreover, the description generation requires a fairly important time to generate this description such as IMAGINE for Synop [16] and UIMS for Kri / AG [20].

Ergonomic guidelines validation tools in a GUI can also be differentiated according to the services that they provide to the user. Indeed, it is easier to check compliance of the interfaces with ergonomic guidelines then generating criticism or recommendations for assistance in the interactive systems evaluation. Generally the majority of existing tools report ergonomic guidelines violations related to the evaluated interfaces. But they don't offer any assistance to obtain good ergonomic quality interface: Ergoval [10], Sherlock [12], Magenta [19], Waex [7] …

Note that Ivory and Hearst distinguish tools depending on their automation level in the evaluation process. Indeed, we find the automation through three design process phases: capture, analysis and criticism. Each step may be automatic, semi-automatic or manual [14]. Most existing HCI evaluating tools provide automatic capture and analysis. But only few tools provide automatically criticism. Generally in this phase, the evaluator has to draw conclusions from the analysis performed by the evaluation system. Thus, we propose in the following section the integration of ergonomic guidelines within graphical controls.

III. Ergonomic Guidelines Integration Within GUI Controls

As mentioned previously, the interactive systems evaluation can be seen as ergonomic guidelines validation in a HCI. We propose to validate the guidelines through customized controls dedicated to the HCI evaluation. Indeed, the component evaluates itself. The proposed controls inherit from the controls supplied by the design environment. Then they are added to the toolbar to create interfaces with "drag and drop" Such an operation is possible with the environments that support WYSIWYG[1] interfaces. Note that custom controls provide the same functions and methods as those supplied with most development environments.

A. *General architecture*

As shown in Figure 1, the component, once created, is initialized on the interface. It inherits from the mother class. Then it consults the ergonomic guidelines set for his object class (button, text field, label, checkbox ...). Once the ergonomic guidelines are inspected, a window is displayed at the screen right side to illustrate the inconsistencies with the involved guidelines (see IV.B). Once the designer achieves the GUI design, he can consult a comprehensive report on the ergonomic inconsistency of the interface.

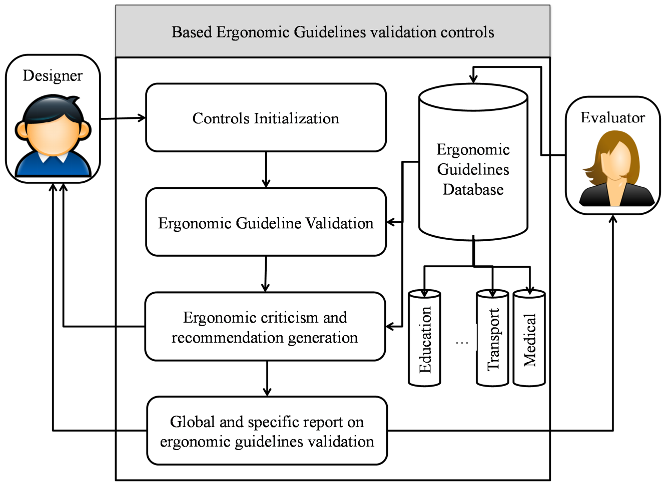


Figure 1. Based evaluation controls general architecture

The ergonomic guidelines vary depending on the inspected component. Some rules are inspected only for a set of controls. The evaluated aspects are shown later in III.A.3.

1) *Advantages of the presented approach*

The originality of this work is show in the following points:

[1] Acronym for What You See Is What You Get: visually composed interfaces. Every time the designer addsor modifies a graphical component, the result is shown immediately on the interface.

- The proposed controls are easy to use in the evaluation process. Indeed it is easy to design interfaces with available controls. The interface designer does not have to manually check the compliance of controls to the ergonomic guidelines.
- The proposed approach can be applied in the interface design at early stages: the application of this method is possible in life cycle early stages when the system is not yet fully finalized. A simple idea on the system interface is sufficient to apply this approach.
- Improvement suggestions are proposed to the evaluator and the designer in an automated way: for each rule is associated a set of ergonomic recommendations to improve the interface ergonomic quality. These recommendations are derived essentially from Vanderdonckt Ergonomic Guide [34], Bastien and Scapin ergonomic criteria [5] and the ISO norms [13].
- Assistance is provided in parallel with the interface design phase: every time the designer adds a new component or modifies one of its attributes, the component checks its compliance with the ergonomic guidelines rules and immediately notifies the designer of all non-conforming aspects and suggestions to address ergonomic inconsistencies, (see Fig.4.c). If the component is consistent with the ergonomic guidelines, a message is displayed to the evaluator indicating the absence of ergonomic inconsistency.

2) The graphical controls

The proposed graphical controls inherit from the design environment basic controls (see Fig. 2). The table 1 shows the basic controls used in the interface design. For each component a set of ergonomic guidelines is checked.

3) Ergonomic Guidelines set

The main ergonomic recommendations to be checked by the controls in a graphical interface are shown in Tab.1. These recommendations are mainly derived from [5], [13] and [34]. For instance, we cite:

- Writing_Font: font with which the text is mentioned in the component (Button name, TextBox text field content ...).
- Writing_Size: the interval that includes the writing size (Label, Button text ...).
- Writing_Color: the writing color to ensure clarity and readability while information displaying.
- Components_Dimension: the graphical controls and the image must be clearly visible, and do not occupy much space in the GUI; they should be neither too large nor too small. They must meet the GUI proportion.
- Text_Length: it's preferable to have clear and concise text fields.
- Image_Density: the interface must not be overloaded with images. Only few images and icons to facilitate the GUI use are appreciated.

TABLE I. EXAMPLES OF ASPECTS COVERED BY THE EVALUATION ORIENTED CONTROLS.

Examples of aspects covered by the evaluation oriented controls. / **Component**	**Writing_Size**	**Writing_Color**	**Writing_Font**	**Images_Dimention**	**Controls_Dimension**	**Information_Density**	**Glabal_Density**	**Background**	**Default_Value**	**Items_Number**
My_Button	X	X	X		X		X			
My_RadioButton	X	X	X				X		X	
My_CheckBox	X	X	X		X		X		X	
My_TextBox	X	X	X		X	X	X		X	
My_Label	X	X	X		X	X	X			
My_ComboBox	X	X	X				X			
My_TabControl							X			
My_ListBoxSelection	X	X	X		X	X	X		X	X
My_PictureBox				X		X	X			
My_Form	X	X	X		X			X		

- Global_Density: the GUI should not be overloaded with controls, the user can navigate easily.
- Baground_Color: The use of dark background may cause problems of legibility in the interface information.
- Cancel_Button: in each window, the user must have the possibility to return to the previous step and cancel the performed action. Thus, we must find buttons that make this possible.
- ...

Fig. 2 illustrates a snippet of the class My_Button. Note that since its creation, the component checks a set of ergonomic guidelines. This class inherits from the mother class provided by Dot Net framework (System.Windows.Forms.Button). Once the Button is created, it creates an array list to save ergonomic errors captured in the ergonomicguidelines validation phase. Then, for each guideline saved in the database, corresponds a method to inspect it. In this case, we find "Writing_size" and "Writing_font" methods that inspect the size writing and font.

At the ergonomic guideline inspection phase, if the created Arrays are empty, the control is coherent according to the inspected guidelines. If not that case, it shows al the ergonomic errors and recommendation.

```
public class My_Button : System.Windows.Forms.Button
{
        ArrayListError_List = new ArrayList ();
        struct Evaluation
        {string error; string recommendation;}
        public int density = 0;
        public Boutton(): base ()
        {
        writing_size (Errors_List, this.Controls[0].Size.ToString());
        writing_font (Errors_List, this.Controls[0].Font.ToString());
        gloabl_density(Errors_list, density);
        ...
        if (Errors_List.Count>0)
                { string errors="";
                string recommendation = "";
                for (int i=0; i<Errors_List.Count;i++)
                { errors +=Errors_List[i].error + '\n';
                recommendation+=Errors_List[i].recommendation+'\n';
                }
Evaluation_Form EF= new Evaluation_For(errors, recommendations);
EF.show();
        }
}
```

Figure 2. A snippet from the source code of an evaluation based component

4) *Provided services by based evaluation controls*

Every time a component is created or modified, it immediately notifies the designer of its compliance or not with ergonomic guidelines. If non-compliance is detected, a message is displayed to the designer to suggest corrections to obtain interface of a good ergonomic quality, Fig. 4-a, Fig 4-b.

B. *Evaluation process*

One advantage of the proposed approach is that it does not require a lot of participants. It requires only a designer for ergonomic guidelines validation process and an evaluator. The evaluator implication in the evaluation process consists on explaining and interpreting the analyses and the suggestion made by the system to the designer. The evaluation process consists of several iterative steps. First the designer or the evaluator specifies ergonomic guidelines to be validated for each component. Then, these guidelines are extracted from ergonomic guides (see III.A.). They will be inspected in the GUI. Finally, a final report of recommendations proposed by the various interface controls can be viewed.

IV. Application to Transport Network Information Support System (ISS)

A. *ISS presentation*

The ISS is a support system for information to regulators in the control room of the public transportation network to enable them to perform their tasks optimally either in normal or degraded network mode.

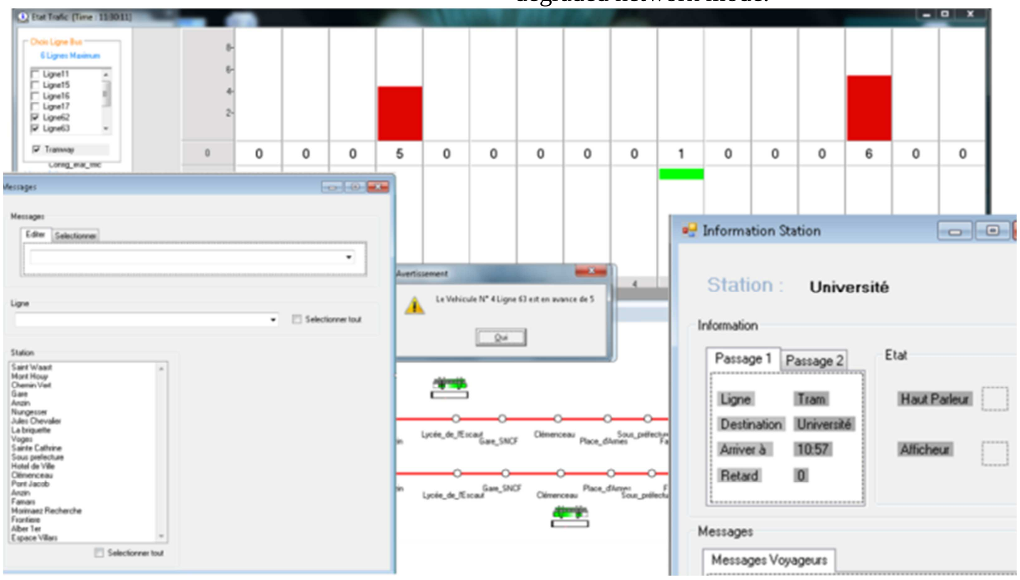


Figure 3. ISS interfaces snippets under design.

In addition, it informs passengers at stations and in vehicles (see Fig.3). The ISS looks to minimize the passengers waiting time. It also enables to ensure, wherever possible, continuity of displacements in multimodal networks. It is therefore to improve the provided service quality to passengers [26] [32] [33].

B. *Design and evaluation of the ISS with the proposed controls*

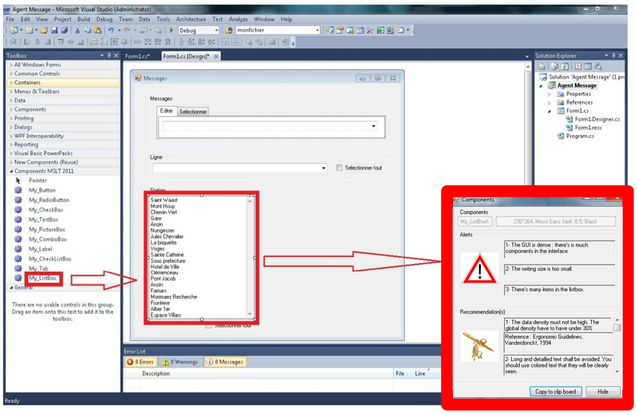
In order to validate our proposal and to ensure the ISS design, we design it with proposed controls. Every time the designer adds a component to the interface, critics on the ergonomic quality will be displayed. Once he finishes the interface design, he can view a report on non-compliance with the ergonomic guidelines and suggested improvements. Indeed, during the ISS design, the designer is notified by a set of recommendations proposed by the various interface controls.

Fig. 4 shows two recommendations samples proposed by the controls. In the first sample, it is the “Station” interface that can indicates on the network stations status, Fig. 4.a. The second sample shows the Message interface through which the user sends messages to vehicles, Fig. 4.b. Figure 4.c shows a window displayed by the proposed controls in order to mention the “Station” interface various ergonomic inconsistencies and recommendations. In this figure, we can see that the interface is too dense and that the used font is too small. In addition to that, there are too many items in the list of choices.

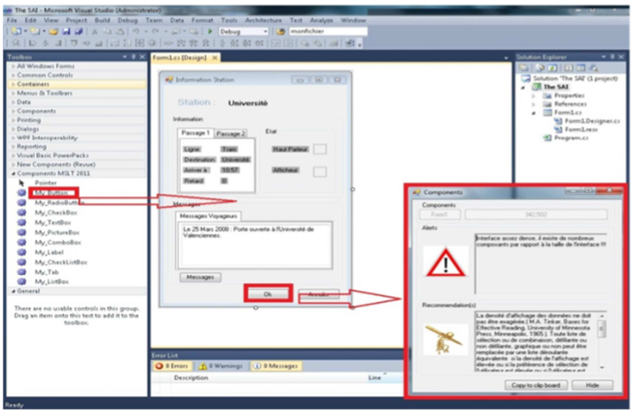
C. Results & Discussion

During the ISS design, some ergonomic guidelines were validated by the user-interface controls itself. We notice that there is a lack of incentive (software ergonomics classical criterion) in the interface. In addition to that, a lack of guidance in the information on how to use the ISS is mentioned. By agreeing to work done in [33], we find that 3 subjects understood slowly how to use the ISS. This is justified by the lack of information in the interface. The majority of the evaluated interfaces have an important information number. The overall density decrease may facilitate the ISS use by the network supervisor. We note another aspect related to the used font number. Indeed, in most cases, the designer merely choosing a single font and one color for the writing, which may present a problem for the user at differentiation between the various disclosures made by the ISS.

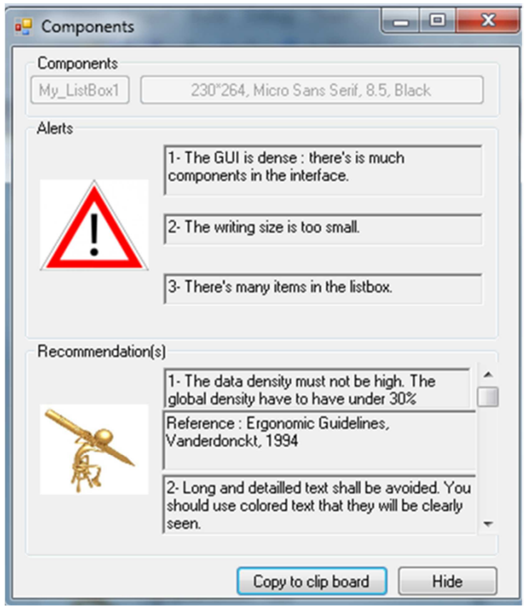
After viewing the generated final report by the proposed controls, we can recommend to the designer to reduce the number of objects in the ISS and to increase the text fields for user guidance. We also recommend adapt larger windows to make the interface less condensed.



(a) The Station interface design and evaluation



(b) Message interface design and evaluation



(c) Examples of proposed ergonomic recommendations when adding the MyListBox1 controls

Figure 4. Use examples of using custom controls for an interactive system evaluation

V. Conclusion & Perspectives

We have briefly presented the interactive system ergonomic quality evaluation. Then we presented our proposal for the validation of ergonomic guidelines in graphical interfaces. This proposal is easy to apply. In addition, the evaluation process is fully automated during the 3 steps: capture, analysis and criticism. Our proposal is built on graphical controls that inspect their own validity in relation to ergonomic guidelines. This proposal was technically validated by an experimental evaluation of a passenger Information Support System. This evaluation allowed us to detect ergonomic inconsistencies in the system.

As research perspective, we propose to make ergonomic guidelines external of the controls in order to integrate easily new guidelines without modifying the proposed controls code. For this we propose to develop an ergonomic guidelines modeling tool by using an ergonomic guidelines definition language. We also propose to implement controls to evaluate web applications. We also aim in future work to integrate other controls offered by graphics development environments. In the other hand, we propose to couple this approach with other approaches that incorporate HCI dynamic evaluation for more detailed results.

ACKNOWLEDGEMENT

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. Arue, M. Farjado, I, Garay, N. (2006). An expert-based usability evaluation of the EvalAccess web service. In R. Navarro-Prieto, J. Lorés (Eds.): HCI related papers of Interacción 2004, Springer, 1-17.

[2] Abran A., Khlifi, A.Suryn, W (2003). Usability Meanings and Interpretations in ISO Standards. Software Quality Journal 11(4), 325-338.

[3] ATRC. (s.d.). A-Prompt, Web Accessibility Verifier. Consulté le November 2010, sur http://www.aprompt.ca/

[4] Balbo, S. (1995) Task models and automation of the Evaluation of User Interface Usability", in Proceedings of OZCHI'95 conference, Wollongong (Australia).

[5] Bastien, J. Scapin, B. (1993). Ergonomic Criteria for the Evaluation of Human-Computer interfaces. Institut National de recherche en informatique et en automatique, France.

[6] Beirekdar, A. Vanderdonckt, J Noirhomme-Fraiture, M (2004) Automated Evaluation of Web Usability and Accessibility by Guideline Review. Proc. of International Conference onWebEngineeringICWE04, Lecture Notes in Computer Science, Munich, July28-30.

[7] Centeno, V.L. (2006). WAEX: Web Accessibility Evaluator in a Single XSLT File. Automated Specification and Verification of Web Systems. WWV '06. 2nd International Workshop, .

[8] Cooper, A. Reimann, R. Cronin, D. (2007). About face: The essentials of interface design (3rd ed.). New York: Wiley.

[9] Delannay, G. A generic traceability tool, available at http://www.info.fundp.ac.be/~pth/fundpdocs/gde.pdf, 2003.

[10] Farenc, C. Palanque, P. Bastien, C. Winckler, M. (2001). Towards a General Guidance and Support Tool for Usability Optimization. International Conference on Universal Access in Human-Computer UAHCI 2001, New Orleans, USA, Lawrence Erlbaum Associates, .

[11] Goodwin K., Cooper. A. (March 2009). Designing for the Digital Age: How to Create Human-Centered Products and Services. Wiley.

[12] Grammenos D., Akoumianakis, D. Stephanidis, C. (2000). Sherlock: A Tool Towards Computer-Aided Usability Inspection. In J. Vanderdonckt & C. Farenc (Eds), Proceedings of the Scientific Workshop on "Tools for Working with Guidelines", Biarritz.

[13] ISO. (2003). ISO/IEC 9126-1: Software engineering, Product quality, Part 1, 2, 3 and 4, (JIS X 0129).

[14] Ivory, M. Hearst, M. (2001). The State of the Art in Automated Usability Evaluation of User Interfaces. ACM Computing Surveys, 33(4), 173-197.

[15] Jarrett, C. Gaffney, G. (2009). Forms that work: Designing Web forms for usability. San Francisco: Morgan Kaufmann/Elsevier.

[16] Kolski, C. Millot, P. (1991). A rule-based approach to the ergonomic "static" evaluation of man-machine graphic interface in industrial processes. International Journal of Man-Machine Studies, 35, pp. 657-674.

[17] Lazar, J. (2009). Universal Usability: Designing Computer Interfaces for Diverse User Populations. Chichester, UK: John Wiley and Sons.

[18] Lecerof, A. Paterno, F. (1998). Automatic support for usability evaluation. IEEE Trans. on Software Engineering. 24 (10).

[19] Leporini, B. Paterno, F. Scorcia, A. (2006). Flexible tool support for accessibility evaluation. Interacting with Computers, vol. 18, issue 5, 869-890.

[20] Löwgren, J. Nordqvist, T. (1992). Knowledge-based evaluation as design support for graphical user interfaces. Proceedings of the Conference on Human Factors in Computing Systems, CHI'92, New York, NY, USA, ACM Pres , 181-188.

[21] Mahatody, T. Sagar, M. Kolski, C. (2010). State of the Art on the Cognitive Walkthrough method, its variants and evolutions. International Journal of Human-Computer Interaction, 26, pp. 741-785.

[22] Nielsen, J. (1994). Usability Engineering. Cambridge: AP Professional.

[23] Preece, J. Rogers, Y. Sharp, H. (2002). Interaction design: beyond human-computer interaction. ISBN 0-471-49278-7. John Wiley & Sons, Inc.

[24] Ricca, F. Tonelle, P. (2000). Web Site Analysis: Structure and Evolution. Proc. 16th IEEE Int. Conf. on Software Maintenance (ICSM'00), 76-86.

[25] Rubin, J. Chisnell, D Spool, J. (New York). Handbook of usability testing: How to plan, design, and conduct effective tests (2nd ed.). 2009: Wiley.

[26] SART. (2007). Système d'Aide à la Régulation de Trafic du réseau de transport valenciennois et de ses pôles d'échanges. Rapport final. France: Inrets.

[27] Seffah A., Donyaee, M. Kilne R.B. Padda, H.K (June 2006). Usability measurement and metrics: A consolidated model. Software Quality Journal, 14 (2).

[28] Sharp, H. (January 2007). Interaction Design, 2nd Edition. Wiley.

[29] Stone, D. Jarrett, C. Woodroffe, M. (2005). User interface design and evaluation. San Francisco: Morgan Kaufmann/Elsevier.

[30] Thatcher, J. Burks, M.R. Heilmann, C. Henry, S.L, Kirkpatrick, A. Lauke, P.H, Lawson, B., Regan, B, Rutter, R. Urban, M. Waddel, C. (2006). Web Accessibility: Web Standards and Regulatory Compliance. Friends of Ed. ISBN 978-1590-59638-8.

[31] Tarby J-C., Ezzedine H., Tran C-D., Laporte P., Kolski C. (2007). Traces using aspect oriented programming and interactive agent-based architecture for early usability evaluation: basic principles and comparison. J. Jacko (Ed.), Human-Computer Interaction, Part I, HCII 2007, Lecture Notes in Computer Science (LNCS), Springer-Verlag, pp. 632-641.

[32] Trabelsi, A. Ezzedine, H. Kolski, C (2004). Architecture modelling and evaluation of agent-based interactive systems. W. Thissen, P. Wieringa, M. Pantic, M. Ludema (Ed.), Proceedings IEEE SMC 2004, International Conference on Systems, Man and Cybernetics, IEEE, The Hague, pp. 5159-5164, octobre, ISBN 0-7803-8567-5

[33] Tran, C.D. Ezzedine, H. Kolski, C. (2008). Evaluation of agent-based interactive systems: proposal of an electronic informer using Petri Nets. Journal of Universal Computer Science, 14, pp. 3202-3216.

[34] Vanderdonckt, J. (1994). Guide ergonomique des interfaces homme-machine. Namur: Facultés universitaires Notre-Dame de la Paix à Namur (Belgique), Presses Universitaires de Namur.

[35] Vanderdonckt, J. (June 1995). Accessing Guide-lines Information with SIERRA. Proc. of 5th IFIP Conf. on Human-Computer Interaction INTERACT'95, Lillehammer, Norway, 311-316.

[36] Wilbert, O. Gallitz, A. (April 2007). The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques. ISBN: 0470053429. Wiley; 3rd edition.

[37] Xiong, J. Farenc, C. Winckler, M.A. (2007) Support for Inspecting Accessibility Guidelines during the Development Process of Web Sites in International Workshop on Web Usability and Accessibility, Nancy - France, 03/12/2007-07/12/2007, Vol. 4832, Springer, p. 470-480.