Towards Model-Based AHMI Automatic Evaluation

Juan Manuel González-Calleros*[1], Jean Vanderdonckt*, Andreas Lüdtke†, Jan-Patrick Osterloh†

* Université catholique de Louvain, Place des Doyens 1, 1348, Louvain-la-Neuve, Belgium

{juan.m.gonzalez, jean.vanderdonckt}@uclouvain.be

† OFFIS e.V., Escherweg 2, 26127, Oldenburg, Germany

{luedtke, osterloh}@offis.de

Abstract Aircraft cockpit system design is an activity with several challenges, particularly when new technologies break with previous user experience. This is the case with the design of the advanced human machine interface (AHMI), used for controlling the Advanced Flight Management System (AFMS), which has been developed by the German Aerospace Center (DLR). Studying this new User Interface (UI) requires a structured approach to evaluate and validate AHMI designs. In this paper, we introduce a model-based development process for AHMI development, based on our research in the EUs 7th framework project "Human". The first goal is to rely on this structured approach to perform automatic evaluation of the User Interface.

1 Introduction

Aircraft cockpit system design is an activity with several challenges, particularly when new technologies break with previous user experience. This is the case with the design of the advanced human machine interface (AHMI), used for controlling the Advanced Flight Management System (AFMS), which has been developed by the German Aerospace Center (DLR). The interaction between the pilot and the AHMI is through the User Interface (UI) composed of traditional control objects (buttons, spin button, menu) and non-traditional (compass rose, aircraft). The transformation of the existing character-based UI (left in Figure 1) for the AFMS into a graphical User Interface (middle in Figure 1) encounters new defies for the development process (analysis, design, implementation, evaluation) and their future usage.

---

[1] Corresponding author

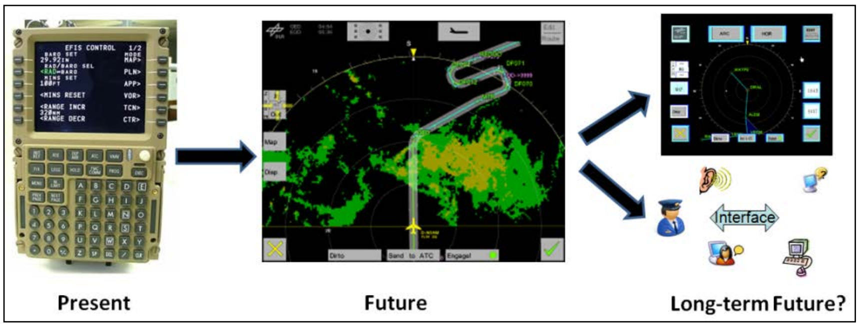

Fig. 1. AFMS evolution

Integrating evaluation in the loop of the design of the AHMI imply the use of pilots and a physical simulator. Thus this is costly and it would be hard to perform traditional UI tests considering that pilots are assets hard to find not just for their cost but also their availability. Moreover, flight simulators, located mostly in aeronautics research centers, are of limited access for long testing sessions. This stresses the need for a new approach, partially substituting pilots and the physical simulator, to conduct research on the AHMI evaluation. The focus of this work is to describe how to perform automatic UI evaluation of the AHMI.

Studying this new UI requires a structured approach to evaluate and validate AHMI designs. We claim that AHMI design is an activity that would benefit from relying on a model-based UI development (MBUID) approach, which offers, in principle, the opportunity to test different AHMI configurations. This chameleonic capacity of the UI in the MBUID context permits us to consider the evaluation of different layouts or the replacement of interaction objects of the AHMI without changing the source code just the models.

In this paper we rely on a structured reference framework, Cameleon, a User Interface Description Language, UsiXML, and a formal representation of the models, meta-models, to express the different aspects of the methodology. The UI of the AHMI is expressed using UsiXML formalism. Usability guidelines are also stored in the same formalism. The UI is checked against the guidelines and violations are listed and a solution is proposed. For instance, the text messages should always start with upper case and the rest of the words in the message, although it is a reserved word, should be in lower case. This kind of features can be evaluated automatically thanks to the use of software tool developed for this purpose, the Usability Adviser. The results of this evaluation complement the research on AHMI cockpit system design that is conducted with more sophisticated techniques where some other tools (virtual simulation platform of the aircraft) and techniques (cognitive modelling of pilots) are used to analyze pilots behaviour in order to identify why they commit errors. Another benefit of relying on the MBUID is that different modalities of interaction could be, in principle, also evaluated, if the models and the transformational knowledge needed exist. We explore this dimension to generate an alternative graphical (3D) representation of the AHMI (2D) or vocal interaction, see right side in Figure 1. The goal of this work is to perform some traditional usability evaluation on the preference between the two different renderings with users.

The reminder of this paper includes the review of the state of the art in the newt section. Followed by, the description of the proposed methodology. Next, the methodology is exemplified through a case study. Finally, the conclusions and future directions of this research are exemplified.

2 State of the Art

Interactive Cooperative Objects (ICOs) used to model aircraft interactive systems [11], such as: air traffic workstations, civil aircraft cockpit and military aircraft cockpit. The formal description for interactive cockpit applications uses Petri nets to describe dynamic and behavioural aspects of systems in the cockpit. A formal model of pilot-automation interaction and the characteristics of the UI are described in [11]. This work compared the effects and benefits of visual cues (labels, prompts, messages) to support mission tasks. However, there is a limitation on the widgets models and guidelines. The design knowledge in the methods is described to support the design of highly interactive systems such as the AHMI as they just model classical WIMP interfaces [11].

There have been some attempts, in the avionics context, to standardize formal methods of some aspect of the UI. The ARINC standard [1] defines protocols to communicate the dialogue and the functional core of cockpit display system [2]. This standard also considers the presentation level, i.e., a set of widgets are included as a recommendation but no design guidelines, no method to design UIs are considered in the standard [2]. Even more, the ARINC standard is not used for primary cockpit applications [11], such as Primary Flight Display and Navigation Display. It only deals with secondary applications involved in the management of the flight such as the ones allocated to the Multiple Control Display Unit [11].

Formal methods have been used in aviation but limited has been its use specifically when addressing the UI design. Existing attempts are partial or limited in their formalisation as they just denote the UI functionality in terms of state transitions but do not go further in the modelling particularly to evaluate multiple UIs.

Even more, among all the User Interface Description Languages (UIDL) complaints with the MBUID, a complete review can be found at [5], we are not aware of any attempt to rely on a MBUID to prototype avionic displays.

3 Model-Based AHMI Design

There is a global consensus about the components of a MBUID methodology [17], which are: a series of models, a language, an approach and a suite of software engineering tools. We rely on the USer Interface eXtensible Markup Language (UsiXML) [7], a formal methodology for describing a MBUID process. Relaying in a language engineering approach [18], UsiXML considers three levels of the language aspect: the syntax, semantics and stylistics of the language. The semantics are expressed as UML class diagrams that correspond to meta-models of the AHMI. The meta-models are transformed into a XML specification, which considers XML Schemas (abstract syntax) for the definition of valid XML. Finally the stylistics is the visual syntax mainly used to depict meta-models.

The proposed method is compliant with the structured CAMELEON reference framework [4]. Largely used in the literature for UI development, the CAMELEON reference framework adheres to the MBUID that has been applied widely to address the development of complex systems. As the frameworks promoted the use of different UI abstractions, in this paper we just focus in the layer that concerns to the concrete description model. The *Concrete UI Model* (CUI) allows both the specification of the presentation and the behaviour of an AHMI with elements that can be perceived by the users [7]. The CUI model is an abstraction of AHMI elements some of which are independent of programming toolkit. For instance, in Figure 2-A the AHMI is rendered in VRML while in Figure 2-B in openGL.

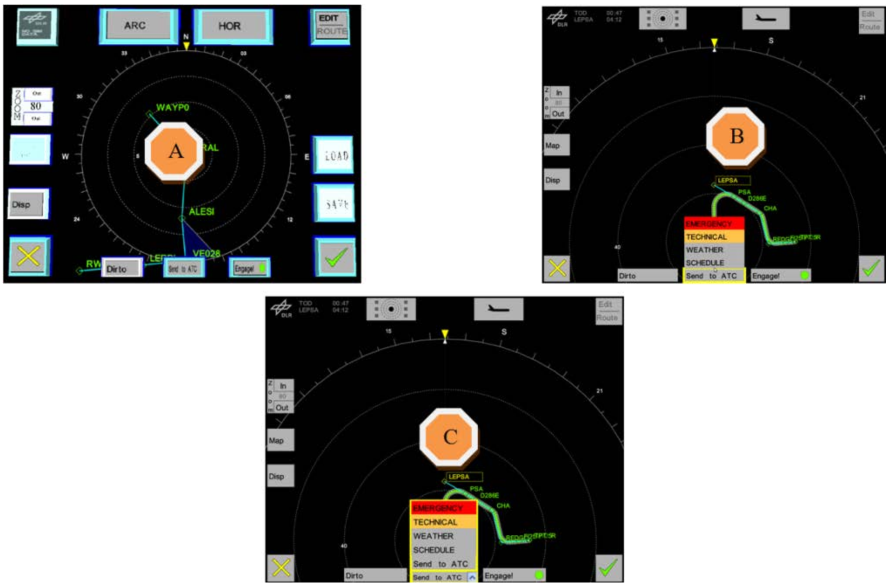

Fig. 2. Exploring diversity of widgets representation for the same task.

4 Evaluating the AHMI User Interface

The evaluation of the AHMI UI considers static aspects (UI layout, position of objects) and dynamic concepts (state of a button during the interaction, colour of the label). UI models are stored and then are object of further evaluation, automatic or manual. In this scenario, usability guidelines over the UI objects (distribution of the widgets composing the UI) could be evaluated.

We have used the semantics of the AHMI formalised with UsiXML to evaluate the UI against guidelines. Special attention was paid to those guidelines for standard certification and quality assurance and to express them in the Guideline Definition Language (GDL) [17], a XML-compliant language that is directly linked to UsiXML.

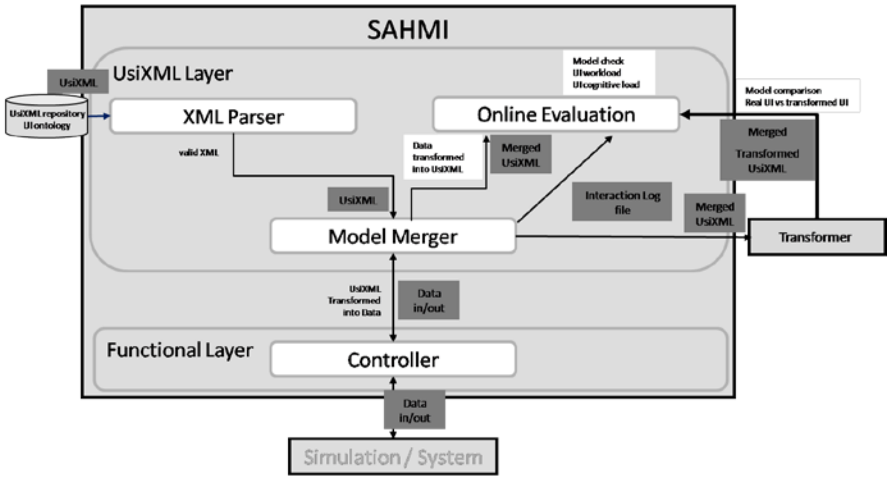

Fig. 3. Symbolic AHMI Architecture

4.1 Integrating UI Evaluation in a Simulation Environment

Different methods exist for evaluating a UI which mainly are divided in two categories: qualitative and quantitative approaches. Crew preferences and all kind of subjective data are gathered using different means, for instance questionnaires. There is always the need for crew members to provide feedback on the UI. Unfortunately, pilots are assets that are hard to find, so include them in the loop for constant UI evaluation is not feasible [13].

In the context of a simulation environment [8,9] where pilots are substitute by cognitive models, and a physical simulation platform by a virtual simulation environment, automatic evaluation of the UI can be done by including a UI evaluation layer to the simulation environment. In Figure 4, the Symbolic AHMI (SAHMI) architecture in the context of a virtual simulation platform is shown. A repository with UsiXML formalism describing the AHMI UI is used. This file is read using a parser that validates the specification and transforms this into a machine readable structure called model merger. The UI is complemented with dynamic and static data accessed via the simulation system. The Cognitive Architecture (CA) is used to simulate pilots' interaction with the AHMI. More details on the CA or the experiments are out of the scope of this paper, they can be found in [9]. Simulated pilots actions over the UI are passed as messages that are processed in the model merger. These data from the simulation system must be transformed to be compatible with UsiXML format. This data is store as a log File history (Figure 4-A).

The transformer module (Figure 3) modifies the specification of the UI trying to test multiple configurations. For instance, in Figure2-C a combo box is used instead a menu (Figure 2-B) for selecting the negotiation type with the ATC. Thus as result the UI timeline could be composed of different version of the UI to perform the same task. The first timeline corresponds (Figure 4-B) to the real simulated system as it is. The second timeline and subsequent would be the result of investigating different renderings of the same UI over time. For instance in Figure 4 the timeline B shows changes in the location of widgets (T1, T2, and Tn) and replacement of a widget (T3). The evaluation layer of the SAHMI keeps a trace of the evolution of the UI during the interaction. The Model Merger layer reconstructs the UsiXML and sends it to store it in the online evaluation tool.

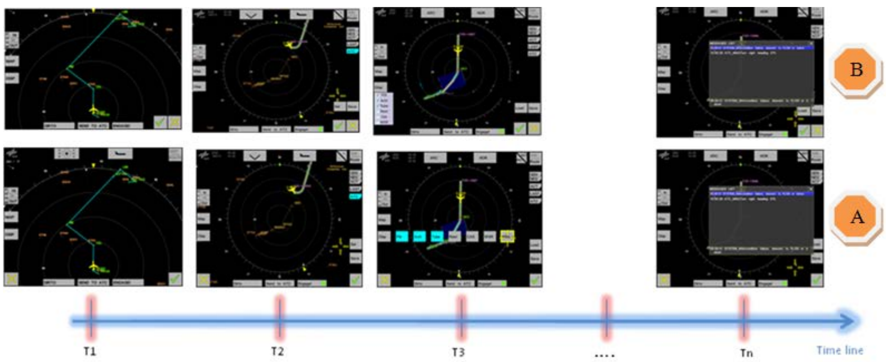

Fig. 4. AHMI UI Evolution over time (A) and modified version over the same UI evolution (B)

4.2 *User Interface Evaluation*

Guidelines evaluation can be automatically performed with the Usability Adviser [16]. Such evaluation can be automatically evaluated with the Usability Adviser [16], a tool to determine the ergonomics characteristics of a UI when it is specified in UsiXML. This tool evaluates ergonomic rules to determine, for instance, visual obstruction, colour coding. The software expresses usability guidelines as logical grammars. For example, a usability guideline that selects appropriate color combinations for the label on a slider, is described as follows: $i \in \text{Slider} : \neg$ (SliderColor(i,white) $\wedge$ LabelColor (i,yellow)).

The AHMI must not differ from a traditional UI. The traditional set of widgets must be used for the AHMI UI as much as possible by imitating their behaviour and graphical representation. This is needed as future pilots would be used to the computer interaction, thus, cockpit display systems should at least be consistent with systems of our daily life [15]. Even more important, traditional UI usability guidelines such as those listed in the ISO 9126 standard can be used to evaluate elements of the AHMI UI. There are some which have been corroborated in the avionics domain, for instance, messages should follow always the nomenclature: first letter in capital and the rest in lower case [13]. There are some other that refers to specific AHMI display systems such as the consistency in the roll index in the compass rose [14].

5 Conclusion

The AHMI is a new innovative system that introduces new challenges for the development of cockpit systems. Development steps including design and evaluation, among others, are normally limited addressed when it refers to the UI. Design knowledge is normally hidden and evaluation is mostly focused on the system functionality rather than of the usability of the system. In this paper we propose to rely on a model-driven approach for the development of AHMI that, among other advantages, can be coupled in a simulation environment. Modeling the SAHMI showed to be an option for UI evaluation. The model of the UI, as described in the paper, can be modified in order to test different UI configurations. Traditional measurements can be assessed like UI workload, color combination. Finally, the modality of interaction of the UI can be object of evaluation. While in this paper we showed how the original 2D rendering can be equally rendered in 3D. A future plan is to automatically generate the AHMI from its model and to submit it to runtime analysis. For the moment, only automated guideline review in perform through the UsabilityAdvisor.

Acknowledgments The research leading to these results has received funding from the European Commission Seventh Framework Programme (FP7/2007-2013) under grant agreement n°21988 Project Human.

References

1. ARINC 661-2, Prepared by Airlines Electronic Engineering Committee. Cockpit Display System Interfaces to User Systems. ARINC Specification 661-2, 2005.
2. Barboni, E., Navarre, D., Palanque P. & Basnyat, S. A Formal Description Technique for Interactive Cockpit Applications Compliant with ARINC Specification 661. In proceedings of SIES 2007 - IEEE 2th International Symposium on Industrial Embedded Systems July 4-6, 2007, Lisbon, Portugal.

3. Bierbaum, C.R., Szabo, S.M. Aldrich, T.B. Task analysis of the UH-60 mission and decision rules for developing a UH-60 workload prediction model, Volume 1: Summary report (AD-A210 763). Alexandria, VA: U.S. Army Research Institute for the behavioral and Social Sciences.
4. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers, Vol. 15, No. 3 (June 2003) 289–308.
5. Guerrero García, J., González Calleros, J.M., Vanderdonckt, J., Muñoz Arteaga, J. A Theoretical Survey of User Interface Description Languages: Preliminary Results, Proc. of Joint 4th Latin American Conference on Human-Computer Interaction-7th Latin American Web Congress LA-Web/CLIHC'2009 (Merida, November 9-11, 2009), E. Chavez, E. Furtado, A. Moran (Eds.), IEEE Computer Society Press, Los Alamitos, 2009, pp. 36-43.
6. Lepreux, S., Vanderdonckt, J., Towards a support of the user interfaces design using composition rules. Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006 (Bucharest, 6-8 June 2006), Springer-Verlag, Berlin, pp. 231-244.
7. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Lopez, V.: UsiXML: a Language Supporting Multi-Path Development of User Interfaces. In: Proc. of 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCIDSVIS'2004 (Hamburg, July 11-13, 2004). Springer-Verlag, Berlin (2005).
8. Lüdtke, A., Weber, L., Osterloh, J.P., Wortelen, B., Modeling Pilot and Driver Behavior for Human Error Simulation. HCI (11) 2009: 403-412.
9. Lüdtke, A., Osterloh, J.P., Simulating Perceptive Processes of Pilots to Support System Design, In Proc. 12th IFIP TC 13 International Conference (Interact 2009), Uppsala, Sweden, August 24-28, 2009, pp. 471-484.
10. McCracken, J.H., Aldrich, T.B., Analyses of selected LHX mission functions: Implications for operator workload and system automation goals (Technical Note ASI479-024-84). Fort Rucker, AL: U.S. Army Research Institute Aviation Research and Development Activity.
11. Navarre, D., Palanque, Ladry, J.F., Barboni, E., ICOs: A Model-Based User Interface Description Technique dedicated to Interactive Systems Addressing Usability, Reliability and Scalability, In: Transactions on Computer-Human Interaction, ACM SIGCHI, USA, Special issue User Interface Description Languages for Next Generation User Interfaces, ACM Press, 16(4), 2009, pp. 18:1-56.
12. Sharon Irving, Peter G. Polson, J. E. Irving: A GOMS analysis of the advanced automated cockpit. CHI 1994: 344-350.
13. Singer, G. and Dekker, S., The ergonomics of flight management systems: fixing holes in the cockpit certification net. Applied Ergonomics, 32 (3), p.247-254, Jun 2001.
14. Singer, G. and Dekker, S., The effect of the roll index (sky pointers on roll reversal errors. Human Factors and Aerospace Safety, 2 (1), p.33-43, 2002.
15. Singer G., Methods for Validating Cockpit Design The best tool for the task, Ph.D. thesis, Department of Aeronautics of the Royal Institute of Technology, Stockholm, Sweden, March, 2002.
16. Vanden Bossche, P., *Développement d'un outil de critique d'interface intelligent : UsabilityAdviser*, M.Sc. thesis, Université catholique de Louvain, Louvain-la-Neuve, 1 September 2006.
17. Vanderdonckt, J., Beirekdar, A., & Noirhomme-Fraiture, M. (2004) "Automated Evaluation of Web Usability and Accessibility by Guideline Review", In: Proc. of 4th Int. Conf. on Web Engineering ICWE'04 (Munich, 28-30 July 2004), Springer-Verlag, Berlin, pp. 17–30.
18. Vanderdonckt, J., A MDA-Compliant Environment for Developing User Interfaces of Information Systems, Proc. of 17th Conf. on Advanced Information Systems Engineering CAiSE'05 (Porto, 13-17 June 2005), O. Pastor & J. Falcão e Cunha (eds.), Lecture Notes in Computer Science, Vol. 3520, Springer-Verlag, Berlin, 2005, pp. 16-31.