HMI Engineering Productivity: the poor child of MDE/MDA trends. A vision for Model-Driven Human-Computer Interaction Engineering

Célestin Sedogbo, Olivier Grisvard, Frédéric Landragin, Jérôme Lard, Sébastien Praud

Thales Recherche et Technologie
RD 128
91767, Palaiseau cedex, France

AUTHOR'S DRAFT

ABSTRACT

The industry needs innovative development techniques that will boost productivity of SW development and maintenance, to be able to offer Adaptive Human-Machine Interface (Adaptive HMI) for their applications. Engineering processes and methods such as Model Driven Engineering (MDE), currently adopted in complex system development, are an appropriate response to this challenge. However, MDE currently provides little, if any, support to HMI design and development, and none to advanced interaction based on adaptivity and multimodality. The MDE paradigm applied to Adaptive HMI (noted here MDI for Model-Driven Interfaces), when combined with distributed and migratory User Interfaces (UIs) could enable significant reduction in time to market.

CATEGORIES AND SUBJECT DESCRIPTORS: D.2.2 [Design Tools and Techniques]: User interfaces; D2.11 [Software Architectures]; H.1.2 [User/Machine Systems]: Human Information Processing; H.5.2 [User Interfaces]; I.3.6 [Methodology and Techniques]: Interaction Techniques.

INTRODUCTION

Digital technology is now pervasive. We interact with such technology, often without realising it, in many settings and forms: kiosk systems in public, desktop computers in offices, handheld PDAs during nomadic work, navigation systems while driving, and embedded entertainment systems at home. However, despite immense technological progress, users' interaction with digital devices is still often too frustrating, inefficient or ineffective, especially when work and leisure activities are spread over several separate devices and occur in different contexts of use.

Many solutions to sub-optimal human-computer interaction (HCI) have been proposed, and some have proved to be effective in R&D prototypes. For example, adaptive and adaptable systems have been successfully developed to avoid the inflexibility of one-size-fits-all designs. However, such solutions tend to be developed and tested in isolation. Furthermore, they remain impractical for mainstream products and for professional applications due to the high development complexity required with currently available technology.

Engineering processes and methods such as Model Driven Engineering (MDE), currently adopted in complex system development, are an appropriate response to this challenge. Our vision is that we must develop a systematic model-based approach covering all life-cycle phases of Adaptive HMI, more precisely Adaptive & adaptable Multimodal Interactive Systems, as well as a distributed and migratory approach.

This paper proposes a vision for application of Model-Driven Engineering concepts to Adaptive Human-Machine Interface engineering

OUR APPROACH

Global view

Context-aware, adaptive applications for tomorrow's ubiquitous mobile applications require a unified development process covering the whole software lifecycle comprising analysis, design, implementation, deployment, testing, and maintenance.

Innovation in model-based architectures for user interfaces

The idea is to define generic modelling concepts (meta-models) for multimodality and context-aware adaptation as well as model transformation methods and mark-up language(s) that enable the deployment of useful and usable Adaptive HMI in a cost-effective manner.

Existing conventional models should be extended to cover new requirements in relation to multimodality and context-aware adaptation, e.g., domain-dependent concepts and task model, context model (user, environment, platform), abstract and concrete UI models. Also, new models will be defined for issues not currently addressed by the state of the art, e.g., interactors and programs models, usability model, workflow and process models.

Innovation in model-driven user interfaces

Model-based approaches enable designers to avoid dealing with a plethora of details, thus reducing the development costs while improving quality. However, model-based approaches such as UML stereotypes, have paid little attention to the design of the UI per se. In addition, they do not provide any support for adaptive and multimodal UI design. The same holds for the MDE/MDA approaches. MDE/MDA is today the standard for model-based system design. As such, we cannot avoid positioning our approach regarding this standard. We will therefore examine how the design process for Adaptive HMI can conform to the recommendations of MDE/MDA.

With regard to interactive systems design, one of the major innovation we propose will consist of using application/system models as a basis from which high-level models, such as task or domain models, will be generated automatically.

The design process for Adaptive HMI, making intensive use of modelling and model transformation techniques, will rely as much as possible on existing standards and tools that support model-based approaches. As such, UsiXML along with its associated modelling tools for forward and reverse engineering (GraphiXML, TransformiXML, ReversiXML, …) will provide a sound starting point.

CONCLUSION ET PERSPECTIVES.

The key idea developed in this paper is to apply mainstream software engineering, like MDE, to the deployment of Adaptive HMI.

Model transformations will be used at design time and integrated into a design process for Adaptive HMI infrastructure. A particular focus is to be put on the capability to reuse as much knowledge as possible out of the application or system model as defined classically by the software engineering community.

BIBLIOGRAPHIE

[Arch 1992] Arch, A Metamodel for the Runtime Architecture of An Interactive System, The UIMS Developers Workshop, SIGCHI Bulletin, 24(1), ACM (1992).

[Balme 2004] Balme, L., Demeure, A., Barralon, N., Coutaz, J. and Calvary, G. CAMELEON-RT: a Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces, second European Symposium on Ambient Intelligence, EUSAI 04, EUSAI 2004, LNCS 3295, Markopoulos et al. pp. 291-302.

[Bass 1998] Bass, L., Clements, P. and Kazman, R. *Software Architecture in Practice*. Addison Wesley Publ., ISBN 0-201-19930-0, 1998.

[Calvary et al. 2003] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L. and Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers 15(3), 289–308, 2003.

[Chatty et al. 2004] S.Chatty, S.Sire, J.L.Vinot, P.Lecoanet, A.Lemort, C.Mertz. Revisiting Visual Interface Programming: Creating GUI Tools for Designers and Programmers. In UIST'04, the Seventeenth Annual ACM Symposium on User Interface Software and Technology. Santa Fe, New Mexico, October 2-27, 2004.

[Coutaz 2002] Coutaz ,J. and Rey G., Foundations for a Theory of Contextors. Third conference onComputer-Aided Design of User Interfaces, CADUI 2002, J. Vanderdonckt, C. Kolski Eds., Kluver Academic Publ., 2002, pp. 13-32.

[Dey et al. 2001] Dey, A.K., Salber, D., Abowd, G.D., A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, anchor article of a special issue on Context-Aware Computing, in the Human-Computer Interaction Journal, Vol. 16, 2001.

[Hong 2001] Hong, I. and and Landay, J. An Infrastructure Approach to Context-Aware Computing. In Human-Computer Interaction ,Vol. 16, (2001).

[Jespersen & Linvald 2003] J. W. Jespersen and J. Linvald. Investigating user interface engineering in the model driven architecture. In Proceedings of the Interact 2003 Workshop on Software Engineering and HCI. IFIP Press, Sept. 2003.

[Kobsa 2004], Kobsa, A. User Modeling and User-adapted Interaction. 14(5), January, 2004, 469-475.

[Luyten 2002] Luyten, K., Vandervelpen, C., Coninx, K. Migratable user interfaces Descriptions in Component-Based Development. DSV-IS 2002, Rostock, Springer Verlag Publ., 2002.

[Nigay 1995] Nigay, L. and Coutaz, J. A Generic Platform for Addressing the Multimodal Challenge, *in Proc. CHI'95 Human Factors in Computing Systems,* ACM New York, Denver, pp.98-105 (1995).