A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties

Víctor López-Jaquero[†‡], Francisco Montero[†‡], José P. Molina[†‡], P. González[†], A. Fernández-Caballero[†]

[†] Laboratory on User Interaction & Software Engineering (LoUISE)
University of Castilla-La Mancha, 02071 Albacete, Spain
{ victor | fmontero | jpmolina | pgonzalez | caballer }@info-ab.uclm.es

[‡] Belgian Laboratory of Computer-Human Interaction (BCHI)
Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium
{ lopez | montero | molina}@isys.ucl.ac.be

Abstract. This work is aimed at the specification of usable adaptive user interfaces. A model-based method is used, which have been proved useful to address this task. The specification created is described in terms of abstract interaction objects, which are translated into concrete interaction objects for each particular platform. An adaptive engine is also proposed to improve the usability at runtime by means of a multi-agent system.

1 A seamless process for adaptation development

Currently different interaction paradigms are emerging due to several factors, such as ubiquitous access to information, the consideration of different user expertise levels, accessibility criteria or the wide range of interaction devices with specific capabilities (screen size, memory size, computing power, etc). In this paper a method is introduced for the specification of user interfaces of highly interactive systems with the capability of self-adapting to the changes in the context-of-use.

To fill the gap between model-based user interface development approaches and adaptive user interface frameworks, we propose enriching the usual model-based user interface development, to include, in a seamless manner, the development of the adaptation facilities required for adaptive user interfaces development. We propose a method for the development of adaptive user interfaces called *AL-BASIT* (Adaptive Model-Based User Interface Method), which extends usual model-based user interface development methods to support the development of adaptive user interfaces in a seamless way. Our proposal starts with requirements analysis to identify the tasks that will drive the design. Also user, physical environment and platform characteristics are collected to complete requirement analysis. These characteristics will be needed to adapt user interface design. In requirements analysis, use cases are used to identify the tasks and to establish a comprehensible channel of communication with the user, using an artefact understandable by the user and the designer. This stage is completed gathering the required data from the potential context of use for the application (user, platform and environment models). Analysis stage in aimed at the transformation of the requirements into a specification easier to handle, and usually in a more compact format. It also brings requirements analysis data closer to designer language. In our approach, we are using UML class diagrams to describe the domain model. To support human role multiplicity, we match each possible role a user can assume when using the user interface with the tasks they can perform. After analysis stage, design phase take place using the proposed tool. The design is based on the description of the identified tasks and their relationships with the domain elements they make use of. The task model is enriched describing the events to change from one task to another with the canonical abstract user interface tools [1]. From this data, an abstract user interface [4] is generated which is independent of both modality and platform. Then, a translation is made to a concrete user interface (CUI) expressed in USIXML (http://www.usixml.org) user interface description language. The coordination between the CUI elements, the application functional core and the final running code is performed by means of connectors, as described in [2][3] This specification is adapted at runtime using a transformational approach. The adaptation engine reasons about the possible adaptation and preserves different usability properties according to the usability trade-off specified in terms of I* specification technique.

2 Conclusions

In this paper we have introduced a method for the development of adaptive user interfaces. It improves the usability of the system by adapting the user interface to the context-of-use at runtime. Thus, the user interface is adapted according to the changes in the context-of-use. For the design of adaptation engine, a multi-agent system is used. The goals of the agents in the multi-agent system are guided by the adaptation trade-off specified by the designer at design time using a goal-driven requirement notation: I*.

3 Acknowledgements

We gratefully acknowledge the support of the spanish PBC-03-003 grant and the SIMILAR network of excellence (http://www. similar.cc).

4 References

1. Constantine, L. *Canonical Abstract Prototypes for Abstract Visual and Interaction Design*. Proceedings of DSV-IS. Springer Verlag, LNCS 2844, 2003.
2. Lopez-Jaquero, V., Montero, F., Fernandez-Caballero, A. Lozano, M.D. *Towards Adaptive User Interface Generation: One Step Closer To People*. 5th International Conference on Enterprise Information Systems, ICEIS 2003. Angers, France, 2003.
3. Lopez-Jaquero, V., Montero, F., Molina, J.P., Fernandez-Caballero, A., Gonzalez, P. *Model-Based Design of Adaptive User Interfaces through Connectors*. DSV-IS 2003. Springer Verlag, LNCS 2844, 2003.