

Model Driven Development of Advanced User Interfaces (MDDAUI) – MDDAUI’06 Workshop Report

Andreas Pleuß¹, Jan van den Bergh², Stefan Sauer³, Heinrich Hußmann¹, and
Alexander Bödcher⁴

¹ University of Munich, Germany
{Andreas.Pleuss, Heinrich.Hussmann}@ifi.lmu.de

² Hasselt University, Belgium
Jan.VandenBergh@uhasselt.be

³ University of Paderborn, Germany
sauer@upb.de

⁴ University of Kaiserslautern, Germany
boedcher@mv.uni-kl.de

Abstract. This paper reports on the *2nd Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI’06)* held on October 2nd, 2006 at the *MoDELS’06* conference in Genova, Italy. It briefly describes the workshop topic and provides a short overview on the workshop structure. In the main part it introduces the four topics discussed in the workshop’s afternoon sessions and summarizes the discussion results.

1 Workshop Topic

The user interface of an application is often one of the core factors determining its success. While model-based user interface development is an important line of research in the human-computer-interaction (respectively human-machine-interaction) community, model-driven application development is an important area in the software engineering community. This workshop aims at integrating the knowledge from both domains, leading to a model-driven development of user interfaces. Thereby, the focus of the workshop lies on advanced user interfaces corresponding to the current state-of-the-art in human-computer-interaction, such as multimedia or context-sensitive user interfaces or multimodal interaction techniques.

The workshop builds up on the results of the previous edition [1, 2], which provided an overview on existing work and the challenges in the area of MDDAUI. On that base, the current workshop aims to go more into specific details and specific challenges on the field. This includes e.g. more specific models for advanced UIs, a stricter compliance to the concepts from model-driven engineering, explicit transformations between UI models which in particular provide concepts to ensure the usability of resulting UIs, and additional integration of informal techniques to achieve a better integration of usability and artistic design into the model-driven development process.

2 Submissions, Participants, and Program

Interested participants were asked to submit a short paper of four pages length in double-column format. We received 18 submission from which 12 have been accepted. The resulting spectrum of participants included people from different areas in human-computer-interaction and software engineering. Besides people from academia, there were also participants working in industrial context.

The workshop took one day during the MoDELS'06 conference. In the morning sessions all accepted papers were presented either at short or as long presentations. The afternoon was mainly reserved for discussions. The detailed program and the preliminary proceedings including all accepted papers can be found on the workshop webpage [3]. Official proceedings will be available at [4].

3 Workshop Discussions

After the paper presentations four discussion groups were formed around the following topics:

- Co-development of Models and Visualizations
- Runtime Interpretation of UI Models
- MDDAUI and Usability
- An Integrated Metamodel for MDDAUI

In the following we summarize the discussion results of these discussion groups, which were also presented at the end of the workshop.

3.1 Co-development of Models and Visualizations

In the context of the discussion, visualization means the concrete visual appearance of the user interface. The user interface development process involves different developer groups, like graphic designer or usability experts. Some of them usually work on concrete visualizations like sketches, hi-fi and low-fi prototypes. Thus, their results have to be synchronized with the models which provides the central information in an abstract way for all parties involved in the development process.

As a consequence, there is a need for tools and transformations which allow a bottom-up process, where the abstract information for the models is derived from the various visualizations. Currently, most model-driven approaches focus on the top-down process. Furthermore, it must be possible to update the visualizations when changes on the model occur. This leads to the need of a round-trip engineering between models and different kinds of visualizations even in an early stage of the development process.

Finally, the discussion addressed the issue that the relative importance of the models versus visualizations changes for different kinds of applications. Visual appearance of the user interface is especially important in applications for consumer products or in multimedia applications like in the automotive sector. On

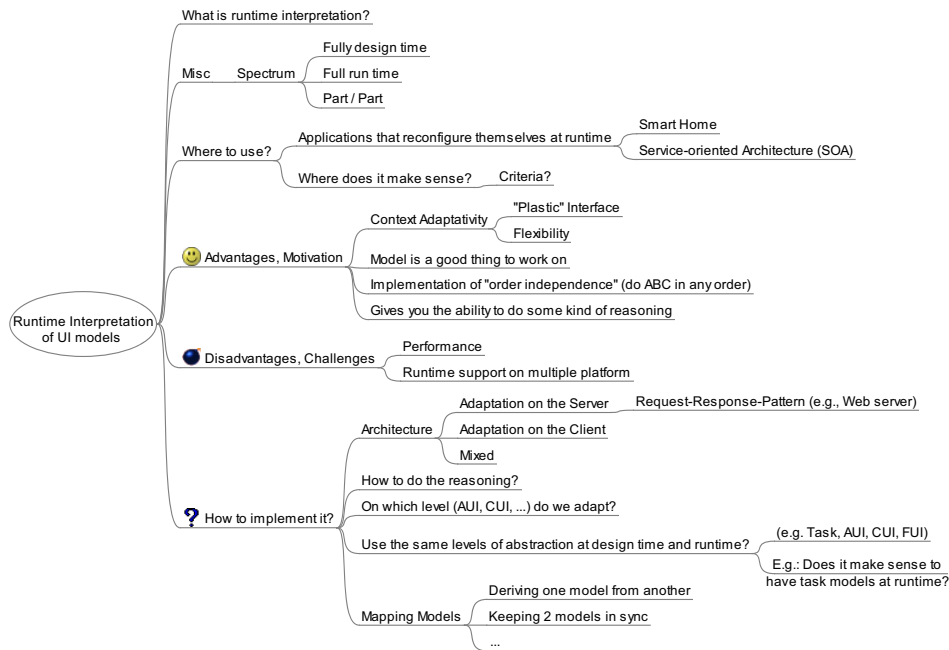


Fig. 1. Overview on the different aspects of runtime interpretation of user interface models

the other hand e.g. for a corporate intranet the flexible adaptation to the continuously changing information seems more important than the concrete visual appearance of the user interface.

3.2 Runtime Interpretation of UI Models

In the context of our discussion, runtime interpretation means that the abstract information from the model is kept during the execution of the modelled application and is interpreted by a specific runtime environment. This is necessary e.g. in ubiquitous computing scenarios where the user interface must be adapted at runtime to the context of the application, e.g. the currently available target devices. In the discussion we aimed for an overview on the current state-of-the-art and the challenges in this area. As a result, we created the mind map shown in figure 1.

3.3 MDDAUI and Usability

A large part of the discussion in this group focused on identification of challenges to create usable interactive applications within industrial settings. A first challenge is the fact that the people involved with the design of these interactive applications are faced with some design decisions that are made without their

involvement, especially regarding the hardware platform. For example, decisions regarding hardware in the automotive industry are largely driven by artistic designers and management decisions based on perceived market needs or desires.

Another important challenge is that the target platform can change during the design process, for example due to market changes. These changes can have an enormous impact on the design of the interactive application, especially for embedded applications where only a limited number of physical controls can be used and strong ergonomic rules apply to the usage of these controls. For example, a decision to replace a rotation knob for navigating through menus by a touch screen in a car can require a complete redesign of the user interface structure caused by the ergonomic rules that are associated with these controls. Using a higher level of abstraction through models may make it easier to cope with these changing hardware platforms.

The usage of models, however, poses some challenges since the teams that design interactive applications are very heterogenous. It therefore is a challenge to communicate designs, especially in abstract models, to all that are involved in the design of interactive applications. One possible path to cope with the problem that was considered to be worth further investigation is the usage of a domain-specific language with a specialized concrete syntax for the (abstract) models.

3.4 An integrated metamodel for UI development

The discussion started with the idea, that most of the different approaches for different kinds of (advanced) user interfaces partially base on the same or very similar modeling concepts (for the core properties of a user interface) and partially very different concepts (e.g. for a specific property of the UI like context-sensitiveness, multimedia context, 3D, or a specific modality). Concepts from model-driven engineering – like explicit metamodels, operations on metamodels and explicit transformations – could perhaps help to define an overall framework to capture these commonalities and variabilities. This could result e.g. in an overall metamodel or a family of languages, which then allows the flexible combination of different concepts (metamodels) according to the properties of the user interface to be built. For example, one can then select the required models to develop a user interface which is context-sensitive but also includes 3D-animations.

To get a feeling about how this could look like, we first collected the most common models for UI development known from the literature. Then we discussed on alternatives how to integrate these different models. Two approaches were considered: The first is the creation of a core metamodel which is common to all MDDAUI approaches, complemented with extensions to cope with specific concerns. However, it seems difficult to agree on such a core metamodel. The second approach is creating metamodels for all MDDAUI approaches complemented with various operations on these metamodels. Examples of such operations are the transformations from one metamodel into another one or merging two metamodels. This allows a flexible and modular specification of models and

transformations and even the co-existence of an arbitrary number of alternative approaches.

For the further discussion we decided for the second approach. A collection of metamodels and transformations can be realized by metamodel repositories or 'metamodel zoos' as proposed by several initiatives on the MoDELS'06 conference. In the last part of the discussion we structured the collected models into packages, which could be for example: A package *Domain* which contains models for the application logic, a package *Context* which contains models for the context of the user interface, a package *Behavior* which contains models for the behavior of the user interface and a package *Appearance* which contains models for the structure and the concrete layout of the user interface.

4 Conclusion

The growing number of workshop participants from different communities indicates the high relevance of model-driven user interface development. The workshop's results show that the involved research areas, software engineering and human-computer-interaction, can both strongly benefit from the integration of their knowledge. In the context of the workshop topic, the workshop contributions show on the one hand that applying concepts and standards from model-driven engineering – like explicit metamodels and transformations or round-trip-engineering techniques – can seriously contribute to the solution of current challenges in user interface development. On the other hand, the area of model-driven engineering benefits not only through the consideration of knowledge from human-computer-interaction itself, but also through new insights and challenges arising from the complex application domain of user interface development.

Acknowledgements We would like to thank the workshop participants for their high quality contributions as well as the program committee members for their help and the valuable reviews.

References

- [1] Pleuß, A., Van den Bergh, J., Hußmann, H., Sauer, S.: Workshop Report: Model Driven Development of Advanced User Interfaces (MDDAUI). In: Jean-Michel Bruel (Ed.): Satellite Events at the MoDELS 2005 Conference, LNCS 3844, Springer 2006
- [2] Pleuß, A., Van den Bergh, J., Hußmann, H., Sauer, S.: Proceedings of Model Driven Development of Advanced User Interfaces. CEUR Workshop Proceedings, Vol. 159, 2005, <http://ceur-ws.org/Vol-159>
- [3] Second International Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI 2006), Workshop Webpage, 2006 <http://planetmde.org/mddaui2006/>
- [4] Pleuß, A., Van den Bergh, J., Hußmann, H., Sauer, S., Bödcher, A.: Proceedings of Model Driven Development of Advanced User Interfaces 2006. CEUR Workshop Proceedings (to appear), <http://ceur-ws.org/>