

Third International Workshop on Model Driven Development of
Advanced User Interfaces

Non Peer-reviewed author version

Pleuss, A.; VAN DEN BERGH, Jan; Sauer, S.; Görlich, D. & Hussmann, H. (2008)
Third International Workshop on Model Driven Development of Advanced User
Interfaces. In: Workshop and Symposia on Models in Software Engineering. p. 59-64..

Handle: <http://hdl.handle.net/1942/9169>

Third International Workshop on Model Driven Development of Advanced User Interfaces

Andreas Pleuß¹, Jan Van den Bergh², Stefan Sauer³, Daniel Görlich⁴,
and Heinrich Hußmann¹

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

goerlich@mv.uni-kl.de

Abstract. The workshop *Model Driven Development of Advanced User Interfaces (MDDAUI)* aims at integrating results from the area of human-computer interaction and user interface modeling with the concepts of model-driven engineering. This paper provides a summary on the third edition of MDDAUI held as part of the MoDELS 2007 conference in Nashville, USA. In particular, it presents the results of the two group discussions of the workshop.

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 is 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 editions in 2005 [1] and 2006 [2]. While the first two editions mainly provided an overview of the topic, the current edition aims at addressing more advanced topics in this area, including usability of the approaches, their integration with informal techniques like visual design and prototyping, and a better utilization of concepts from model-driven engineering, e.g., more flexible transformations leading to better adapted user interfaces. The research area is still evolving as new user interface technologies and paradigms are coming along and model-driven technologies are becoming more mature.

2 Submissions, Participants, and Program

Interested participants were invited to submit a paper of four pages length in double-column format. We mainly asked for technical papers presenting concrete solutions that consider the concepts of model-driven engineering, e.g., specified in terms of a concrete metamodel. We received fourteen submissions and accepted nine of them as the result of an intensive peer-review process by the programme committee. Five papers were selected to be presented as long presentations and four for short presentations at the workshop. Workshop participants basically came from two different groups: Some were frequent participants of the MoDELS conference who have a strong background in model-driven development while others were mainly based in the area of human-computer interaction (or human-machine interaction).

The workshop took one day during the MoDELS conference. In the morning sessions all accepted papers were presented; according to the foregoing selection process either in a long presentation or a short presentation. After each talk there was some discussion and in addition there was time for additional discussion and general questions at the end of each session. In the afternoon the topics for group discussions were selected and participants joined one of the two discussion groups. The detailed program can be found on the workshop Web page [4]. Official proceedings including all accepted papers are available electronically [3].

3 Workshop Discussions

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

- Usability of Model-Driven UI Development for the Developers
- Combination of Modeling Techniques

At the end of the workshop, both groups presented their discussion results which are summarized in the following.

3.1 Usability of Model-Driven UI Development for the Developers

A core issue in user interface development is usability: The user interfaces resulting from a user interface development process must be usable, and the development process should ensure usability as much as possible. In addition, the development tools and methods itself should be usable as well. This, in turn, will improve the usability of the developed user interfaces. The discussion summarized here focuses on the usability of a model-driven user interface development process.

A specific characteristic of user interface development is the participation of different experts, like graphic designers, interaction designers, and programmers. The integration of these experts is one of the main benefits models can provide in user interface development. The discussion group aimed at further specifying the role of models in user interface development. Finally, the following statement was found adequate:

Models shall act as a kind of bridge between input from various people involved in UI development (end users, domain experts, UI developers, management people, etc.) to integrate all this knowledge and to transfer it into the software engineering process.

Participants agreed that currently no modeling language completely satisfies in covering all aspects of all the different experts. Moreover, it was considered likely that a “universal” language of this kind, covering all possible situations and projects, will be too complex to be used in practice. An attractive solution may come from domain-specific user interface modeling languages (DSUL) and corresponding tools. It was mentioned that the required meta-tools for this approach are not easy to handle, and probably the definition of a DSUL for a specific project, company or domain may be expert work or a consultancy task.

In order to actually be accepted by user interface developers, such a modeling approach has to offer significant practical advantages. From the comparison with other technologies, which were around for some time before having been embraced by the developer community (code version management, IDEs), it becomes clear that a successful tool must leverage reuse of previous work. Things to be reused in the user interface context are, among others: patterns, anti-patterns, usability heuristics, usability metrics, interface components, etc. In particular, reuse here includes knowledge about the usability of user interfaces.

Furthermore, it has to be taken into account that the people who will actually work with a modeling tool are quite heterogeneous, as mentioned above. Each of these user groups will have a specific view on a model. So for instance, end users and domain experts can be involved best if the tool comprises a facility to carry out prototyping experiments or to incorporate the results of such experiments. For modeling experts, tools for creating abstractions from actual interaction sequences will be most important. For user interface developers, the tool shall provide detailed technical information; maybe even on code level. In order to keep all these views consistent under modifications in various views, techniques like those known as “round-trip engineering” from CASE tools are desirable.

3.2 Combination of Modeling Techniques

In model-driven user interface development, a large spectrum of different approaches exists. Although there are many commonalities between these approaches, they are often isolated from each other. However, there is variability between them because of their distinguished focus or the particular application domain. For example, some of them support context-sensitive user interfaces, collaborative applications, mixed interactive systems, multimedia user interfaces, etc.

This discussion aims at a better understanding of the relationships between the different models and concepts in user interface modeling. The resulting benefits can be in first place a better general understanding. In longer terms, it might also be possible—considering the concepts of model-driven engineering—to provide operations on the different metamodels in user interface development

to support a seamless spectrum of user interfaces to be developed. For example, a user interface can be multimedia *and* context-sensitive. In the ideal case, modeling support for such a user interface can be provided by reusing and combining modeling concepts (metamodels) for these two aspects. The combination of metamodels probably has to be performed manually, but a general overview on the landscape of existing modeling concepts can strongly help the modeling experts to identify and reuse the required concepts. It should be mentioned that basically it would also be possible to construct an overall metamodel for user interface development which already covers all possible aspects of user interfaces, but in practice this would be a very hard and error-prone task.

In the discussion we identified the following issues as the starting point:

- Identify relevant dimensions to structure the models.
- Identify relationships between user interface models.
- Identify relationships from user interface models to other kinds of models (of the software engineering domain) in application development.

Initially, two dimensions were identified to be most relevant for structuring the user interface metamodels:

Semantic aspects: The aspects of the user interface represented by a specific metamodel; i.e., the kind of information in the model. Examples are the user tasks to be supported by a user interface or its dynamic behavior. A specific aspect of the user interface can be represented by different modeling concepts, i.e., different metamodels. For example, the user tasks can be represented by *ConcurTaskTrees*, or *MAD*. *Discourse Models* do not directly represent user tasks but can (partially) replace task models as they also specify the purpose of the user interface but on a different level of abstraction. The dialogue or interaction to be performed by the user interface can be specified for example using *Statecharts* or *Petri Nets*.

Level of abstraction: The level of abstraction on which a metamodel represents the respective aspects of the system. Usually, the level of abstraction is related to the position of the model within the development process: In early stages the user interface will be modeled in a more abstract way while models specifying the final implementation must be very concrete. However, the concrete temporal order of models within a development process depends on the chosen process. In particular, it might be possible that some very concrete information is already available at an early stage of the development. In this case, it can be useful to be able to specify it immediately.

Figure 1 provides a first attempt to arrange some typical user interface models within these two dimensions. Two kinds of relationships are specified:

Contributes to: A (meta-)model contributes to another (meta-)model, i.e., the target model is created based on information from the foregoing model. In particular, this can mean that the target model is derived by a transformation from the source model. For example, the *Abstract Presentation* and the *Dialogue* are usually created based on a *Task* model.

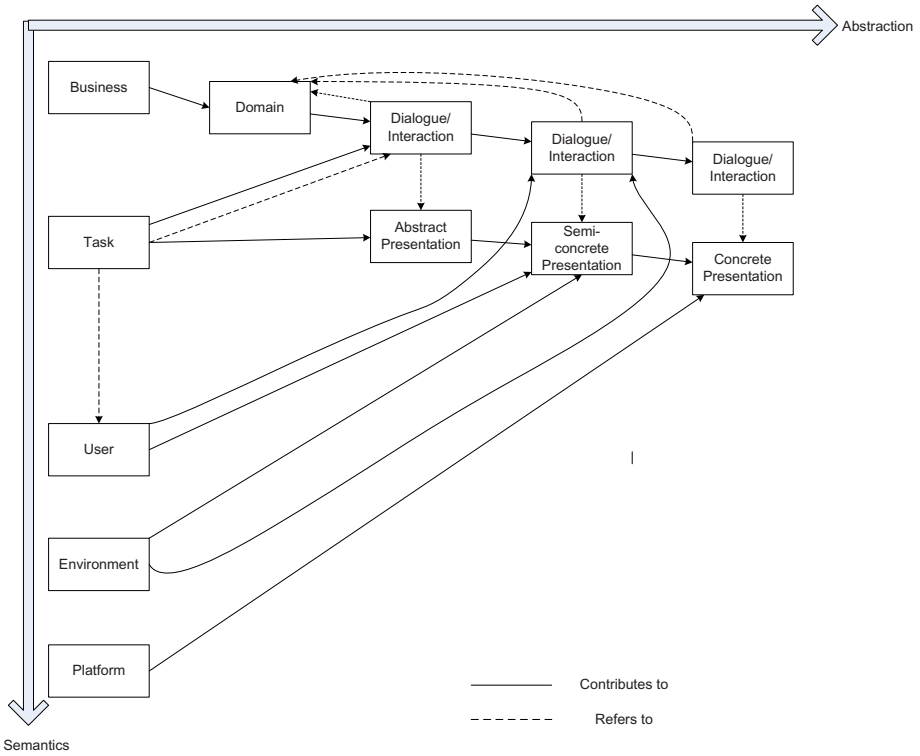


Fig. 1. A first sketch towards a schema structuring common metamodels in user interface modeling and their relations

Refers to: If a model refers to another, existing model, this means that some model elements in the former model contain references to model elements specified in another, the latter model. For example, the *Dialogue* usually refers to the *Domain* model as for example interactive user interface elements usually trigger some operations from domain classes.

The models in the presented schema basically can be applied to the development of any kind of user interface. Some domains require models for additional information, e.g.:

- user interfaces for mixed interactive systems require specification of physical objects,
- user interfaces for multimedia applications require specification of media objects.

4 Conclusion

The area of model driven engineering is continuously evolving and, as the workshop discussions show, the area of user interface modeling can directly profit from

these advances. This does not only concern better tool support but also the foundations. Currently, well-defined metamodels allow comparing and reusing user interface modeling concepts. Tunability of transformations can be used as mechanism to ensure the usability of developed user interfaces. In the future, advanced concepts of model driven engineering, like model weaving, might be used for more flexible integration of different concepts for advanced user interfaces, like for user interfaces supporting collaboration, multimedia, *and* mixed-reality.

Acknowledgements. We thank the workshop participants for their high quality contributions as well as the program committee members for their 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: Bruel, J.-M. (ed.) MoDELS 2005. LNCS, vol. 3844, pp. 182–190. Springer, Heidelberg (2006)
- [2] Pleuß, A., Van den Bergh, J., Hußmann, H., Sauer, S., Bödcher, A.: Model Driven Development of Advanced User Interfaces (MDDAUI) - MDDAUI 2006 Workshop Report. In: Kühne, T. (ed.) MoDELS 2006. LNCS, vol. 4364, pp. 100–104. Springer, Heidelberg (2007)
- [3] Pleuß, A., Van den Bergh, J., Hußmann, H., Sauer, S., Görlich, D.: MDDAUI 2007 – Model Driven Development of Advanced User Interfaces 2007. In: CEUR Workshop Proceedings, vol. 297 (2007), <http://ceur-ws.org/Vol1-297>
- [4] Third International Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI 2007) (2007), Workshop Web page, <http://planetmde.org/mddaui2007/>