Model-Driven Development of Advanced User Interfaces

Jan Van den Bergh
Hasselt University - tUL – IBBT
Expertise Centre for Digital Media
Wetenschapspark 2
3590 Diepenbeek, Belgium
Jan.vandenbergh@uhasselt.be

Gerrit Meixner
DFKI
Trippstadter Str. 122
67663 Kaiserslautern, Germany
Gerrit.Meixner@dfki.de

Kai Breiner
Fraunhofer IESE
67663 Kaiserslautern, Germany
breiner@cs.uni-kl.de

Andreas Pleuss
Lero
University of Limerick
Limerick, Ireland
Andreas.Pleuss@lero.ie

Stefan Sauer
University of Paderborn
s-lab – Software Quality Lab
33098 Paderborn, Germany
sauer@s-lab.upb.de

Heinrich Hussmann
University of Munich
Amalienstr. 17
80333 Munchen, Germany
heinrich.hussmann@ifi.lmu.de

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract
The workshop on model-driven development of advanced user interfaces will be a forum of multi-disciplinary discussion on how to integrate model-driven development with the often more informal methodologies used in user-centered design. Starting point of the discussion will be the tools, models, methods and experiences of the workshop participants.

Keywords
Model-driven development, user-centered design, models.

ACM Classification Keywords
D.2.2. Design Tools and Techniques: User Interfaces, H.5.2 User Interfaces (user-centered design), I.6.5 Model development (modeling methodologies) H.5.m Miscellaneous.

General Terms
Algorithms, Design, Human Factors, Languages

Background: Model-Driven Development
Model-Driven Development (*MDD*, also *Model Driven Software Development*, *Model Driven Engineering* [1][2]) has become an important paradigm in software development. It uses models as primary artifacts in the

development process. Models are often visual models, like UML models, but can also be represented in textual formats like XML. Each model is denoted by a modeling language. In MDD, a modeling language is defined by a model itself, the so-called *meta-model*. A model-driven development process usually makes use of different models on different levels of abstraction. Model *transformations* are used to transform a model (semi-) automatically into another (usually less abstract) model and finally into the final implementation code.

MDD provides a large number of powerful concepts and tools to deal with models, meta-models, and model transformations. They provide support for systematic and efficient software development.

MDD for User Interfaces

Model-driven development of user interfaces applies the principles of MDD to the target domain of user interfaces. Thereby, the modeling concepts have evolved from the large area of m*odel-based user interface development* [3][4]. While these approaches have not spread widely into practice in the past, the emerging need for device independence makes model-driven approaches more and more important [5]. Since one major advantage of MDD is that there are different models for particular purposes (e.g. different views in UML or other modeling languages of the same system, according to the stakeholders) this also applies to MDD for user interfaces. Here, one well-known use-case is to separate the content (*what* is displayed) from the design (*how* it is displayed) into distinct models.

For instance, the paradigm of ubiquitous computing assumes user interfaces to run on diverse target platforms in a consistent way. User interfaces are expected to adapt at runtime to the current application context and available devices or even migrate at runtime between different devices. Platform-independent, abstract models provide an excellent base to address such requirements.

In the previous four editions of this workshop we have seen a lot of models, transformations and tools for model-driven user interface development. These approaches also tackled advanced user interface features like multi-platform and plasticity [6][7], context-sensitiveness [8], multimedia [9][10], 3D and augmented reality [11][12], ambient production environments [13], wearable sensors [14], interactive TV [15], and many others.

Motivation: Enhancing the User Experience

As mentioned above, the previous workshop editions have shown that concepts for model-driven user interface development are already becoming mature. However, the quality of the resulting user interfaces, in terms of the user experience, has always been one of the most difficult issues. In some cases, the user interfaces generated from the models can even improve the usability as they are for instance very consistent. However, it comes to a drawback if some parts of the user interface require individual design e.g. because of its complexity or to increase the likeability of the product.

For these reasons, an optimal development method should support both: systematic model-driven user interface development and individual design knowledge as usually resulting from manual, informal methods in user-centered design.

Workshop Goals

The current edition of this workshop aims to focus on challenges, opportunities, practical problems, and proposed solutions to integrate model-driven user interface development and informal design methods and tools. This includes topics like:

- Tools supporting model-driven development of advanced user interfaces, for instance visual tools or tools for non-experts in model-driven development.
- Models related to one or more HCI-aspects that positively affect the user experience of the modeled application.
- Project experience on user interface development using a model-driven development approach.
- Support for the combination of models and informal design knowledge in a model-driven development process.

To answer these questions, the workshop aims to bring together experts from all possible sides in user interface development, like experts in model-driven user interface development, experts in user-centered design, graphics designers, interaction designers, etc.

This workshop aims to foster interdisciplinary discussion between these different viewpoints. Therefore, we will build working groups consisting of people from different backgrounds. Depending on the individual background, we expect the following benefits for participants:

Participants with background in modeling will have the possibility to discuss with people from different backgrounds:

- Where do they see strengths and limitations of the model-driven approaches? Where is a more individual, manual design required?
- How can model-driven approaches be made more accessible? What could be promising tools?

Participants with a non-modeling background have the opportunity to discuss with modeling experts:

- What is the current state-of-the-art in model-driven user interface development? What do they cover?
- How could I adopt concepts from model-driven development? What are available tools? Where are the links to my area?

Finally, all participants together are encouraged to contribute to the central question:

How to better integrate model-driven approaches and informal methods from user-centered design?

Workshop Format

The workshop will start with a short introduction by the organizers, followed by short presentations by the participants about their tools, methods and experiences, and plenary discussion. The presentations and discussion will be held before lunch.

After lunch the participants will work in small (inter-disciplinary) groups of 4 to 5 persons around a focused topic. The results of this group work will be presented and discussed in a closing plenary session. Publication of these results as position papers in the proceedings will be considered.

References

[1] Tom Stahl, Markus Völter, Model-Driven Software Development, Wiley, 2006.

[2] Steven Kelly and Juha-Pekka Tolvanen, Domain-Specific Modeling: Enabling Full Code Generation, Kindle Book, 2008.

[3] P. A. Szekely, Retrospective and Challenges for Model-Based Interface Development, in F. Bodartand J. Vanderdonckt: DSV-IS'96, Springer, 1996, pp. 1–27.

[4] P. P. da Silva, User Interface Declarative Models and Development Environments: A Survey, in DSV-IS'00, 2000, pp. 207–226.

[5] Myers, B., Hudson, S. E., and Pausch, R., Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.* 7, 1, 2000, pp. 3-28.

[6] G. Botterweck, A Model-Driven Approach to the Engineering of Multiple User Interfaces, MDDAUI'06, CEUR Proceedings Vol. 214, 2006.

[7] J.-S. Sottet and G. Calvary and J.-M. Favre, Towards Model Driven Engineering of Plastic User Interfaces, MDDAUI'05, CEUR Proceedings Vol. 159, 2005.

[8] J. Van den Bergh and K. Coninx, Using UML 2.0 and Profiles for Modelling Context-Sensitive User Interfaces, MDDAUI'05, CEUR Proceedings Vol. 159, 2005.

[9] S. Sauer, M. Dürksen, A. Gebel, D. Hannwacker, GuiBuilder - A Tool for Model-Driven Development of Multimedia User Interfaces, MDDAUI'06, CEUR Proceedings Vol. 214, 2006.

[10] A. Pleuss, MML: A Language for Modeling Interactive Multimedia Applications, ISM 2005, IEEE Society Press, 2005.

[11] J. De Boeck, J. M. Gonzalez, K. Coninx, J. Vanderdonckt, Open Issues for the development of 3D Multimodal Applications from an MDE perspective, MDDAUI'06, CEUR Proceedings Vol. 214, 2006.

[12] G. Gauffre, E. Dubois, R. Bastide, Domain Specific Methods and Tools for the Design of Advanced Interactive Techniques, User Interfaces, MDDAUI'07, CEUR Proceedings Vol. 297, 2007.

[13] K. Breiner, O. Maschino, D. Goerlich, G. Meixner, Towards automatically interfacing application services integrated in a automated model-based user interface generation process, MDDAUI'09, CEUR Proceedings Vol. 439, 2009.

[14] F. Balagtas-Fernandez, H. Hussmann, Modeling Information from Wearable Sensors, MDDAUI'09, CEUR Proceedings Vol. 439, 2009.

[15] F. Paternò, S. Sansone, Model-based Generation of Interactive Digital TV Applications, MDDAUI'06, CEUR Proceedings Vol. 214, 2006.