Fourth International Workshop on Model Driven Development of Advanced User Interfaces

Gerrit Meixner, Daniel Görlich
German Research Center for Artificial Intelligence (DFKI),
67663 Kaiserslautern, Germany
{Gerrit.Meixner, Daniel.Goerlich}@dfki.de

Kai Breiner
Fraunhofer Institute for Experimental Software Engineering (IESE),
67663 Kaiserslautern, Germany
breiner@cs.uni-kl.de

Heinrich Hußmann, Andreas Pleuß
Universität München,
LFE Medieninformatik
80333 München, Germany
{heinrich.hussmann, andreas.pleuss}@ifi.lmu.de

Stefan Sauer
University of Paderborn,
Faculty of Computer Science
33098 Paderborn, Germany
sauer@upb.de

Jan Van den Bergh
Expertise Centre for Digital Media (EDM)
Universiteit Hasselt, Campus Diepenbeek
3590 Diepenbeek, Belgium
jan.vandenbergh@uhasselt.be

ABSTRACT

Model Driven Development (MDD) is an important paradigm in Software Engineering. In MDD, applications are specified systematically using abstract, platform independent models. The models are then transformed into executable code for different platforms and target devices. Model-driven techniques become ever more prominent in any kind of application, such as multimedia and Web, ubiquitous and automotive applications.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces.

General terms: Design, Human Factors, Languages

Keywords: HCI, MBUID, MDD, Task Model

INTRODUCTION

The workshop will be a platform for discussing the modeling of advanced user interfaces, such as interfaces supporting complex interactions, visualizations, multimedia representations, multimodality, adaptability or customization. It will contribute to a better integration of knowledge from the Human-Computer Interaction community and the Software Engineering community. Guiding principle is the demand for a flexible composition of various different models to support the model-driven development of user interfaces with a high degree of usability and customization.

We solicit papers addressing one or more of these issues:

- Models required for modeling (specific aspects of) advanced or non-standard user interfaces clearly stating their added value for the targeted applications compared to the relevant models discussed in literature.
- Adaptation and customization mechanisms for model transformations leading to tailored user interfaces with a high degree of usability.
- Integration of informal techniques and tools from traditional UI design, i.e. by transformations from/to (non-trivial) UI design tools and different kinds of prototypes.
- Project experience on user interface development using a model-driven development approach.
- Problems and requirements on model-driven engineering emerging from the application area of model-driven user interface development.

Submitted papers to our workshop addressed the following major topics:

IUI'09, February 8–11, 2009, Sanibel Island, Florida, USA.
ACM 978-1-60558-331-0/09/02.

- Automatically generating user interfaces out of abstract descriptions, while addressing various challenges in this field.

- Models which form the requirements for user interfaces also serve in some cases as a basis of a possible generation processes.

- Development and Experience of supporting tools and frameworks in the field of model driven user interface development.

In the following we give a short abstract about some of the submitted papers.

GENERATING USER INTERFACES

The transformation driven architecture (TDA) is a system building (in particular, tool building) approach that is based on model transformations, interface meta-models with corresponding engines, and event/command mechanism. The paper describes a meta-model and the corresponding engine for graph diagram presentations within TDA. The facilities of the meta-model and the engine include static diagram presentations, as well as graph diagram animations.

One of the issues involved in automatic generation of user interfaces is the presentation of content from the domain of discourse according to its purpose in the current state of the human-machine dialogue. The paper addresses this issue through including the intention of a given content presentation using communicative acts, and through generating it specifically according to the type of communicative act, that indicates the intention.

A position paper presents a view on model driven user interface development and relates it to other approaches. The methodology is based on task models that are attributed and merged with a navigational model to derive user interface models. A toolset to support this development approach is introduced which is well-integrated and itself based on the eclipse modeling framework EMF.

Ambient Intelligence or Ubiquitous Computing confronts the production industry with a new diversity of usage situations. In previous work, the authors have shown the adaptation of a task-oriented, model-based Useware engineering process to future paradigms by extending existing models and shifting the development/generation of the user interface (UI) from development time to run-time. When separating the UI design from the application engineering process, the problem of the generated UI interfacing the according service functions emerged, this is addressed in this paper.

MODELS AND REQUIREMENTS

Developers are enforced to use user-centered development processes to guarantee a high usability for their product. The Useware Markup Language (useML) 2.0 is a user-centered task-oriented modeling language, which is used in the Useware-Engineering process for developing intelligent user interfaces. With version 2.0 some major changes have been made to increase the expressiveness of useML, i.e. with adding temporal operators the language has been equipped for semi-automatic dialog model generation. To support developers, an intuitively, graphical useML-Editor has been developed. The paper introduces the Useware-Engineering process, the changes in useML 2.0 and the useML-Editor (Udit).

The goal of the authors' research is to create a modeling environment for non-expert users to allow them to easily create applications for mobile health monitoring. In the paper, they focus on the design and representation of the modeling constructs that represent information taken from wearable sensors. They want to find a way to represent sensor information such that it is not confusing and is easily recognizable by the non-expert user.

Smart environments bring together multiple users, (interaction) resources and services, resulting in complex and unpredictable interactive computing environments in which users thus have difficulties to build up their mental model of such interactive systems. To address this issue users need possibilities to evaluate the state of these systems and to adapt them according to their needs. In this work the authors describe the requirements and functionalities for evaluating and controlling interactive spaces in smart environments from the system and the user perspective.

Separating requirements from user interface specifications often leads to unusable systems or to systems that do not support the users' needs. The authors address this issue by introducing explicit transformations between models of these different "worlds". In fact, they show how to transform artifacts in a model specifying requirements to artifacts in a model specifying a user interface, and vice versa. From the perspective of model-driven transformations, this means inverse transformations. And it means transformations on the same abstraction level. They also transform from a more abstract to a more concrete model of a user interface, and vice versa. In effect, this allows starting either from requirements and getting support for transforming to a user interface, or from a user interface prototype and getting support for transforming to requirements.

DEVELOPMENT AND EXPERIENCE

The authors describe an approach and an environment for setting up Wizard of Oz experiments to test multimodal interaction (through mobile devices) for multimedia content delivery. This is based on a meta-model and a reference client-server architecture, whose implementation is discussed. The approach allows progressive refinement of the Wizard of Oz environment into a deployed application, through the use of a single meta-model, to be progressively specialized, and a service-oriented approach which allows the substitution of implemented services to simulated ones. A visual interface is provided for the definition of the wizard interface, of the metadata relating content delivery and possible interactions on it, and of some aspects of the client interaction.