Model-Driven Development of Platform-Independent Mobile Applications Supporting Role-based App Variability*

Steffen Vaupel[1], Gabriele Taentzer[1], René Gerlach[2], Michael Guckert[2]

Abstract: The use of mobile applications has become an indispensable part of human interaction and especially of urban life. This will lead to rapidly increasing numbers of applications and users that make the development of mobile applications to one of the most promising fields in software engineering. Due to short time-to-market, differing platforms and fast emerging technologies, mobile application development faces typical challenges where model-driven development (MDD) can help. We present a modeling language and an infrastructure for the model-driven development of native apps in Android and iOS. Our approach allows flexible app development on different abstraction levels: compact modeling of standard app elements such as standard data management and increasingly detailed modeling of individual elements to cover specific behavior. Moreover, a kind of variability modeling is supported such that apps variants for a range of user roles can be developed. Several apps including a mobile learning app, a conference app, and a museum guide with augmented reality functionality demonstrate the usefulness of our approach.

Keywords: mobile application, model-driven software development, variability

1 Introduction

An infrastructure for model-driven development (MDD) has a high potential for accelerating the development of software applications. While just modeling the application-specific data structures, processes and layouts, runnable software systems can be generated. Hence, MDD does not concentrate on technical details but lifts software development to a higher abstraction level. The heart and soul of MDD is the domain-specific modeling language. It comes along with a tool environment consisting of textual or visual model editors and appropriate code generators for the desired target platforms (as, e.g., Android and iOS). For the development of our MDD infrastructure, we have chosen an agile *bottom-up* process [VSRT15], starting with a domain analysis and feature identification of mobile applications, template extraction from re-implemented prototypes, and iterative language extension.

2 Domain and approach

Mobile apps are developed for diverse purposes – from mere entertainment to serious business applications. While focusing mainly on data-oriented business apps, our approach allows to enrich them by entertainment and educational elements, or sensor and external hardware access. A particular case is using the built-in camera to recognize objects and

*This work was partially funded by LOEWE HA (State Offensive for the Development of Scientific and Economic Excellence) project no. 355/12-45: PIMAR – Platform Independent Mobile Augmented Reality.

[1] Philipps-Universität Marburg, Hans Meerwein Straße 1, 35032 Marburg, Germany, {svaupel, taentzer}@informatik.uni-marburg.de

[2] KITE - Kompetenzzentrum für Informationstechnologie, Technische Hochschule Mittelhessen, Wilhelm-Leuschner-Straße 13, 61169 Friedberg, Germany, {rene.gerlach, michael.guckert}@mnd.thm.de

augment the live view with different types of virtual objects, which is called augmented reality (AR) [GMG+15]. This feature is useful for industry applications as well as for apps in education and tourism sectors.

Although there are already approaches to model-driven development of mobile apps such as MD^2 [HMK13], our contribution differs considerably in design and purpose of the language. Our approach focuses on data-driven apps with role-based variants (Figure 1). The entire approach has three user roles: *app developers* who create the application, *providing users* who may configure the application, and finally *end users* of the app.

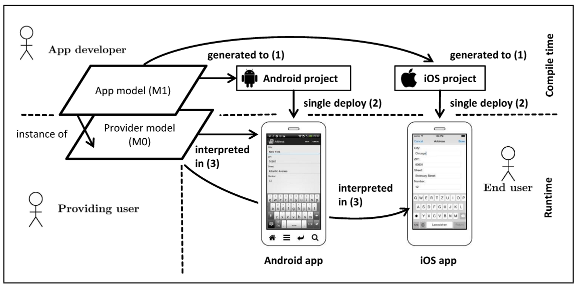

Fig. 1: Cross-platform generation of role-based apps

3 Modeling language, MDD infrastructure, and case studies

The general approach to the modeling language is component-based. An *app model* consists of a *data model* defining the underlying class structure, a *GUI model* containing the definition of pages and style settings for the graphical user interface, and a *process model* which defines the behavioral facilities of an app in the form of processes and tasks. *Provider models* are instances of app models.

The Eclipse-based MDD infrastructure provides a visual model editor (including validation rules) and contains two code generators. In addition to the work presented in [VTH+14] and [PIMAR], the MDD infrastructure has been evaluated at five differently focused case studies showing the applicability and usefulness of the approach. Team members and students have created a *conference app* for the MoDELS' 14 conference with conference organizers and participants as user roles, a *learning app* with teachers and learners as user roles, a *museum guide* including AR-functionality with museum providers and visitors as user roles, a control app for power sockets (*SmartPlug*), and a *TV-Reminder*.

References

[GMG+15] Guckert, Michael; Malerczyk, Cornelius; Gerlach, René; Taentzer, Gabriele; Vaupel, Steffen; Fatum, Michael: Plattformunabhängige Entwicklung mobiler Anwendungen mit Augmented Reality-Funktionalität. Anwendungen und Konzepte der Wirtschaftsinformatik, (3):5, 2015.

[HMK13] Heitkötter, Henning; Majchrzak, Tim A.; Kuchen, Herbert: Cross-Platform Model-Driven Development of Mobile Applications with MD^2. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, pp. 526 – 533, 2013.

[PIMAR] PIMAR: Platform-Inpendent Development of Mobile Apps with Augmented Reality. http://www.uni-marburg.de/fb12/swt/forschung/software/pimar/, 2015.

[VTH+14] Vaupel, Steffen; Taentzer, Gabriele; Harries, Jan Peer; Stroh, Raphael; Gerlach, René; Guckert, Michael: Model-driven development of mobile applications allowing role-driven variants. In: Model-Driven Engineering Languages and Systems, pp. 1 – 17, LNCS 8767. Springer, 2014.

[VSRT15] Vaupel, Steffen; Strüber, Daniel; Rieger, Felix; Taentzer, Gabriele: Agile bottom-up development of domain-specific IDEs for model-driven development. In: Proceedings of FlexMDE 2015: Workshop on Flexible Model-Driven Engineering, pp. 12 – 21, Vol. 1470. CEUR-WS.org, 2015.