Chapter 12
Model-Driven Development of Data-Centered Mobile Applications: A Case Study for Android

Jorge Amadeu Alves Pereira da Silva
Instituto Politécnico de Viana do Castelo, Portugal

Sara Paiva
Instituto Politécnico de Viana do Castelo, Portugal

Antonio Miguel Rosado da Cruz
Instituto Politécnico de Viana do Castelo, Portugal

ABSTRACT
Increasing adoption of mobile smart devices demands a growing number of mobile applications (apps). Each of these applications must often be deployed to different mobile platforms, such as Android, iOS or Windows. Many of these applications are data-oriented, enabling the user to manage information, by creating, updating, deleting and retrieving data on his smart mobile device. By using a model-driven development approach, it is possible to generate a platform independent user interface model from a domain model, which represents the information structure of the application domain, and then have different code generators for each different target platform. This chapter presents such an approach together with a case study for Android apps.

INTRODUCTION
The increasing adoption of mobile smart devices, throughout the world, is demanding a growing number of mobile applications (apps). Nowadays, people use a variety of these applications for daily activities such as getting driving directions, finding a pharmacy or checking the weather. Other applications exist that allow the user to create from scratch his personal information such as daily notes, book/cd inventory, and grocery shopping list, amongst others. A common feature of these last applications is the need to
perform simple operations on data, such as creating, retrieving, updating or deleting data, making them fall under the umbrella of data oriented applications. Through generalization and parameterization of these common features, the development process can be streamlined and automated leading to a less time consuming and less costly apps development process.

There are currently a variety of tools for mobile application development. The decision to use a native or a cross-platform tool must take into account the benefits and disadvantages of both approaches. Native apps provide a more fluid and responsive user interface, a better performance, the ability to use all platform functionalities and a simpler distribution process. Additionally, there are less security issues in native apps, support for developers is more easily found, the design creation is simpler and the performance is better, as it is supported by the operation system itself. On the other hand, the costs are reduced when cross-platform tools are used, since cross-compilation allows the same sources to generate outputs for different platforms avoiding the need to have specialized teams in each platform intended to target. For this reason, cross-platforms’ development presents a high rate of code reusability. Finally, it is generally easier to find web developers than mobile devices’ native programmers, making native skills usually be more expensive (Madaudo & Scandurra, 2013).

Model driven development (MDD) approaches, like Domain Specific Modeling (DSM) (Kelly & Tolvanen, 2008), or the OMG’s Model Driven Architecture (MDA) (Warmer et al., 2003), are based on the successive refinement of models and on the automatic generation of code and other sub-models, allowing the software engineers and developers to focus on platform independent modeling activities rather than programming activities. This allows them to focus on concepts of the problem domain, and the way they shall be modeled in order to produce a software solution, rather than being diverted by technical issues of the solution domain. Within an MDD setting, code can be automatically generated to a great extension, dramatically reducing the most costly and error prone aspects of software development (Frankel, 2003).

In the next section, an overview of MDD techniques and of the characteristics of Android native applications is given. Afterward, an overview of existing approaches to the model-driven development of mobile apps is given.

The main focus of the chapter lies, then, on presenting a model-driven approach for generating Android native applications. Then, a discussion on future and emerging trends and some recommendations on future research directions are made. Finally, some concluding remarks are presented.

BACKGROUND

Model-Driven Development

Introduction

Model-driven development (MDD), as suggested by its name, is a software development paradigm driven by the activity of modeling software (Warmer et al., 2003), in which models are first class citizens, being code pushed to a second plan.

MDD approaches software development by constructing models that may be refined (transformed) through different levels of abstraction, from a platform independent level, or a computation independent level, to a platform specific model that is directly mapped to final code (see figure 1).
Related Content

Novel Taxonomy to Select Fog Products and Challenges Faced in Fog Environments

Predictive Modeling for Imbalanced Big Data in SAS Enterprise Miner and R
[www.igi-global.com/article/predictive-modeling-for-imbalanced-big-data-in-sas-enterprise-miner-and-r/210567?camid=4v1a](www.igi-global.com/article/predictive-modeling-for-imbalanced-big-data-in-sas-enterprise-miner-and-r/210567?camid=4v1a)

Leveraging the Internet of Things (IoT) Paradigm Towards Smarter Applications
[www.igi-global.com/chapter/leveraging-the-internet-of-things-iot-paradigm-towards-smarter-applications/230644?camid=4v1a](www.igi-global.com/chapter/leveraging-the-internet-of-things-iot-paradigm-towards-smarter-applications/230644?camid=4v1a)

Planning and Implementation of Cloud Computing in NIT's in India: Special Reference to VNIT