Chapter 13

Migrating JAVA to Mobile Platforms through HAXE: An MDD Approach

Pablo Nicolás Díaz Bilotto
Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

Liliana Favre
Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina

ABSTRACT

Software developers face several challenges in deploying mobile applications. One of them is the high cost and technical complexity of targeting development to a wide spectrum of platforms. The chapter proposes to combine techniques based on MDA (Model Driven Architecture) with the HaXe language. The outstanding ideas behind MDA are separating the specification of the system functionality from its implementation on specific platforms, managing the software evolution, increasing the degree of automation of model transformations, and achieving interoperability with multiple platforms. On the other hand, HaXe is a very modern high level programming language that allows us to generate mobile applications that target all major mobile platforms. The main contributions of this chapter are the definition of a HaXe metamodel, the specification of a model-to-model transformation between Java and HaXe and, the definition of an MDA migration process from Java to mobile platforms.

INTRODUCTION

Nowadays mobile devices come with their users all the time and everywhere. Among other novel features, mobile devices contain global positioning sensors, wireless connectivity, built-in web browsers and photo/video/voice capabilities that allow providing highly localized, context aware applications. Mobile phones have become as powerful as any desktop computer in terms of applications they can run, however, the software development in mobile computing is still not as mature as it is for desktop computer and the whole potential of mobile devices is wasted (Waserman, 2010).
Although mobile technologies present new opportunities for services and businesses, they also present development and implementation challenges. Some mobile applications must also determine the user location before offering the service and then track the location to adapt services and information accordingly. Besides, an additional challenge is to achieve the required level of security, reliability and quality of mobile applications. Various authors describe challenges of mobile software development, for example, in (Dehlinger & Dixon, 2011) authors highlight creating user interfaces for different kinds of mobile devices, providing reusable applications across multiple mobile platforms, designing context aware applications and handling their complexity and, specifying requirements uncertainty. Thompson et al. (2011) remark issues related to ensuring that the application provides sufficient performance while maximizing battery life.

A current problem in the engineering community is the rapid proliferation of mobile platforms (Lettner, Tschernuth & Mayrhofer, 2011). The high cost and technical complexity of targeting development to a wide spectrum of platforms, has forced developers to make applications tailored for each type of device. Within mobile development, many companies have different development teams redoubling the software engineering efforts for functionally similar mobile applications. In many cases, developers prefer to implement an application once and deploy it to different platforms with minimal effort. In this direction, an open source multiplatform programming language called HaXe has emerged integrating the native behaviors of the different platforms targeted in development projects (Dasnois, 2011).

To manage a huge diversity of technologies, mobile development needs novel technical frameworks for information integration and tool interoperability. Some works propose to exploit the MDD (Model Driven Development) paradigm to simplify multi-device development (Brambilla, Cabot, & Wimmer, 2012) (Dunkel & Bruns, 2007). A specific realization of MDD proposed by the Object Management Group (OMG) is MDA (Model Driven Architecture) (OMG MDA, 2014).

The outstanding ideas behind MDA are separating the specification of the system functionality from its implementation on specific platforms, managing the software evolution from abstract models to implementations, increasing the degree of automation of model transformations, and achieving interoperability with multiple platforms. Models play a major role in MDA, which distinguishes at least Platform Independent Model (PIM) and Platform Specific Model (PSM). An MDA forward engineering process focuses on the automatic transformation of different models that conform to MOF metamodels. The essence of MDA is the Meta Object Facility Metamodel (MOF) that allows integrating different kinds of software artifacts (MOF, 2008) (MOF, 2011). The MOF 2.0 Query, View, Transformation (QVT) metamodel (QVT, 2012) allows expressing model transformations.

The OMG ADM Task Force (ADMTF) has defined a set of metamodels aligned with MOF that allow describing various aspects of the software modernization (ADM, 2015). Metamodels such as Knowledge Discovery Metamodel (KDM) and Abstract Syntax Tree Metamodel (ASTM) aim at improving the process of understanding and evolving software applications and enabling architecture-driven reverse engineering (KDM, 2011) (ASTM, 2011).

CASE tools based on MDA do not support forward engineering processes for HaXe (CASE MDA, 2015). We consider beneficial to integrate HaXe with MDA. The first step in the integration direction is to have a definition of HaXe aligned with the MDA standards, particularly MOF. Then, we defined a MOF metamodel for HaXe, which allows specifying meta-level transformations from transformation languages such as QVT or ATL (Atlas Transformation Language) (Jouault & Kurtev, 2005) (Jouault et al, 2008) (Jouault et al., 2006). Specifically, this chapter will include a description of an MDA migration process from object-oriented code (Java in particular) to different mobile platforms. The main steps of