Chapter 14
AlexandRIA: A Visual Tool for Generating Multi-Device RIAs

ABSTRACT

Model-Driven Development (MDD) tools for Rich Internet Applications (RIAs) development are focused on software modeling, and they leave automatic code generation in a second term. On the other hand, Rapid Application Development (RAD) tools for RIAs development enable developers to save development time and effort by leveraging reusable software components. AlexandRIA is a RAD tool that allows developers to automatically generate both source and native code of multi-device RIAs from a set of preferences selected throughout a wizard following the phases of a User Interface (UI) pattern-based code generation approach for multi-device RIAs. In this chapter, the use of the UI design process behind AlexandRIA is demonstrated by means of a sample development scenario addressing the development of a cloud services Application Programming Interfaces (APIs)-based cross-platform mobile RIA. This scenario is further revisited in a case study that addresses the automatic generation of an equivalent application using AlexandRIA.

1. INTRODUCTION

Nowadays, developers have the need of specifying the features and functionalities of RIAs (Rich Internet Applications) with legacy requirements to reduce development efforts and ensure less error-prone applications. Here, the automation tools for software development activities have become a major feature of RIAs development processes. From this perspective, most of the automation tools for software development are based on MDE (Model-driven Engineering) approaches such as RUX-Tool (Linaje et al., 2007), which is a software tool that automates the Rich User eXperience Method (RUX-Method) allowing engineering the adaptation of legacy model-based Web 1.0 applications to Web 2.0 GUIs. In a nutshell, RUX-Tool is focused on software modeling leaving the automatic code generation in a second term.

Moreover, there is another approach which is focused on automatic code generation through settings establishment: Adobe™ AIR™ Launchpad. It allows generating ready-to-compile source code
of AIR™-based applications i.e. desktop applications and applications for Android™, Apple™ iOS and BlackBerry™ Tablet OS platforms deployed on the Adobe™ AIR™ runtime. Nevertheless, Adobe™ AIR™ Launchpad is not based on a RIAs development process; besides, it does not entirely cover the generation of multi-device RIAs because it does not consider Web browser-based RIAs.

The aforementioned drawbacks are improved by AlexandRIA. AlexandRIA is a code generation software tool that allows developers to generate both source and native code of cloud services APIs-based multi-device RIAs from a set of preferences selected throughout a wizard following the phases of a UI pattern-based code generation approach for multi-device RIAs. This code generation approach is mainly focused on RIA’s UI (User Interface) details; nevertheless, it incorporates high-level abstractions for distributing business logic operations between client and server as well as for defining advanced client-server communication mechanisms.

AlexandRIA agrees with the RAD (Rapid Applications Development) tools philosophy in the sense that it uses a set of reusable software components (Fraternali, 1999). These components encapsulate functionalities provided by several cloud services APIs (Application Programming Interfaces) such as Twitter™ REST, Flickr™ and Google™ Custom Search APIs, to mention but a few.

In addition, AlexandRIA is focused on generating multi-device RIAs. Here, the term multi-device refers to the ability of RIAs to deploy consistently on different Web browsers, desktop operating systems and mobile platforms as Web, desktop and mobile applications, respectively, starting from the same code base. It is important to notice that, this term is closely related to the plugin-based RIAs such as the Adobe™ Flex™ and Microsoft™ Silverlight™-based RIAs.

In this chapter, the use of the UI design process proposed as part of the code generation approach implemented by AlexandRIA is demonstrated within the constraints of AlexandRIA by means of a sample development scenario addressing the development of a cloud services APIs-based native mobile RIA for different mobile devices. This scenario is further revisited in a case study addressing the automatic generation of an equivalent application by using AlexandRIA.

2. ALEXANDRIA: A UI PATTERN-BASED APPROACH FOR GENERATING MULTI-DEVICE RIAS

AlexandRIA automates a UI pattern-based code generation approach for multi-device RIAs which is intended to address the following drawbacks of current proposals on RIAs engineering such as RUX-Method (Linaje et al., 2007), PPRD (Martínez-Nieves et al., 2010) and UWE-R (Machado et al., 2009): they do not entirely address multi-device RIAs; therefore, they do not cover the development of multi-device RIAs in an automatic or semi-automatic way.

As a proof of concept, Adobe™ Flex™ 4.5 and PhoneGap™ 1.0 were selected as the cross-platform development frameworks for implementing the aforementioned code generation approach. Thereby, AlexandRIA is currently based on both ActionScript and JavaScript technologies. Finally, it is important to notice that the code generation algorithm implemented by AlexandRIA is a domain-specific approach focused on generating cloud services APIs-based multi-device RIAs, i.e., the functionalities of the applications to be generated by AlexandRIA are implemented in terms of data and operations from cloud services APIs as is widely explained in section 14.3 of this chapter.

Although this code generation approach is mainly focused on RIA’s UI details, it incorporates high-level abstractions for distributing business logic operations between client and server as well as for defining advanced client-server communication mechanisms. The cloud services APIs operations are out of the scope of the business