ADM-Based Migration from JAVA Swing to RIA Applications

Samir Mbarki, Department of Computer Science, Ibn Tofail University, Kenitra, Morocco
Naziha Laaz, Department of Computer Science, Ibn Tofail University, Kenitra, Morocco
Sara Gotti, Department of Computer Science, Ibn Tofail University, Kenitra, Morocco
Zineb Gotti, Department of Computer Science, Ibn Tofail University, Kenitra, Morocco

ABSTRACT

Companies are investing a lot of resources and effort for migrating their legacy applications and adapting them with the rapid technological changes. For this reason, the authors are interested in the modernization of desktop applications developed in Java Swing to Web 2.0 applications. Therefore, an ADM approach is applied in order to develop a tool named FlexMigration allowing automatic reverse engineering of Swing GUI to obtain a RIA GUI. The usefulness of this tool is the automation of the migration process with the extraction of the actions encapsulated in possible anonymous classes. As an illustration, they present along this paper a reengineering of a small legacy chat application. The authors explain its migration process to generate a similar Flex Graphical User Interface.

KEYWORDS

Abstract Syntax Tree Meta-Model (ASTM), Anonymous Inner Class, Architecture-Driven Modernization (ADM), Graphical User Interface Meta-Model (GUIM), Java Development Tool (JDT), Knowledge Discovery Model (KDM), Model Driven Engineering (MDE), Reverse Engineering, Rich Internet Application (RIA)

1. INTRODUCTION

In recent years, the importance of intangibles in the process of value creation has emerged with ever greater emphasis. Intangibles, as a resource without physical substance, are a key driver underpinning business competitiveness

Since 1950, software systems have become an absolute necessity for companies. During each decade, new technologies appear according to different paradigms. Hence, many programming approaches have been established (e.g. functional languages, procedural languages, 4G languages, object languages, etc.). Furthermore, modeling languages were subject to the same evolution as programming languages. They have become more powerful in expressing requirements at a high abstraction level. No one denies that software engineering focusing on models is the future.

Consequently, it is essential to learn good modeling practices to capitalize on the advantages of the models fully in order to track changes in technology and evolve to better software development practices using MDE. The Object Management Group introduced the MDA approach in 2000 (Blanc, 2004). The ADM was defined later, in 2007 (http://adm.omg.org). The main goal of the ADM is to provide standards in the field of reverse engineering. These standards are based on models and meta-models for modernizing legacy software systems. To meet the modernization requirements, the ADM defines two main meta-models: the ASTM and the Knowledge KDM (http://www.omgwiki.org/admtf).
In the present work, we conduct a modernization process to migrate from a Java Swing GUI to a similar RIA GUI. The reengineering process is divided into three steps: the reverse engineering, the restructuring and the forward engineering (Chikofsky, 1990). This paper presents the FlexMigration tool which is developed to modernize legacy Swing GUI’s. FlexMigration is generic and extensible since it allows the extraction of knowledge about GUI components, GUI components inter-relationships and application business aspects. This knowledge is represented in a KDM Platform Independent Model. The first step in our case is the parsing of a Java application. As output, we obtain two models: AST and GUI models. Our parsing algorithm uses JDT classes. In many Java applications, the modularity and reusability provided by the object oriented paradigm is not respected. In this case, the Java chat application is reduced to one class. All the actions are defined in anonymous inner classes. Our parser has to identify these anonymous classes in the legacy application and represent them in the AST model.

The rest of this paper is organized as follows: Section 2 is dedicated to the related work. In section 3, we explain the reverse engineering phase in detail. Section 4 is dedicated to the restructuring phase. Section 5 describes the forward engineering phase. Finally, section 6 concludes the work and presents perspectives.

2. RELATED WORK

At the present time, software modernization is a specific kind of evolutionary maintenance paradigm to solve reengineering problems (Silva et al., 2006). The advent of ADM approach allows the growth and expansion of the evolutionary maintenance. So many researches, both on model driven engineering and software modernization, have been conducted in recent years. The most relevant are (Mbarki & Erramdani, 2008; Mbarki & Erramdani, 2009; Pérez-Castillo et al., 2013; Rahmouni & Mbarki, 2014; Ramón et al., 2013; Rodríguez-Echeverría et al., 2011):

In (Mbarki & Erramdani, 2008; Mbarki & Erramdani, 2009), the authors have designed a MVC2 Web meta-model which allows them to elaborate model-to-model transformations from UML class diagram. In the second work, the authors have set up two transformations; the first one was a PIM to PIM transformation, allowing the refinement of the analysis-level class diagram. The second one was a PIM to PSM transformation, specific to a MVC2 Web application. The transformation algorithm defined permits to iterate on the design-level class diagram and generates an XML file containing all actions, forms and forwards to JSP pages that can be used to generate the required code of the target application. These works have been recently extended to cover N-tiers applications (Rahmouni & Mbarki, 2014).

The work proposed by Roberto et al, respects the Model-Driven Modernization principles and uses its standards. It focuses on the modernization of the JavaEE applications. Roberto et al’ main contribution is the definition of a systematic process for WA-to-RIA modernization by applying MDE principles, techniques and tools. The goal of their modernization process consists in generating a RIA client from the legacy WA presentation and navigation layers and its corresponding service-oriented connection layer with the underlying business logic at server side (Rodríguez-Echeverría et al., 2011).

In (Pérez-Castillo et al., 2013), the authors presented a tool for the reverse engineering called ANDRIU. This tool uses KDM standard for migrating applications to the android platform while it focuses on the migration of Java/swing applications to rich interfaces. In the reverse engineering phase, the authors extract the java Swing code in a specific AST model.

Ramón Sánchez et al. present a solution for positioning UI components because they have targeted migration of RAD applications. This platform does not define layout managers, and widgets
Requirements Engineering for Cloud Application Development
www.igi-global.com/chapter/requirements-engineering-cloud-application-development/70138?camid=4v1a

An Information Services Inspired Framework for Assessing Scientific Book Quality
www.igi-global.com/article/information-services-inspired-framework-assessing/68792?camid=4v1a