

Chapter 5

An Overview of RIAs Development Tools

Giner Alor-Hernández

Instituto Tecnológico de Orizaba, Mexico

Viviana Yarel Rosales-Morales

Instituto Tecnológico de Orizaba, Mexico

Luis Omar Colombo-Mendoza

Instituto Tecnológico de Orizaba, Mexico

ABSTRACT

Rich Internet Applications (RIAs) development has traditionally been addressed using framework-based development approaches (i.e., using application frameworks), which usually comprise tools such as Standard Development Kits (SDKs), class libraries, and Integrated Development Environments (IDEs). Nevertheless, another development approach that relies on Model-Driven Development (MDD) methodologies and tools has recently emerged as a result of the academic and commercial effort for alleviating the lack of development methodologies and support tools especially designed for the development of RIAs. In this chapter, a new classification of RIAs development approaches is proposed by introducing a third category: Rapid Application Development (RAD) approaches. Thereby, the chapter reviews not only IDEs for framework-based RIA development; it also addresses other support tools for MDD and RAD such as code generation tools. Additionally, the features, scope, and limitations of the analyzed tools are discussed by means of a series of usage scenarios addressing the RIAs implementation.

1. INTRODUCTION

RIAs engineering is an emerging area of the Software engineering which lacks of development approaches and software tools in comparison with Web engineering. Therefore, the development of RIAs is mainly driven by a set of programming languages such as ActionScript and JavaScript as well as application frameworks such as Adobe™ Flex™ and Microsoft™ Silverlight™.

DOI: 10.4018/978-1-5225-3422-8.ch005

However, some proposals based on the MDD (Model Driven Development) approach have recently emerged with the aim of solving the aforementioned requirements. In fact, according to Toffeti, Comai, Preciado and Linaje (Toffeti et al., 2011), RIAs development is currently addressed 1) by using application frameworks which provide SDKs (Standard Development Kits), class libraries, IDEs (Integrated Development Environments), among other development tools, 2) by using MDD tools which generate executable code starting from high-level software models.

The use of software tools based on the RAD (Rapid Application Development) methodology is a current trend on RIAs development. Thus, the aforementioned classification has been extended by adding a category for clustering approaches based on the RAD methodology. In this chapter, the most representative software tools in each category of the extended classification are described.

Additionally, the features, scope and limitations of the tools are discussed by means of a series of usage scenarios addressing the implementation of RIAs. In detail, these usage scenarios face some of the RIA development aspects identified by Toffetti (Toffetti et al., 2011). It is important to notice that, the usage scenarios are not achieved as use cases but as technical discussions summarizing the lessons learned in the development of sample RIAs by using the tools described in this chapter; therefore, technical details about implementation are not included. In most cases, these usage scenarios involve not only coding but also building activities, namely compiling and debugging, so that they are not completely accurate neither for the UI (User Interface) design tools nor for the RAD and MDD tools but for the IDEs described in this chapter. As far as possible, the execution of the resulting applications in debug mode is depicted by some screenshots throughout this chapter.

2. TOOLS FOR FRAMEWORK-BASED RIAs DEVELOPMENT

This section presents the most popular software tools available for developing RIAs based on the four major technologies for RIAs development. It is important to notice that most of the software tools considered in this section are Integrated Development Environments (IDEs). An IDE is a software application that provides comprehensive facilities to developers for software development and ideally integrates source code editing, visual designing, debugging and building capabilities. These features are considered for describing some IDEs in following subsections.

IDEs for RIAs development can be classified into the following two major groups accordingly to its architecture: 1) standalone applications and 2) plug-in applications. Besides, a standalone application, a plug-in application is a set of software components that adds specific functionalities to a larger software application. From this perspective, it is important to notice that there are popular IDEs based on plug-in architectures such as Eclipse™, which can be extended for supporting many programming languages and frameworks through plug-ins. Therefore, there are many IDEs based on Eclipse™ available as both standalone and plug-in applications. For practical purposes, the aforementioned classification is used in this section. Furthermore, it is important to notice that besides the official development tools, i.e., the tools developed, supplied and maintained by technology owners, there are third-party development tools, i.e. tools developed, supplied and maintained by third-party open source communities or external software companies.

32 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/an-overview-of-rias-development-tools/188204

Related Content

Quality, Improvement and Measurements in High Risk Software

Edgardo Palza Vargas (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 733-748).

www.irma-international.org/chapter/quality-improvement-measurements-high-risk/77730

Certifying Properties of Programs Using Theorem provers

J. Santiago Jorge, Victor M. Gulias and David Cabrero (2007). *Verification, Validation and Testing in Software Engineering* (pp. 252-301).

www.irma-international.org/chapter/certifying-properties-programs-using-theorem/30754

Attribute Decoration of Attack–Defense Trees

Alessandra Bagnato, Barbara Kordy, Per Håkon Meland and Patrick Schweitzer (2012). *International Journal of Secure Software Engineering* (pp. 1-35).

www.irma-international.org/article/attribute-decoration-attack-defense-trees/66406

Generation of Concurrency Control Program by Extending Functions in Genetic Programming

Teruhisa Hochin, Tatsuya Saigo, Shinji Tamura and Hiroki Nomiya (2014). *International Journal of Software Innovation* (pp. 13-27).

www.irma-international.org/article/generation-of-concurrency-control-program-by-extending-functions-in-genetic-programming/120516

Performance Evaluation of SHA-2 Standard vs. SHA-3 Finalists on Two Freescale Platforms

Pal-Stefan Murvay and Bogdan Groza (2013). *International Journal of Secure Software Engineering* (pp. 1-24).

www.irma-international.org/article/performance-evaluation-of-sha-2-standard-vs-sha-3-finalists-on-two-freescale-platforms/101890