ROMULUS: Domain Driven Design and Mashup Oriented Development based on Open Source Java Metaframework for Pragmatic, Reliable and Secure Web Development

Boni García, Juan C. Dueñas, Jose I. Fernández-Villamor, Adam Westerski, Mercedes Garijo
Universidad Politécnica de Madrid
ETSI Telecomunicación
Avenida Complutense 30, 28040 Madrid, Spain
{ bgarcia, jcduenas, jifv, westerski, mga}@dit.upm.es

Carlos A. Iglesias
División I+D+i
Germinus XXI (Grupo Gesfor)
Avenida Manoteras 32, 28050 Madrid, Spain
cif@germinus.com

***Abstract*— Web software development is one of the most active areas and fastest growing industries in software and services development in Europe. In particular, Java Enterprise Edition is the mainstream European technology option for one million European developers. Since web development is not still a mature area, the proliferation of frameworks and components has both increased the required skills of web engineers, and has considerably reduced their productivity. For that reason, the evolution of existing Java based web applications is a very hard and time-consuming task. ROMULUS project has researched Domain Driven Design (DDD) for web application development on Java by means of an open source metaframework. A web metaframework is an abstract layer that collects the main aspects of current web frameworks, such as persistence, security, web flow or authentication.**

Keywords: Web engineering, metaframework, domain driven design, mashup, reliability.

I. Introduction

ROMULUS project is aimed at providing an agile environment to build web applications by focusing on reusability and reliability through Domain Driven Design (DDD), mashup development, and automatic testing. Some deficiencies found in Java Enterprise Edition development inspired web frameworks such as the Spring Framework by Rod Johnson [1] or agile frameworks such as Ruby on Rails [2]. Although ROMULUS agrees with Ruby on Rails creators in the diagnosis of the situation that lead to low productivity with current mainstream web development practices in Java community, the proposal of ROMULUS is a different paradigm to develop web applications taking advantage of new trends in software engineering, such as DDD [3], Model Driven Architecture (MDA) [4], and mashup oriented development combined with agile development methodologies [5].

The main concept of ROMULUS is researching on novel methods for increasing productivity and reliability of web software development, particularly focused on Java web development. The basic idea behind this approach is the employment of a web metaframework. A web metaframework is an abstract layer which offers a common application programming interface to a set of pluggable existing Java frameworks such as Spring or JDO (Java Data Objecs) to transparently provide persistence, presentation, internationalization services, among others.

In order to integrate a new framework in the metaframework, these interfaces had to be implemented. ROMULUS follows a DDD strategy, where developers focus on domain model definition. This domain model had to be linked with the metaframework interfaces using Java annotations. An MDA process was in charge of generating the application code from the domain classes and the targeted framework. The main advantage of using a metaframework instead of using directly the frameworks is that this abstraction layer allows migration to other frameworks in a seamless way, and provides a unified interface for development tools. In addition, the use of this kind of software artifact provides a useful mechanism to evolve existing Java web applications reducing effort.

Furthermore, the metaframework provides unprecedented ability to mash up remote pieces of web applications. This is possible not only at user interface level (web 2.0 style mashup) but also at data level, thanks to the use of the W3C semantic web languages as backend data integration layer, and portal and enterprise level, packaging common functionality for portal components and enterprise systems.

Best practices and integration with popular integrated development environments are provided as well as an innovative layer to mash up the activities of otherwise separated projects. The project researched both in the development process and in the inclusion of soft goals of the system, in terms of scalability, security and reliability.

In order to have a serious impact, the project did not start from scratch, as it is based on two mature open source projects, Roma and Liferay, which were extended according to the proposal needs and following an open source project development methodology, in order to disseminate and exploit the results of the project. The idea of using these projects is to solve the formerly mentioned development productivity problems. Roma allowed making Java application development easy and integrate frameworks and tools using a metaframework, which reduced implementation time. Liferay allowed integrating a leading enterprise open source portal framework with relevant industrial impact.

II. OBJECTIVES

ROMULUS project has the goal of contributing to the promotion of a new open paradigm for the development of web systems using Java technology.

The first big goal of the ROMULUS project is increasing productivity of the web software development, focused on Java, by following **DDD** principles and implementing a **metaframework**. The proliferation of frameworks and tools, has both increased the required skills of web engineers (for choosing, mastering, integrating and maintaining), and has reduced considerably their productivity. A key objective of the project is the usage of DDD based on a metaframework that abstracts from the frameworks, together with an MDA approach which generates the application from the metaframework onto the de-facto standard and popular frameworks. Plug-ins for the most popular Integrated Development Environments (IDEs) to assist the development process will be provided.

The second objective is increasing software productivity by using **mashup** oriented development. This objective fosters the research on how web application development can be speeded up thanks to the reuse of existing services and components, as well as the defined methodologies. The project researches on several types of mashups. The first one is composed by *web services mashups* (such as Google Maps or Yahoo Pipes). The service descriptions can be successfully used in the application domain model description, favoring the web service dynamic discovery and composition. In second place, we have studied the *data-level mashups*. A set of methodologies, best practices and guidelines are being defined in order to help the metaframework user during complex data mashup processes. Also, a semantic approach is used to be able to publish internal data and import external data from the open web using RDF as interchange format, with no programming skills. In third place, this project researches on the definition of *portal mashup services* and their ability to reduce the development time, increasing its reliability. The metaframework provides portal services consumed by portal components and scripting languages. Finally, the project researches on how the definition of an *enterprise mashup service* can combine information from enterprise search engines, web services, messaging systems, and data integration solutions with information from external services.

The third goal is enhancing the quality of SW development by involving **soft goals** such as reliability, traceability, security and performance, as well as other quality attributes from the conception of the software. In software development, soft goals such as security or scalability are not usually taken into account from the very beginning in the development process. Soft goals are usually related to quality requirements and do not have a clear-cut criterion for their satisfaction, nor can they be allocated to a small subset of elements in the architecture. Romulus have performed research on security, traceability, reliability and performance soft goals, and their integration in the development process, based on the identification of soft-goals aspects and the usage of annotations against these aspects in the metaframework. Soft-goals annotations based on aspects, used both in the modeling and the implementation phases contribute to automate code and tests generation.

The fourth and final objective is the balance between client and server **scripting** technologies and languages. Current trends in web application development are changing the balance between client and server technology. A key research objective of this project is to research on the symbiotic combination of client and server technologies and how this can enable more usable application and faster integration or creation of pure client side technologies. Traditionally all interaction and state storage was managed by the server. The browser's function was to act as a very thin client. While this is desired for certain types of applications, it makes development much harder for complex applications with high interactivity and larger needs for keeping application state. With the appearance and generalization of richer client side technologies it is now possible to use the browser as a smarter client, potentially reducing development time while improving the desired characteristics of the developed application.

III. COMPOSITION

The ROMULUS project (FP7-ICT-2007-1) was organized in the context of the Information and Communication Technologies (ICT) research program, under the Seventh Framework Programme (FP7) for research and technological development. From Spain, Italy, Germany, Ireland and Romania, the following affiliations were involved in the project.

Informática **Gesfor** is a Spanish company with multinational presence. Its main core business is providing value to their customers through the concentration of consulting and services in the IT sector. Gesfor has led the Romulus project. **AssetData** S.R.L. is an Italian company which provides value added consulting services in the IT area. AssetData is active in the open source world contributing to several projects, first of all Roma Metaframework, which is one the basis of the ROMULUS project. **Liferay** Gmbh is a German company specialized in offering solutions based on the open source Liferay portal family of products in the European market. Universidad Politécnica de Madrid (**UPM**) is the first and largest of the Spanish Technical Universities. **Imola** Informatica S.R.L. (IMOLINFO) is an Italian company offering consulting, skill transfer and advanced software development. Digital Enterprise Research Institute (**DERI**) has been created thanks to a joint partnership between National University of Ireland (Galway), Hewlet Packard and Science Foundation Ireland. National Institute for Research and Development in Informatics (**ICI**) has been acting as a leading unit in Romanian IT R&D since its creation in 1970.

ROMULUS ran from January 2008 until December 2009, with a total funding of 3.1 million euro.

IV. ACHIEVEMENTS

The following table summarizes the wide variety of open source projects to satisfy Java web developers' needs under the ROMULUS project. These projects have been released under Apache 2.0 license.

TABLE I. OPEN SOURCE PROJECTS DEVELOPED IN ROMULUS

Aspect	Product	Web Page
Productivty	Roma	http://www.romaframework.org
	IDE4Romulus	http://www.ict-romulus.eu/web/ide4romulus
Enterprise	Liferay	http://www.liferay.com
	ESB4Romulus	http://www.ict-romulus.eu/web/esb4romulus
Mashups	DERI Pipes	http://pipes.deri.org
	MyCocktail	http://www.ict-romulus.eu/web/mycocktail
	SEAP	http://lab.gsi.dit.upm.es/apache/adam/seap
Semantics	LD2SD	http://ld2sd.deri.org/
	Microservices	http://lab.gsi.dit.upm.es/microservices
Security	Wapiti	http://www.ict-romulus.eu/web/wapiti
	OWASP	http://www.owasp.org
Reliability	ATP4Romulus	http://www.ict-romulus.eu/web/atp4romulus
	Segovia	http://www.ict-romulus.eu/web/segovia
Community	JSR	http://jsr.germinus.com
	MokaByte	http://www.mokabyte.it
	JavaTeam	http://www.ict-romulus.eu/web/javateam
End User	Scrooge	http://scrooge.germinus.com
	Cornelius	http://sourceforge.net/projects/cornelius

Roma Metaframework allows developers the creation of enterprise level Java applications with low effort following a real DDD approach. It's a new way to conceive the application: anything is a POJO (Plain Old Java Object), from the Graphic User Interface (GUI) forms to the persistent objects [6].

IDE4Romulus is a tool that provides IDE (Eclipse and NetBeans) integration facilities to Roma Framework.

Liferay portal is a tool that allows the easy creation of Enterprise Information Portal (EIS), being a website which integrates information, people and processes across organizational boundaries. The typical example of EIS is an enterprise intranet.

ESB4Romulus is a set of tools which allows the integration of Roma with an ESB (Enterprise Service Bus) layer. The Roma Enterprise module allows the exposition of web services defined in Roma over the ESB. The service is exposed by annotating the service implementation class, and the module realizes everything is needed to create the composite application to be deployed in the ESB.

DERI Pipes [7] is a visual environment for the construction of semantic web mashups. It delivers a wide range of operators to aggregate and transform data from various formats such as RDF, RDFa, Microformats or raw XML.

MyCocktail is a web application that provides a GUI for building mashups easily. Information obtained of REST (REpresentational State Transfer) services can be modified with operators and later presented with a wide variety of renderers. It has been integrated with DERI Pipes into a Visual Mashup environment [8].

Sindice Extended API Platform (SEAP) is a framework that enables creation of semantic web mashups. The architecture of mashups developed with SEAP is based on a semantic search engine located in the back-end, the mashup facilities exposed as web services in the middle and final products using the mashups as part of their SOA architecture. SEAP has been tested and used in practice to construct a number of practical solutions, including Sindice Widgets [9].

Linked Data to Software Development (LD2SD) [10] is an initiative that proposes serialization of various assets connected to software development into linked data. The goal is to maintain better control and awareness inside software projects by publishing project metadata with explicit indications about connections between different project resources [11][12].

Microservices provide a lightweight RESTful service description framework. They ease application description to quickly allow functionalities such as automatic discovery, automatic execution, automatic documentation, and automatic testing. Other approaches to service description try to describe every possible service. On the contrary, the idea behind microservices is to provide simple suboptimal descriptions for every kind of service [13].

Wapiti (Web application vulnerability scanner / security auditor) allows developers to audit the security of their web applications. It performs "black-box" scans, i.e. it does not study the source code of the application but scans the web pages of the deployed web application, looking for scripts and forms where it can inject data.

OWASP (Open Web Application Security Project) is a worldwide open community focused on improving the security of application software. Their mission is to make application security visible, so that people and organizations can make informed decisions about security risks.

ATP4Romulus (Automatic Testing Platform) is a tool which performs automatic test case generation, execution and reporting for Roma based web applications. It performs unit, integration and system level testing. The generated test cases check the functionality, performance, and structure of the web under test. It integrates existing testing frameworks such as JUnit, TestNG, DBUnit, HtmlUnit, JMeter, and JUnitPerf [14][15].

Segovia is an application that allows the user to control application vulnerabilities. It also shows reports about security tools.

The **JSR** project contributes to the promotion of a new paradigm for developing Java technology systems. It proposes the domain model as conductor in the design, for the development of Java web applications.

MokaByte is an Italian web journal for Java developers.

JavaTeam is a forge portal based on Liferay, a platform that allows users to make a collaborative development and also integrates Romulus tools for mashup development (MyCocktail) and security test (Wapiti) in the portal.

Scrooge is an end user application for personal and business economical management systems, and the study of new security tools necessary for the application of this new technological, social and economic model to online banking.

Cornelius is an innovative software application for project management entirely built using the Romulus Metaframework.

All in all, the **Romulus Metaframework** is an integrated pack of the presented tools in terms of productivity (Roma and IDE4Romulus), enterprise (Liferay and ESB4Romulus), mashups (DERI Pipes, MyCocktail, and SEAP), semantics (L2DSD, and Microservices), security (Wapiti and OWASP), and reliability (ATP4Romulus and Segovia).

V. Results & Lessons Learned

The ROMULUS project contributes to the open source web development community using Java technologies in the server side. As a result, we have created and improved 18 open source communities –see table 1– working in different aspects of web development, glued together by the Romulus Metaframework. The key idea behind the development process proposed by the ROMULUS project is the concept of metaframework. Using it, developers can take the advantage of existing frameworks, such as Spring or JDO, simply changing the application configuration. That way the evolution process is greatly simplified.

ROMULUS project has also researched in new trends on software engineering such as DDD or mashup development. The domain is an in-memory model representation for the data, logic, and requirements employed by a software application. In one hand, the domain is a specific piece for each single application. On the other hand, soft goals such as reliability and security could be reused to a specific domain. These concepts are both involved in the ROMULUS proposal: developers should focus on the creation of the domain and the soft goals will be automatically covered. Therefore the maintainability and reusability of Romulus-based web applications will improve, because the main piece a developer should take care is the core of each application: the domain. Mashup development is an increasingly growing technique in order to evolve web applications towards systems composed by means of available components and services. ROMULUS project has covered this by means of open tools and methods. The Romulus Metaframework reduces the evolutional maintenance due to frameworks evolution, since once the metaframework MDA has been evolved, Romulus applications can be updated to new versions of a metaframework or migrated to a new framework with a 90% of automation.

The proposed technologies have been employed in several case studies, e.g. Scrooge and Cornelius. These demonstrators show how Romulus provides a bunch of integrated technologies which simplifies significantly the Java web development. The demonstrators have also shown some of the main advantages of using a metaframework, such as the ability of changing the view layer without extra programming, the introduction of soft-goals during all the process, or how a client-driven versus server-driven interface can be built. In addition, some mashup facilities have been integrated in the demonstrators in order to show its potential to integrate enterprise services, web services, semantic search engines, or portal components.

References

[1] Johnson, R., “Expert One-on-One Java EE Design and Development” Wrox Press, October 2002.

[2] eWeek, “Ruby on Rails: Making Programmers Happy”, Available at http://www.eweek.com/article2/0,1895,1880280,00.asp. November 2005.

[3] Evans, E., “Domain-Driven Design: Tackling Complexity in the Heart of Software”, Addison Wesley, 2003.

[4] Beydeda, S., Book, M., Gruhn, V., “Model-Driven Software Development”, Springer, 2005.

[5] J. I. Fernández-Villamor, L. Díaz-Casillas, and C. A. Iglesias, “A comparison model for Agile Web Frameworks”. Proc. of the 3rd Euro American Conference on Telematics and Information Systems (EATIS08), 2008. Aracaju, Brazil. September 2008.

[6] P. Martín, G. Hernández, C. A. Iglesias, L. Garulli, and G. Maestro, “Janus, a new View Generator of Roma Framework based on templates”. Proc. of the Symposium on Telematics Engineering, 8th edition, Universidad Politécnica de Cartagena, Spain. September 2009.

[7] D. Phuoc, A. Polleres, C. Morbidoni, M. Hauswirth, and G. Tummarello, “Rapid Prototyping of Semantic Mash-Ups through Semantic Web Pipes". Proc. of the 18th International World Wide Web Conference (WWW09), ACM. Madrid, Spain. April 2009.

[8] A. Westerski, “Integrated Environment for Visual Data-level Mashup Development”. Proc. of the 10th International Conference on Web Information Systems Engineering (WISE09). Poznan, Poland. 2009.

[9] A. Westerski, A. Iqbal, G. Tummarello, and S. Decker, “Sindice Widgets: Lightweight embedding of Semantic Web capabilities into existing user applications”. Proc. of the 4th International Workshop on Semantic Web Enabled Software Engineering (SWESE08). Karlsruhe, Germany. 2008.

[10] A. Iqbal, O. Ureche, M. Hausenblas, and G. Tummarello, “LD2SD: Linked Data Driven Software Development”. Proc. of the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE09). Boston, USA. July 2009.

[11] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao, “Describing Linked Datasets - On the Design and Usage of voiD, the ‘Vocabulary of Interlinked Datasets’”. Proc. of Linked Data on the Web Workshop (LDOW09) at WWW09. Madrid, Spain. April 2009.

[12] M. Hausenblas, W. Halb, Y. Raimond, L. Feigenbaum, and D. Ayers, “SCOVO: Using Statistics on the Web of Data”, Proc. of the Semantic Web In-Use Track (ESWC09). Greece. February 2009.

[13] J. I. Fernández-Villamor, C. A. Iglesias, and M. Garijo, “Microservices: lightweight service descriptions for REST architectural style”. Proc. of Second International Conference on Agents and Artificial Intelligence. Valencia, Spain. January 2010.

[14] B. García, J. C. Dueñas, and H. A. Parada, “Automatic Functional and Structural Test Case Generation for Web Applications based on Agile Frameworks”. Proc. of the 5th International Workshop on Automated Specification and Verification of Web Systems (WWV09). Hagenberg, Austria. July 2009.

[15] B. García, J. C. Dueñas, and H. A. Parada, “A Template-Based Method for Automatic Unit Test Case Generation of Web Applications”. Proc. of the 6th Workshop on System Testing and Validation (STV08). Madrid, Spain. December 2008.