

Romulus

Building Java Web Applications in a more productive, reliable and secure way.



Romulus is an industrial Open Source Project for improving significantly Java Web Development, in terms of productivity, security and reliability. Romulus approach is based on a metaframework integrated with enterprise systems and IDEs.

At a Glance

Project:

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

Project coordinator

Informática Gesfor (Spain).

Partners from:

Asset Data (Italy), Liferay Gmbh (Germany), Universidad Politécnica de Madrid (Spain), Imola Informatica (Italy), National University of Ireland – DERI (Ireland) and Antional Institute for Research and Development in Informatics – ICI (Romania).

Duration:

01/01/08-31/12/09

Total cost:

2.192.005€

Programme:

Seventh Framework Programme. Theme ICT-2007 1.2. Service and Software Architectures, Infrastructures and Engineering.

Further information:

More information of the project can be found at <http://www.ict-romulus.eu>



ROMULUS

Why Romulus?

Web Software Development is one of the most active areas in software development in Europe, and Java Enterprise Edition is the preferred option for one million Europeans, with more than 38.6% of European developers according to Evans Data. Web development is still not a mature area, which has brought many new technologies and frameworks. Engineers are not able to master all these technologies, which affects seriously to their productivity.

The appearance of a new software development paradigm for web applications not based on Java such as Ruby on Rails and modern web application mashup techniques has questioned the traditional approach of web software development, and has shown relevant increments in productivity and reliability. Nevertheless, the wide adoption of Java in the European companies is claiming a Java suitable answer for web development.

Romulus project has the goal of contributing to the promotion of a **new open paradigm for development web systems with Java technology**. The project researches on domain driven design for Web Application Development on Java and provides an open source **Java metaframework** for developing web applications.

Romulus impact and Europeanness

The impact of Romulus in European software and services development is based on the fact that most software and services are developed in Java Enterprise Edition in Europe, being Java the most popular language and skills on Java Enterprise Edition are the most demanded in Europe. Europe has 1 million Java developers out of 4.5 million in the world.

Romulus has the goal of improve productivity in Java Web Development. Due to the impact of software on

economy, reducing the cost of software development and improving software quality are important economic objectives in Europe. In addition, Romulus is delivered as an Open Source project, which promotes cost reduction and improvement of innovation.

Romulus is carrying out an active dissemination in open source communities (Roma Metaframework, Liferay and OWASP), as well as standardisation in W3C within the groups *Semantic Web Deployment Working Group* and *RDB2RDF Incubator Group*.

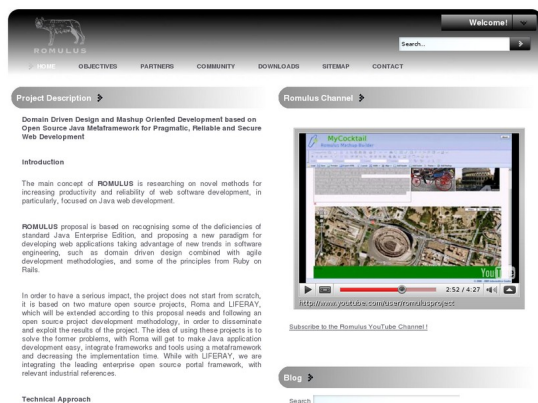


Figure 1: Romulus site <http://www.ict-romulus.eu>

Romulus consortium

Romulus brings together complementary expertise from industry and academia. Gesfor contributes with its expertise in security contributing to OWASP (Open Web Application Security Project) and web development. ASSETDATA brings its expertise agile development and Java standardisation. LIFERAY brings its product expertise in portal technology with their leading Liferay Portal product. IMOLA contributes with its expertise in integration of enterprise systems and its standardisation in the Java community. DERI provides expertise in semantic web technologies applied to software engineering. UPM and ICI contribute with their expertise in researching on emerging software development methods.

Scientific and Technical Goals

Romulus proposes to advance in Java Web Development with the four scientific goals shown in Figure 2.

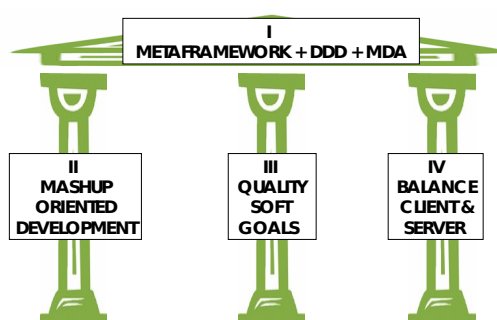


Figure 2. Romulus Scientific and Technical Goals

Metaframework and Domain Driven Design

While many times, developers focused on the details of the technologies they use, Romulus proposes to

focus on domain engineering, delegating to a metaframework and making usage of code generation techniques for generating most of the web application from this domain model.

Java has many available robust frameworks which provide packaged functionality, such as persistence, security, web flow or authentication, but mastering all these frameworks requires too much specialised staff. Romulus proposes the usage of a metaframework, extending Asset's Roma Metaframework. A web metaframework is an abstract layer that collects the main aspects of current web frameworks and defines generic interfaces for these functionalities. The main benefits of a metaframework are that developers access the metaframework, and their applications can select the targeted framework, adapting the web application to their needs.

Romulus provides a suitable agile **methodology** to develop applications with the Meta Framework concepts and improve the **Roma Meta Framework** itself. This methodology is based on Iterative development, Domain Driven Design, Test Driven Development, Soft goals, division by Aspect and Best Practices. In the first year, the main improvements to the Roma Meta Framework have been a new **plug-in** architecture, a new console **wizard** to check and install upgrade in transparent way, the creation of IDE4Romulus (the new **Eclipse IDE plug-in**) and the integration of **Logging Aspect**.

The most important module developed within Romulus is **Janiculum**, an alternative implementation to the Echo2 View module. Janiculum provides facilities for automatic generation and customisation of the View Aspect based on standard web technologies, such as HTML, CSS and JavaScript. Previous applications are 100% portable to the new module in the spirit of the Meta Framework.

Furthermore, an integration with the **Tevere Flow** Open Source product is provided. Tevere Flow is a full transactional workflow engine with the Web Editor integrated. It supports most used workflow patterns and now it is fully integrated in Roma through the new **Workflow Aspect** and internally through the **Service Aspect**. The need to have a Semantic Aspect has emerged in order to support **Data and Services Mashups for Enterprise Integration** and integrate Semantic Powered Development Tools. A basic implementation has been developed to export the domain as RDF just by annotating the interested classes.

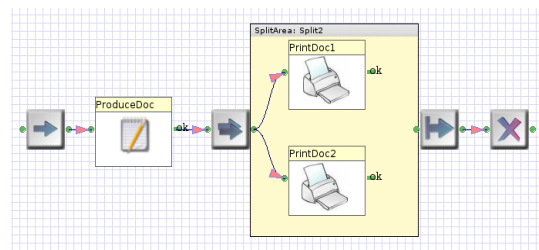


Figure 3: Workflow aspect integrated in Roma

Mashup oriented development

Romulus researches 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 (mashup web services, data level mashups, enterprise

mashups and portal mashups) in order to integrate external services, enterprise systems, data sources or portal components.

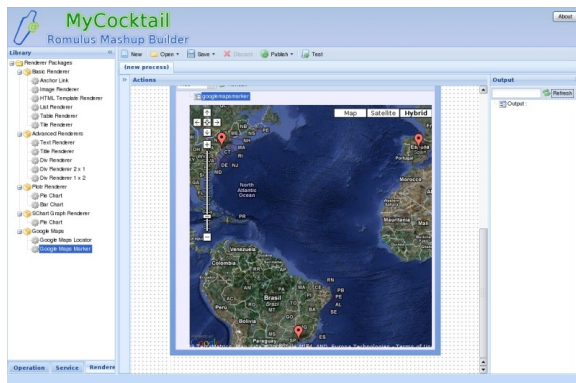


Figure 4. MyCocktail Mashup Builder
[<http://www.ict-romulus.eu/web/mycocktail>]

Regarding **data-level mashups**, methods for exporting and importing structured data from bug tracking software via linked data have been applied and various widgets to consume this data have been developed.

Further, a research into aggregation using the award-winning semantic DERI pipes has been performed. For **portal mashups**, two approaches have been followed to achieve the goal of mashing up the functionality that a portal delivers. The first approach makes portal functionality accessible from other platforms, such as Facebook or Netvibes. All the developments are integrated in Liferay and are available out of the box for any user.

The second approach provides the ability to build new applications combining specific parts of already existent applications. Different alternatives for fragment composition have been analysed and additionally, a lightweight JavaScript library has been developed allowing exploiting this in a straightforward manner.

Regarding the integration of **Web services mashups**, two approaches have been considered: using annotations inside the code to access external services on the one hand, and the creation of these mashups with a graphical tool on the other hand, MyCocktail, the Romulus Mashup Builder.

Finally, for the **integration of enterprise mashups**, the evolution from Web to enterprise mashups has been described and a reference-architecture for Romulus has been introduced. This architecture is based on the ESB component, while the integration mashups are based on BPEL standard and on an extension of the BPEL editor provided in NetBeans IDE.

Soft Goals for higher quality

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 (non-functional) requirements and do not have a clear-cut criterion for their satisfaction, nor can be allocated to a small subset of elements in the architecture. Romulus researches 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.

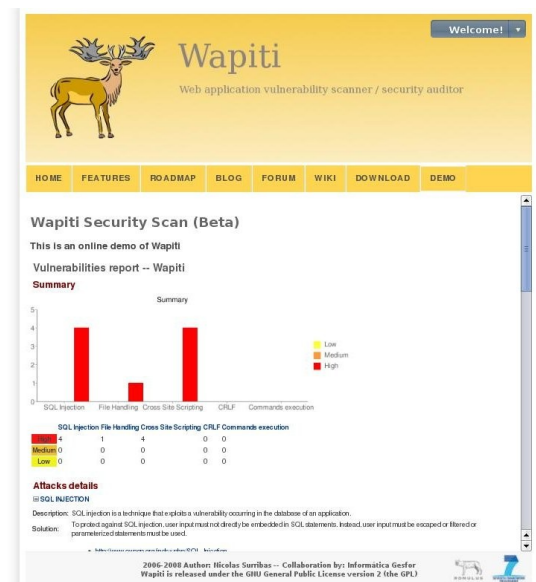


Figure 5. Security report generated by Wapiti
[<http://www.ict-romulus.eu/web/wapiti>]

In the Security field, work has been carried out to assess how Roma MetaFramework can improve its security. A set of security annotations have been proposed as the result on the research on several RAD frameworks. Also, several contributions have been made to the security tool Wapiti, a black box security testing tool, including performance enhancements, code refactorings for easier extensibility, a report generator and improvements to some attack detectors.

In order to ensure the quality of the software has been developed ATP4Romulus, a testing tool for Romulus applications. This tool allows to perform automatic tests of the applications generated using Roma Framework. It carries out several tests about different aspects as CRUD, View and I18N.

Balance between client and server

Nowadays we are moving from a server-side centric technologies to client-side centric technologies, and it is needed to research on the right balance between client and server technologies depending on the requirements. In addition, web2.0 is promoting an active role of end-users that should be included when designing configurable applications. Romulus researches in this direction in order to meet different kinds of interface requirements.

Several options have been provided to developers and end users (with different skills) to simplify the process of applications development.

Portlets development using scripting languages reduces entry barriers to different developers, while increasing productivity and maintaining full access to existent API. All that is needed is to create a portlet descriptor and write all the logic in the selected scripting language, providing mechanisms for application integration.

Beyond that, scripting technologies deliver the possibility to create applications from the client side, writing the code that will generate the view of the application. As there is no need for deployment, developers productivity is increased.

Applications can be customized, writing the code that will implement certain function, to meet customers needs. As an example, a form where user can define their own validation functions as complex as desired (even validating a field against others) has been developed. Skilled users (with proper permissions) can update the JavaScript code and it gets executed both in the client and in the server, with only one definition.

Portlets can be arranged together and they behave jointly, exchanging information among them, providing, for example, easier wiki navigation or tagged content visualization.

Provided documentation (available in project's homepage) contains additional information about client side/server side balance and pure client side components, allowing advanced visualizations of data based on client resources.

Vertical Demonstrators

In order to check the applicability of Romulus' technologies in real contexts and demonstrate how they can improve considerably productivity in web development, some demonstrators are being created: Musiteca, Cornelius and Scrooge.

Musiteca is a musical content management application, where the user can watch or listen music pieces. Cornelius is an application that allows to manage projects and company's employees. And Scrooge is an application for personal and enterprise financial management.

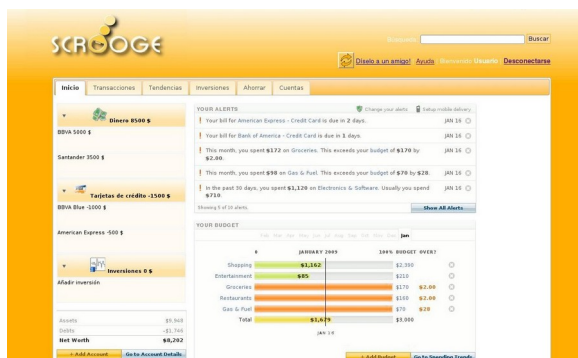


Figure 6: Scrooge application

For further information:

Informática Gesfor

Avda. de Manoteras, 32 Edificio Gesfor
28050 – Madrid (Spain)
Tel (+34) 91 304 80 94
Fax (+34) 91 754 50 52
Web Page: <http://www.ict-romulus.eu>

Software & Service Architectures and Infrastructures

European Commission - Information Society and Media DG
Office: BU25 3/134 B-1049 Brussels
Email: info-st@ec.europa.eu
Tel: (+32) 2 298 93 02
Fax: (+32) 2 296 70 18
Web Page: <http://cordis.europa.eu/fp7/ict/ssai/>