

Towards a Model-driven Development of Web Applications

José Luis Herrero, Pablo Carmona and Fabiola Lucio

*Department of Computer and Telematics Systems Engineering, University of Extremadura,
Avda de Elvas s/n, Badajoz, Spain*

Keywords: Model-driven Architecture, Web Services, Web Applications, Rich Internet Applications.

Abstract: In the last years, traditional web applications have evolved from static web pages to dynamic applications, and this is the reason why the complexity of this type of applications has been increased. With the appearance of AJAX and Web 2.0 technology, a new breed of applications for the Internet has emerged. However, as web applications become more and more complex, the quality degree is negative affected, since initial stages of software life cycle are not considered in the development process of this type of applications. In order to solve this problem, this paper proposes a model-driven architecture to support web application development from the design to the implementation model. With this aim, the following tasks have been performed: a new profile extends UML metamodel with new concepts from the web domain, a new framework supports the developing of web applications by composing web elements, and also provides different types of communication patterns, and finally, a transformation model generates web applications from the UML extension proposed. The main contribution of this work is an increase in the quality degree of web applications provided by the model-driven architecture proposed, but other advantages that can be mentioned are a high performance degree achieved by a prefetching cache mechanism, and a high reusability, since web elements can be reused in different web applications.

1 INTRODUCTION

The rapid evolution of Internet has transformed the development process of web applications and different types of applications for the Web have arisen. Traditional web pages were based on HTML tags and script code, but however the type of applications that could be developed was limited and a low degree of interactivity was achieved. With the appearance of Web 2.0, and the introduction of Asynchronous JavaScript and XML (Garret 1995), which supports the modification of individual web elements without refreshing the entire web page, the performance degree of this type of applications has been increased, achieving a new breed of web applications with a high degree of interactivity has emerged.

Rich Internet Applications, RIA (Paulson, 2005) have emerged under the umbrella of these new technologies, and have gained much attention and acceptance, because a web browser is the only execution environment required avoiding users the *installation of new software or plugin. For this reason, software vendors are actually adapting their applications to this new trend: Adobe has presented

Adobe Integrated Runtime (AIR) that extends web development to the desktop, Google has introduced Google Docs as is a new technology to work with online documents, and Microsoft has developed Windows Presentation Foundation to provide support for web browser applications (WBAs). The interest exposed by all these software vendors demonstrates the attention this technology is gaining. However, as web applications become more and more complex, the quality degree is negative affected because initial stages of software life cycle are not considered in the development process of this type of applications. Another important implication of this evolution is that the size of web applications is increasing in exponential rate, which provokes a severe increment in latency, since web applications are required to be downloaded before they can be used.

In order to solve these problems, the Model Driven Architecture (MDA), has been adopted to increase the quality degree of web applications. This architecture unifies and simplifies modelling, design, implementation, and integration of applications by defining software fundamentally at the model level, which increases the quality of complex software

systems based on creating high level system models and automatically generating system architectures from the models. Also, a new framework is proposed to minimize latency by incorporating prefetching cache mechanisms.

The objective of this paper is to propose a model-driven architecture to develop web applications with a high quality degree. This task has been achieved at three levels: first a new profile incorporates web application concepts and extends UML metamodel, then a new framework supports downloading web components dynamically only when they are required, and incorporates several communication patterns between web applications and remote resources, and finally a transformation model generates web applications from the UML extension proposed.

The paper is organized as follows: section 2 presents the motivation with an example of a scenario and related works are summarized, a UML profile is proposed in section 3. Then a new framework to develop web applications is explained in section 4, while web components are defined in section 5. Next, the transformation model from a is proposed in section 6, and finally the paper concludes in section 7.

2 MOTIVATION AND RELATED WORKS

The rapid evolution of web technologies has contributed to the appearance of new type of web entities to support client's requirements. Currently, web applications are complex software composed by different types of components, and this is the reason why the development process of this type of software is so difficult. Different AJAX frameworks have been proposed to provide additional benefits for web applications; as such the push-based concepts have been applied to develop web applications that require real time event notifications. According to this, Echo3 is a platform for building web applications that approaches the capabilities of rich clients, and presents a JavaScript client-side, which is executed directly by the client's web browser, while a Java server-side is stored in a Servlet Container. GWT is a Google proposal for building and optimizing complex web applications, AJAX applications are written in Java and compiled into a highly optimized JavaScript code that runs across all browsers. Dojo Offline is a cross-framework that enables web applications to work offline, where the information generated by the user

is stored inside the web browser. In (Mesbah, 2008) presents a comparative of these approaches, and proposes a new architectural style (SPIAR) which captures the essence of AJAX applications.

Trends in web application development are not limited to the specification of new frameworks, the design level is also considered. A review of the principles of structured programming and the preferred characteristics of web applications is presented by (Marquis 2002). This work presents a mapping of how the traditional guidelines may be applied to web applications development. In the field of MDA, there is a consensus about the benefits that this technology offers to web application development: a reduction of sensitivity to the inevitable changes that affect a software system (Atkinson and Kuhne, 2002), a reduction of cost and complexity (Miller and Mukerji, 2003), and provides an increase of abstraction (Booch et al., 2004). An interesting analysis about the existing problems in the field of web engineering and how they can be solved by model-driven development approaches is presented in (Gitzel et al., 2007), identifying the problems encountered in the development process of web applications such as their dependence on the HTTP protocol, compatibility issues due to the heterogeneity of web browsers, and the lack of performance because of the increase in the latency degree.

A variety of methodologies to develop web applications have been proposed. First, we can mention OOHDMM (Lange, 1995) which defines abstraction and composition mechanisms in an object-oriented framework to, on one hand, allow a concise description of complex information items, and on the other hand, allow the specification of complex navigation patterns and interface transformations. Other interesting researches are a proposal for modeling web applications extending UML (Conallen, 1999), and UWE-R (Koch et al., 2002) that defines a UML extension with new stereotypes, which is focus on navigation, presentation and server interaction. Finally, WebML (Ceri et al., 2000) is a visual language for specifying the content structure of web applications and the organization and presentation of contents in one or more hypertexts. A more extensive work on the evolution of Web engineering methodologies can be found in (Fraternali 2010).

3 WEB APPLICATION MODELING

This section proposes an extension of UML that incorporates specific concepts of the web application domain, and with this aim a new profile is presented (Figure 1).

The main element of this profile is the <<Web Resource>> stereotype that represents a general element in the web. A software entity in the web is represented by the <<URL resource>> stereotype which incorporates its description and a set of keywords to facilitate searching tasks. A general application for the web is represented by the <<Web application>> stereotype which can include several web elements represented by the <<Web Element>> stereotype. Web components are defined using the <<Web component>> stereotype which can be active or paused according to the requirements of the application.

External entities are represented by the <<Web service>> stereotype, and can be bound to a web application or to a web component. A web service must specify the communication port (<<Port>> stereotype), the operations that provides (<<Operation>> stereotype), the type of in/out messages and out messages of all its operations (<<inMessage>> and <<outMessage>> stereotypes

respectively). Finally, web services can be invoked using different communication protocols (<<Synchronous>>, <<Callback>> and <<Poll>> stereotypes). These protocols are explained in a later section.

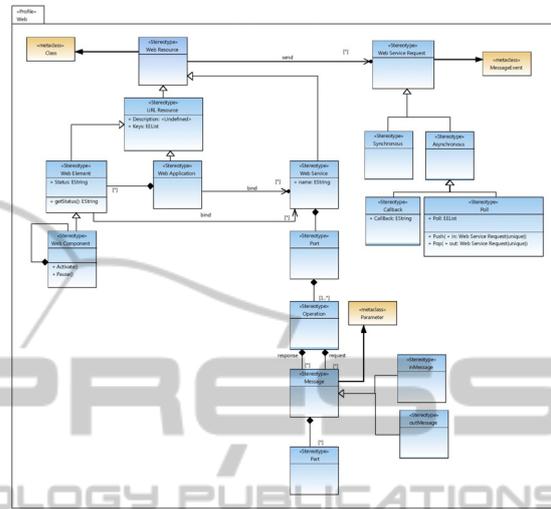


Figure 1: UML extension.

An example of a web application is shown in Figure 2. The example presents a web editor which provides different types of controls to define web content.

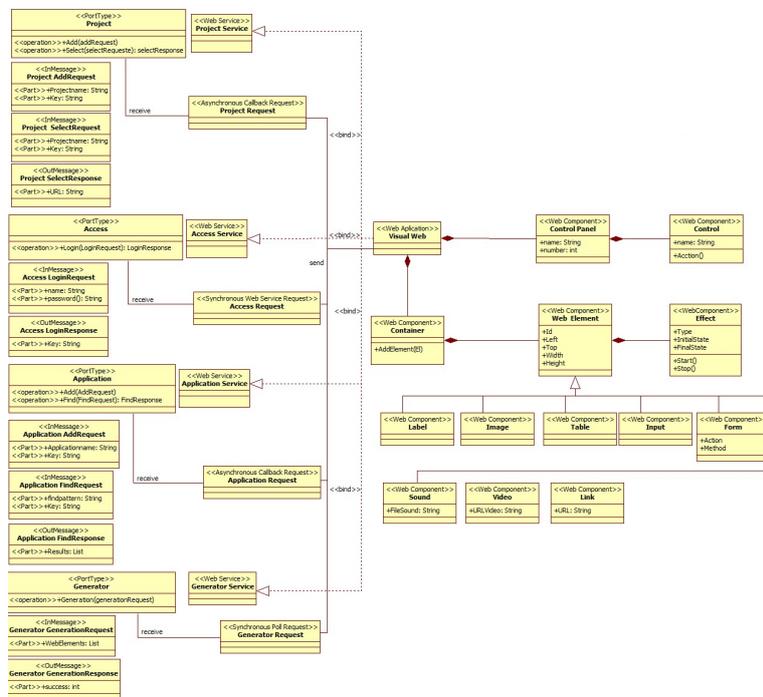


Figure 2: Web editor design.

This application is downloaded and activated in the user's web browser, and it is composed by different elements that support visual modelling, project management, code generation and access control. In this scenario, a web application is built by composing different software entities that can be activated in the client's side (web components), and remote operations, which are invoked using services technology.

4 THE FRAMEWORK

In this paper we considered that web applications are characterized by the following properties:

- Downloaded: they are downloaded from the web and activated in a web browser environment.
- Heterogeneous: web applications are composed by different types of web entities. Local entities are downloaded while external entities can be invoked using web services technology.
- Distributed: The elements that conforms a web application can be distributed in the web and can be executed locally or remotely.

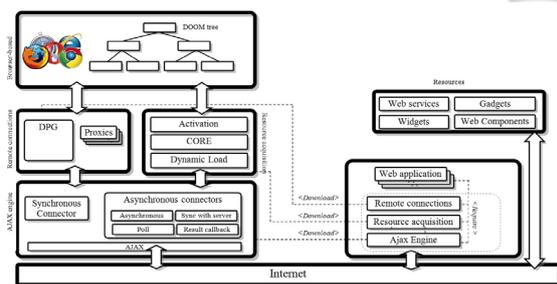


Figure 3: The framework.

This type of applications requires new mechanisms to provide the downloading and activation of web components at the client side, and the invocation of remote services. The framework (Figure 3) is deeply explained in our previous work (Herrero et al., 2011).

4.1 Web Services

Complex tasks may require special computers to be performed, or may demand too much time to be downloaded (latency). For these reasons web elements are not the best approach in these situations. The solution comes from computer networking, where resources are distributed across multiple computers, providing a highly load-balanced application. According to this trend,

complex tasks are hosted in remote computers and can be requested through a communication mechanism.

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL), and other systems interact in a manner prescribed by its description using SOAP messages. Web services is the model selected to connect web applications for two reasons: i) offers cross-platform, cross-programming and internet-firewall-friendly, and ii) can be invoked from web applications using the HTTP protocol and SOAP messages, which simplifies the connection mechanism.

The binding process proposed in this paper (Figure 4), asks a service registry for a specific service, and generates a proxy object from the WSDL specification returned. This proxy is responsible for marshaling the invocations and sends them to the correct service provider.

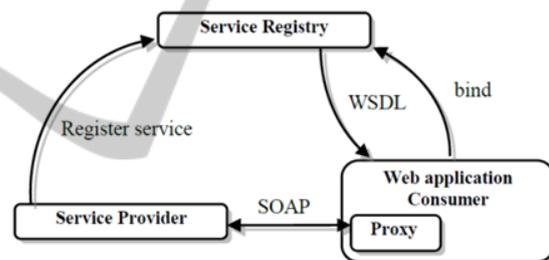


Figure 4: Binding process.

In this proposal web applications are composed by heterogeneous web components. A web component is a web resource that can be downloaded and activated in a web application. The specification of a web component is divided into the following elements:

- Definition: the name of the web component, according to a namespace, and a description are included in this element. The set of requirements required.
- Interface: this element provides a successful interaction mechanism with the web application and specifies the set of operations provided by the web component using the IDL language.
- Functionality: the code of all the operations is stored in this element.

4.2 Web Component Activation

The activation mechanism of web components is supported by the resource acquisition component of the frame proposed in a previous section. The protocol to activate a Web component is as follows:

1. When a web application requires an external URL resource, a message is sent to the resource acquisition component which builds a request and passes it to the Ajax engine module.
2. The Ajax engine module transforms the request into several asynchronous calls according to the callback pattern.
3. When the responses have been received, the dynamic load module parses the answers and builds a local object.
4. Finally, the core module activates the local object into the web application by adding new modules in the DOM tree. At this point, all the operations provided by the web component can be invoked by the web application.

4.3 Performance Quality Factor

The main problem of this proposal is that performance degree is severely affected by latency, because web components are required to be downloaded before they can be used, and this is the reason why performance is negatively affected. In order to reduce this problem and increase the performance quality degree of web applications, we propose a proxy cache mechanism that intercepts requests from the web application and performs the following tasks:

- Stores a list of downloaded web elements.
- Checks if a requested web element has been downloaded previously, avoiding duplicated requests.
- Applies a prefetching algorithm that identifies the web elements to be downloaded before they are requested.

Data prefetching has been widely used in memory cache strategies, where it has been used to anticipate cache misses and fetch data from the memory system before the processor needs the data (Tien-Fu, 1997) and (Kai-feng et al., 2005). In this section a software prefetching technique is adopted to minimize latency in web applications. However, the main problem of this technique is to predict the web elements that will be requested, and to download them without interrupting the normal workflow of the web application.

In order to identify the web elements that will be downloaded in advance, the cache proxy intercepts

the source code when a web application is downloaded, and performs a searching process to locate the web elements that will be required. This process identifies specific patterns in the source code of the web application that correspond with the requesting or manipulation of web elements. The result of this process is a list of predicted web elements that will be downloaded in a background process.

5 MODEL GENERATION

One of the keys of MDA is the capacity of defining transformations from higher-level models to platform specific models, guided by a set of transformation rules. This section describes the process to transform an UML design, according to the extension proposed, into a web application.

- The web application stereotype is transformed into the specific components of the framework (Remote Connections, Object Acquisition and Ajax Engine).
- The web service stereotype activates a binding process that compares the WDSL specification of the web service with the UML design. The binding relationship must provide new information in order to achieve this task (service location, authentication).

Figure 5a, 5b, and 5c show the code generated automatically by a generation tool.

```
<WEBCOMPONENT>
  <NAME>Container</NAME>
  <TYPE> LOCAL</TYPE>
  <DESCRIPTION....</DESCRIPTION>
  <VERSION>1.0</VERSION>
  <AUTHOR>...</AUTHOR>
  <REQUIREMENTS>
    Web Application
  </REQUIREMENTS>
</WEBCOMPONENT>
```

Figure 5a: Container web component definition.

```
Interface Container{
  void add (Element el)
}
```

Figure 5b: Container web component interface.

```
function Container{
  this.WebElement = new Array() ;
}
Container.addElement=function(Element el){
}
```

Figure 5c: Container web component functionality.

The generation tool has been developed using Eclipse with the following plugins:

- Papyrus Project: this plug in is used to define the new profile and model the web application example.
- Acceleo project: a model can be covered according to Acceleo primitives.

6 CONCLUSIONS

In this paper a model-driven architecture to develop web applications is proposed and, with this aim, different stages of software development process are considered. At design level, a new profile extends UML metamodel with new concepts from the web application domain. It is also proposed a new framework to support the dynamic download of web elements, and provides different communication patterns (synchronous, asynchronous: one way message, pool and callback) with remote web services. Finally, a transformation model generates web applications from the UML extension. The main benefits provided by this proposal are:

- The quality degree of web applications is increased, because early stages of software development process are considered.
- The delegation of tasks to other computers through web service technology provides a highly load-balanced application.
- As web elements are deployed independently, they can be reused in different web applications, and therefore, the reusability degree is increased.

A prototype of a web editor application has been presented along this paper, the scenario of the example has been introduced, the design, according to the UML extension, has been presented, and finally, an example of a generated web component (container) have been defined.

Future works will try to incorporate new types of web entities such as applets or dynamic web contents generated by server languages into the development process of web applications. Moreover, historical information will be included in the prefetching algorithm in order to predict more

efficiently the web elements required to be downloaded in advance.

REFERENCES

- Atkinson C., Kuhne T., 2002. The role of metamodeling in MDA, in: International Workshop in Software Model Engineering (in conjunction with UML '02).
- Booch G., Brown A., Iyengar S., Selic B., 2004. An MDA manifesto, MDA Journal.
- Ceri, S., Fraternali, P., Bongio, A., 2000. Web modeling language (WebML): a modeling language for designing Web sites. In: Proceedings of the 9th World Wide Web Conference (WWW'09), pp. 137–157.
- Conallen J., 1999. Building Web Applications with UML. Addison-Wesley Professional, pp. 320
- Garrett J., 2005. Ajax: A new approach to web applications. Adaptive path. <http://www.adaptivepath.com/publications/essays/archives/000385.php>, viewed on 10/03/2012.
- Gitzel R., et al, 2007. Using established Web Engineering knowledge in model-driven approaches Science of Computer Programming, Volume 66, pp 105-124.
- Herrero, J. L., Carmona P., Lucio F. 2011. Web services and web components. Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on , vol., no., pp.164-169, 19-21
- Kai-feng Wang, et al., 2005. Path-based next N trace prefetch in trace processors. Microprocessors and Microsystems, Volume 29, Issue 6, pp 273-288.
- Koch, N., Kraus, A., and Hennicker, R. 2001. The Authoring Process of the UML-based Web Engineering Approach. In First International Workshop on Web-oriented Software Technology (IWWOST01).
- Lange D., 1995. An object-oriented design approach for developing hypermedia information systems. In: Sprague R, editor. 31st Annual conference on systems science.
- Marquis G. P., 2002. Application of traditional system design techniques to web site design. Information & Software Technology, 44(9):507-512.
- Mesbah, A. van Deursen, 2008. A component-and push-based architectural style for AJAX applications, Journal of Systems and Software, Volume 81, Issue 12, pp. 2194-2209.
- Miller J., Mukerji J., 2003. MDA. Guide version 1.0.1, OMG document number: omg/2003-06-01, OMG, 12.6.2003. <http://www.omg.org/cgi-in/doc?omg/2003-06-01>.
- Paulson L. D., 2005. Building rich web applications with Ajax. Computer, IEEE, vol.38, no.10, pp.14-17.
- Tien-Fu Chen., 1997. Reducing memory penalty by a programmable prefetch engine for on-chip caches. Microprocessors and Microsystems, Volume 21, pp 121-130.