Web services and web components

J. L. Herrero, F. Lucio, P. Carmona
Department of Computer and Telematics Systems Engineering
University of Extremadura
Badajoz, Spain
jherrero@unex.es, flucio@unex.es, pablo@unex.es

***Abstract*—New trends in software development have proposed different interaction mechanisms for software applications. For instance, the Software as a Service (SaaS) model has defined a new delivery and access method, and installation and update processes in the user's computer are not required, software is hosted in a server and can be used over Internet instead. Middleware platforms are required in SaaS frameworks in order to support software services. Currently, web applications are taking full advantages of SaaS benefits, because they can be delivered over Internet and developed according to a service model. However, web applications are still under a maturity process and a low degree of efficiency and reusability are achieved. This paper proposes a web service-based framework according to component software engineering (CBSE), with the aim of developing efficient and reusable web applications. Moreover, web services are the middleware technology that provides full accessibility for this type of components.**

Keywords: Software as a service, Web services, web application, component-based development, RIA

I. INTRODUCTION

Software as a Service (SaaS) is a new trend in software community that supports software on-demand. In this model, software is not required to be installed or updated in the user's computer, it is hosted in a server instead [1]. The benefits provided by SaaS applications are:

- **Delivery:** reduce the costs of code delivery to the customer, because software is delivered over Internet.
- **License:** users do not need to purchase a software perpetual license.
- **Maintenance**: software maintenance is supported by the Saas vendor.
- **Support:** the infrastructure that provides access to software services is also supported by the Saas vendor.

Software service operators have the responsibility of hosting, updating and maintaining the software applications, while user's can use them anytime, anywhere. In order to support this type of software, a middleware platform is required; web services and SOA are the main adopted technologies to develop Saas platforms. The services provided by SaaS platforms can be used directly by users, but also, these services can be accessed by applications in order to increase their functionality.

Web applications are emerging as a new trend in software community and are taking full advantages of SaaS benefits. Among all possible web application definitions, we have chosen in this paper the following: "a web application is an online-only application that runs in a web browser". According to this definition, web browsers are assuming a new role, and they can be viewed as a container in which web applications can be loaded and activated. Different technologies have been introduced in the development process of web applications in order to increase the efficiency degree, as such Rich Internet Application (RIA) enhances the interactivity of desktop applications, and provides mechanisms to run instructions inside the web browser, thereby circumventing the slow and synchronous communication loop between the server and the client. Now, RIA technology is becoming the mainstream as the user interface not only in personal computers but also in embedded devices.

However, as web applications becomes more complex, new problems arise: a) RIA technology requires web applications to be completely downloaded before they are activated, therefore this have a highly negative impact on the efficiency degree, b) a lack of reusability degree is achieved by this type of applications.

Different technologies provide an increase in the reusability degree of software applications. On one hand, Component-based Software Engineering (CBSE)[2] has emerged as a technology for the rapid development of software applications, in this model software is built assembling components and therefore, a high degree of reusability is achieved. On the other hand, web services make applications cross-platform and increase the reusability degree of distributed software, this type of applications require automatic discovery UDDI [3], composition WSDL [4] and execution SOAP [5].

This paper proposes a SaaS framework that provides facilities to develop web applications with a high degree of

This work has been funded by Junta de Extremadura, and cofunded European Regional Development Fund ((ERDF-GR10116)

978-1-4577-1127-5/11/$26.00 ©2011 IEEE

efficiency and reusability, for this reason the following technologies have been combined:

- SaaS technology supports access to software units as services.
- Web services are the selected middleware platform which provides accessibility to SaaS software.
- Web applications and components are built according to the component based model.
- RIA mechanisms increase the efficiency degree of web applications.

The rest of the paper is organized as follows: in section 2 related works are introduced. The component methodology is defined in section 3 and 4. The web service-based framework is introduced in section 5 and 6. Finally, conclusions are explained in section 7.

II. RELATED WORKS

Software companies have supported different projects in order to open new trends in web application development. OpenAjax Alliance is an organization of leading vendors, open source projects, and companies that are dedicated to the successful adoption of open and interoperable web technologies. OpenLaszlo [6] is an open source framework that provides the development of Rich Internet Applications. Besides, a new language (LZX) has been defined to describe web applications. Adobe Flex [7] is a highly productive, free, open source framework for building expressive web applications. Google Docs [8] is a new technology to work with online documents. Microsoft [9] has developed Windows Presentation Foundation (formerly code-named "Avalon"), which provides support for Web Browser Applications (WBAs). Sun Labs Lively Kernel [10] has implemented an interactive web programming environment that ran in a web browser. Other similar alternatives have been supported by the software community; Dojo Offline [11] is a cross-framework that enables web applications to work offline. The information generated by the user is stored inside the web browser and sent to the web server when the web connection is activated. In [12] it is proposed a web application model and a framework that are suitable for portable desktop environment. This framework packed web applications which are downloaded and installed in the user's computer. Other works have integrated new tools inside web browsers, POW [13] proposed to integrate a web server inside Firefox web browser. Finally, in [14] it is analyzed the limitations, challenges and opportunities related to web browser as an application platform.

Other works have proposed different SaaS applications frameworks. The responsibilities and the infrastructure of a Saas application [15], a SaaS framework from technical and business perspectives [16], and an integration of SaaS applications and software Cloud [17], have been proposed.

III. COMPONENTS

Component-based Software Engineering denotes the process of building software by (re)using pre-built software components. The main benefit of component-based software is the "time-to-market" [18], thus reducing the cost of developing the software. A software component is a unit of composition with contractually specified interfaces and explicit context dependencies. Additionally, a component is a software element that conforms to a component model and can be independently deployed and composed [19]. Component based software is developed interconnecting building blocks, therefore, a high degree of reusability and modularity is achieved by this type of applications [20].

In this paper, it is proposed a component model that provides mechanisms that support web components as services. According to the software as a service model, a web component is defined as a software unit, hosted in a server, which provides additional functionality for web applications.

Web components can be classified, according to the location where they are activated, as follows:

- **Client web component:** this type of web component is downloaded into the user's web computer and activated in the web browser.
- **Server web component:** it resides in the server computer and answers to client's requests. This type of web component is activated only at server side.
- **Hybrid web component:** it involves web components that reside in the server's computer, but at the same time, provide some functionality that must be downloaded and activated in the user's computer.
- **Framework web component:** this type of web component is part of the framework. Usually, frameworks that provide communication between remote elements require the installation of different modules in the user's computer. In contrast, the framework proposed in this paper does not require this installation process, since each time a web application is invoked, framework web components are downloaded and activated.

A. *Web component development*

The specification of a web component is defined in three layers:

- **Definition layer:** the name of the web component, according to a namespace, and a description is included in this layer. Moreover, the set of requirements to activate the web component are specified. A new language, called Web Component Definition Language (WCDL), has been defined, and it is an extension of XML with the addition of new elements to define web components.
- **Interface layer:** this layer defines how the data flows between the user and the web component. The information presented in the interface layer is classified as follows:

This work has been funded by Junta de Extremadura, and cofunded European Regional Development Fund ((ERDF-GR10116)

- In element: it denotes that some information must be introduced by the user and processed by the web component.
- Out element: it specifies that some information of the web component must be displayed to the user.
- In/Out element: it represents information that can be in and out at the same time.

Web Components can be displayed in different ways according to the environment where they are activated. As such, a web component can be attached to more than one interface, and each interface can be adapted to a concrete situation. This is the reason why the link between web components and interfaces is dynamic, that is, an interface is linked to a web component just before its activation.

- **Functional layer:** the algorithm that solves a specific problem is specified in the functional layer.

Figure 1 shows the structure of a web component.

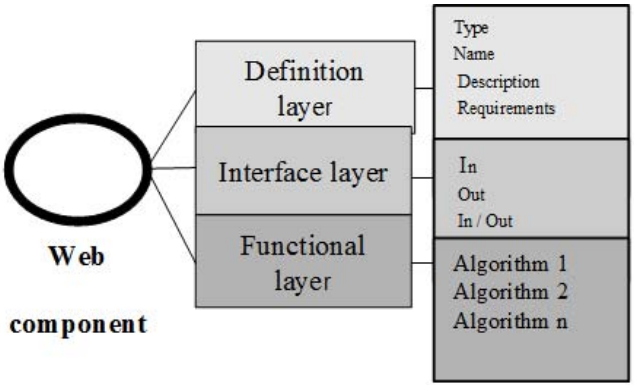

Figure 1. Web component structure

IV. WEB SERVICE-BASED FRAMEWORK

The web service-based framework proposed in this paper seeks to provide a new environment to develop web applications. The framework integrates component-based methodology and web services architecture, which provides a high degree of reusability, and RIA mechanisms, which increases the efficiency degree of web applications. The basic structure of the framework is shown in figure 2.

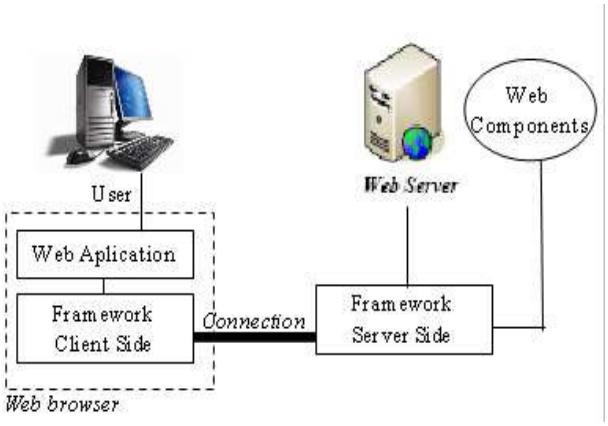

Figure 2. Framework basic structure

Web applications are executed inside user's web browser, and every time a web component is required, a connection defines the relationship between the web application and the web component. According to the type of the relationship, connections can be classified as follows:

- **Client link**: when this type of connection is required, the web component is downloaded and activated in the user's web browser. Client and framework web components require this type of connection
- **Server link**: this type of connection does not require the web component to be downloaded, in this case, interactions between web applications and web components are remote. Server web components are linked to this type of connection.

Hybrid web components require both type of connections because they imply local and server functionality. In the above figure, two different sides can be identified: the client side defines the elements that are activated in the user's web browser, and the server side, which provides operations to store and manage web components and applications. Both sides are explained in the following subsections.

A. *Framework client side*

This is the part of the framework that must be activated in the user's computer. The framework client side is divided into the following modules:

- **AJAX Engine Module:** In order to improve web services performance, different alternatives [22, 23, 24] have proposed to introduce Ajax technology in web service architecture. The integration of web services with AJAX technology provides asynchronous connections, which increases the efficiency degree of web applications because the client can continue its execution while the answer is being processed by the server. The AJAX engine

This work has been funded by Junta de Extremadura, and cofunded European Regional Development Fund ((ERDF-GR10116)

module provides operations to support asynchronous invocations.

- **Dynamic load module**: downloads web components when they are required and once a web component is completely downloaded, an event notification is activated
- C**ore module**: provides mechanisms to web component manipulation inside user's web browser. When a web component download notification is cached by this module, all its requirements are tested, if all of them are satisfied, the web component is activated and the interface is linked. In other case, the activation is delayed until every requirement can be satisfied.
- **Web Application module**: This is a container in which the source code of the web application is stored. Besides, this module is responsible for creating the connection between web components and web applications.

In figure 3 it is shown the structure of the framework client side. All these modules are not installed in the user's computer, as it has been mentioned, they are downloaded over Internet each time a web application is invoked.

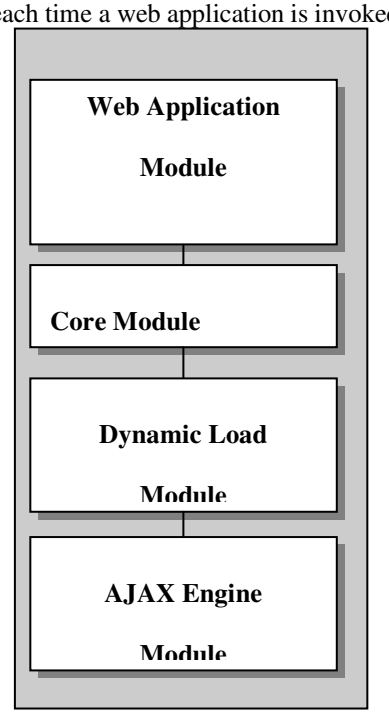

Figure 3. Framework client side

The process to activate and use a web application is described in the following steps:

- **Step 1: Web application reference.** Each time a web application is required, the user must introduce in the web browser the URL where the web application is hosted. This invocation activates, in the server side, the service that stars the activation of the web application.
- **Step 2: Web application download and activation.** The server returns the source code and the web application module stores and activates the web application inside the user's web browser.
- **Step 3: Web component download.** When a web component is required, a request is sent to the server.
- **Step 4: Web component activation:** When a web component is returned (client or hybrid web component) or linked (server web component), the core module decides to activate or not according to the evaluation of its requirements.
- **Step 5: Web component attachment.** Once a web component is activated, it is attached to the web application. Only when the attachment process has finished, the web component can be used.

In figure 4 it is displayed this process and the steps required to use a component-based web application.

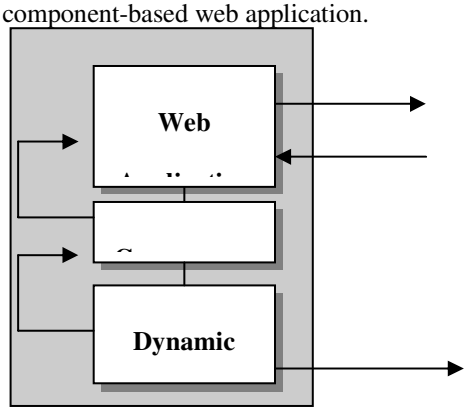

Figure 4. Web component-based application process.

B. Framework server side

This module provides operations to store, search and manipulate web components and web applications are defined in the framework server side, and are the followings:

- **Web Component Repository**: this is the repository where web components are stored.
- **Web Component Services:** provides web services to select, find and return web components. When a request is received in the framework server side, the web component is found in the component repository and returned to the user.
- **Web component Manager:** supports web component repository and offers operations to store, update and delete web components in the web component repository.
- **Web Application Repository**: web applications are stored in other repository called web application repository.

This work has been funded by Junta de Extremadura, and cofunded European Regional Development Fund ((ERDF-GR10116)

- **Web Application Manager Module:** this module provides services to select and return web applications.
- Web application manager: supports web application repository.

The relationships between web components (component link) are also specified in the web component repository, for this reason new metadata are included in order to define for each component link, the source and target web component, and also the type of the relationship (requirement, composition). Moreover, additional metadata define the relationship between web applications and web components (application link), and includes the set of web components required by a web application. The framework server side provides the following operations:

- **Web application activation**: denotes the set of services required to receive, process and activate a web application request. When a request is received by web application services, the source code of the web application is extracted from the web application repository, and sent back to the user.
- **Web component activation:** Every time a web component is required, a request is received and the web component service module is activated. The information of the web component is extracted from the web component repository and sent back to the user.

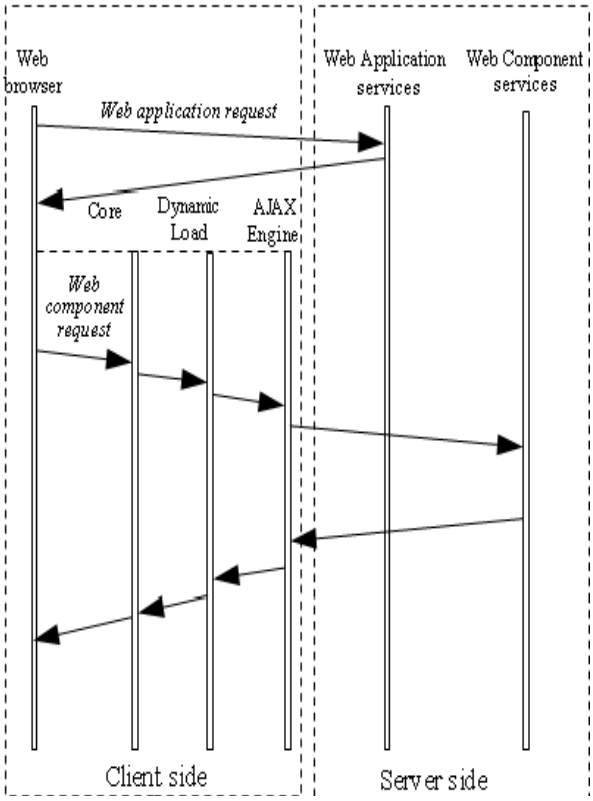

Figure 5. Web component-based application process.

As it is shown in figure 5, when a user introduces the URL of a web application, the web application services module is activated and the request is processed at server side. The web application is found in the web application repository and the source code is extracted and returned to the user. At the same time, the modules of the framework client side are also extracted and returned (core, dynamic link and AJAX engine). When this information is received in the user's web browser, the modules are built and activated inside the web browser. Finally, the web application source code begins to execute, and each time a web component is required, a request is sent to the server in order to activate the selected web component. According to the type of the web component, the activation can imply download and activate the web component in the user's web browser (client web component), obtain web component link in order to provide remote references (server web component), and in the case of hybrid web components, both actions must be performed.

The combination of different technologies in the framework proposed provides the following benefits.

- **High degree of efficiency:** RIA mechanisms provides asynchronous connections which increases the efficiency degree of web applications.
- **High degree of reusability:** component-based methodology allows web components to be reused in different web applications.
- **Installation and update are not required**: the Saas model avoids installation and update processes because software is always downloaded on demand.
- **Cross-platform**: web services support communication between different type of platforms. Moreover, as web applications are activated inside web browsers, they are independent from operating systems and hardware and software configurations.
- **Selective download:** web applications are not required to be completely download before they can be used, instead the framework provides mechanisms to download and activate web components only when they are required.

V. CONCLUSIONS

The World Wide Web is changing the way applications are developed. Now, web applications can be activated in a web browser, and no installation and update process are required. However when web applications become more complex, the necessity of developing a framework to support this complexity arises.

This paper proposes a framework to develop web applications according to the Saas model, component-based software engineering, web services architecture and RIA technologies. A web application is defined as a composition of web components, and web components are downloaded on demand and linked to web applications dynamically. A high degree of efficiency and reusability are achieved by component-based web applications.

This work has been funded by Junta de Extremadura, and cofunded European Regional Development Fund ((ERDF-GR10116)

REFERENCES

[1] Software & Information Industry Association, “Backgrounder: Software as a Service", February 2001

[2] Ning, J.Q: Component-based software engineering (CBSE). Assessment of Software Tools and Technologies, 1997., Proceedings Fifth International Symposium on Digital Object Identifier (1997).

[3] Bellwood, T., Clément, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y. L., Januszewski, K., Lee, S., McKee, B., Munter, J., von Riegen, C. “UDDI Version 3.0. Published Specification”, http://uddi.org/pubs/uddiv3.00- published-20020719.htm, 2002

[4] Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S., “WSDL Web Services Description Language (WSDL) 1.1”, http://www.w3.org/TR/wsdl, 2001.

[5] Box, D., Ehnebuske, D., Kakivaya, g., Layman, A., Mendel-sohn, N., Nielsen, H. F., Thatte, S., Winer, D., “Simple Object Access Protocol (SOAP) 1.1”, http://www.w3.org/TR/SOAP/, 2000

[6] http://www.openlaszlo.org/

[7] http://flex.org/

[8] http://docs.google.com/

[9] http://msdn.microsoft.com/es-es/windows/ff798266.aspx

[10] http://labs.oracle.com/projects/lively/

[11] http://dojotoolkit.org/offline

[12] Ki-Hyuk Nam, Ki-Seok Banh, Wan Choi: A method for distributing web applications. IEEE International conferene on advanced communication technology. ICACT 2008

[13] http://davidkellogg.com/pow/

[14] Taivalsaari, A.; Mikkonen, T.; Ingalls, D.; Palacz, K : Web Browser as an Application Platform. Software Engineering and Advanced Applications, 2008. SEAA '08. 34th Euromicro Conference

[15] Guoling Liu; , "Research on independent SaaS platform," *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on* , vol., no., pp.110-113, 16-18 April 2010.

[16] Kai Tang; Jian Ming Zhang; Zhong Bo Jiang; , "Framework for SaaS Management Platform," *e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on* , vol., no., pp.345-350, 10-12 Nov. 2010

[17] Feng Liu; Weiping Guo; Zhi Qiang Zhao; Wu Chou; , "SaaS Integration for Software Cloud," *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on* , vol., no., pp.402-409, 5-10 July 2010

[18] Updating a Modular Product: How to Set Time to Market and Component Quality Lifang Wu; De Matta, R.; Lowe, T.J.;Engineering Management, IEEE Transactions on Volume: 56 , Issue: 2

[19] George T. Heineman and William T. Councill: Component-Based Software Engineering Putting the Pieces Together. Addison-Wesley, Boston, MA ,880, June 2001.

[20] Repenning, A.; Ioannidou, A.; Payton, M.; Wenming Ye; Roschelle,J.: Using components for rapid distributed software development. Software, IEEE Volume: 18 , Issue: 2 Publication Year: 2001 , Page(s): 38 - 45

[21] http://www.w3.org/TR/ws-arch/

[22] Using AJAX to Optimize Web Services Performance ZhenXing Chen;Computational Intelligence and Software Engineering, 2009. CiSE 2009. Publication Year: 2009 , Page(s): 1 - 4

[23] Design of a Web-Based Building Management System Using Ajax and Web Services Jianbo Bai; Hong Xiao; Tianyu Zhu; Wei Liu; Aizhou Sun. Business and Information Management, 2008. ISBIM '08. Publication Year: 2008 , Page(s): 63 - 66

[24] Ajax and Web Services Integrated Framework Based on Duplicate Proxy Pattern. Yifu Gan; Huirong Yang; Information Technologies and Applications in Education, 2007. ISITAE '07. Publication Year: 2007 , Page(s): 297 - 302

This work has been funded by Junta de Extremadura, and cofunded European Regional Development Fund ((ERDF-GR10116)