2007 4th International Conference on Electrical and Electronics Engineering (ICEEE 2007)
Mexico City, Mexico. September 5-7, 2007

From Desktop Applications Towards Ajax Web Applications

J. Sergio Zepeda, Sergio V. Chapa.
Department of Computer Science. CINVESTAV
Av. Instituto Politécnico Nacional 2508. 07300 México D.F., México
E-mail:jsergy@yahoo.com, schapa@delta.cs.cinvestav.mx

***Abstract*— Ajax is a set of different technologies that work together to create new and powerful Web applications. Ajax is demonstrating its usefulness in real world applications. The most important Internet companies as: Google, Yahoo, Amazon, Microsoft, are developing rich Web applications based on Ajax.**
Many developers do not know how to use these technologies to build Ajax Applications.
In this paper, we present an overview about Ajax. Here, we discuss the term Ajax and the technologies used. Also, we show how Ajax is working inside, and how the technologies work together to achieve a rich behavior.

Keywords: Ajax, Web, Web 2.0, Web applications, Internet, Information System, Software Engineering.

I. Introduction

Recently in February 2005, the term *Ajax (Asynchronous JavaScript and XML)* was coined by Jesse James Garret [1], [2], [3]. The term Ajax attracted the interest of Web developers community, because after that the essay [3] appeared, there were plenty of comments, discussion and an explosion of questions about how to use this technique, its capabilities and shortcomings.
The strongest demonstrations of the usefulness and popularity of this technique in web design with real Web applications are: *Google Maps, Google Suggest, Gmail, Yahoo Mail,* and more recently *Windows Live*. Their main attraction is a widely rich interaction with the client-side, very similar a desktop application [2].
Ajax applications give certain benefits over desktop applications because these are independent of : specific operative system, virtual machine, plug-ins in the browser, or external programs to install in the user's desktop computer. These features attracted a strong interest of Internet users, and now are very populars. These applications only need a modern browser with access to Internet in client-side, and offer easier support, faster, responsive and rich user interaction.
The possibility of move desktop applications towards Ajax Web applications is now a reality, *Google Docs* is a good example about this fact. With *Google Docs*, you can write documents or you can use a spreadsheet through a browser. With all these advantages that Ajax applications offer, there is a big interest in knowing how to achieve this behavior. The term Ajax is not clear for some people and there are wrong ideas about it. Some people think that it is a new language, or new technology, or new product, or only other new framework. But, all those ideas are not true. Some papers discuss the Ajax term, but not in sufficient detail.
In this paper, we discuss the term with more detail, and we show how Ajax is working inside a Web application. In summary, this is an overview that analyze all parts involved in a Ajax applications.
This paper is organized as follows. Section 2, gives an explanation about the term Ajax and its real concept.
Section 3, describes the synchronous communication used for classical Web applications, and asynchronous communication used for Ajax Web applications.
Section 4, we show how the technologies are integrated, to produce that Ajax applications can have similar behavior to desktop applications. Section 5, presents some features of Ajax applications. Finally; in section 6, we present our conclusions.

II. Ajax

Ajax is not a technology and not a new programming language. Ajax is a new way of think, design, develop, and a new style to programmer Web applications.
Also, Ajax is a new technique that uses a set of open standards technologies, with support by cross-browser and cross-platform compatibility [4]. These are not untested or new technologies, instead; these technologies are mature, stable, widely-used and well-known.
The technologies work together in different levels, each one with specific functionality to create a powerful new development model for Web Applications [5]. These technologies, levels, languages and protocols are:

- The Hypertext Transfer Markup Languages: *HTML, XHTML, XML.*
- Language to describe the presentation of a document: *Cascading Style Sheets (CSS).*
- Interaction and dynamic display: *W3C Document Object Model (DOM).*
- Client-side language and parse data: *JavaScript.*
- Asynchronous data transfer and communication with the server : *XMLHttpRequest object (XHR).*

IEEE Catalog Number: 07EX1762C
ISBN: 1-4244-1166-1
Library of Congress: 2007923398

- Manipulation, transformation and data exchange: *XML, XSLT, HTML, JSON, Plain Text.*
- Transfer Protocol: *HTTP, HTTPS.*
- Server-side languages: *JSP, JSF, Perl, Ruby, PHP, Cold Fusion, ASP, and CGI applications.*

These are not the unique technologies and languages, but are the most usually used, more populars, and open standards with free use; with exception in the last point with Server-side languages as Cold Fusion, ASP and ASP.NET. In the point client-side languages, there are others, but those are proprietaries and are not interesting in this paper. The Ajax web applications can have different combinations to be created, because select only one in each level is necessary. This way is easy to see that Ajax is not tied to specific programming language, specific data exchange format, or specific Server-side language.
In summary, Ajax uses a combination of established Web technologies. It is the combination of these technologies that makes Ajax unique on the web [6].
Ajax applications can be programmed with only a text editor optimizing your code, or if you prefer you can use an expensive and complicate framework and add thousands of code lines sometimes which are unnecessary.

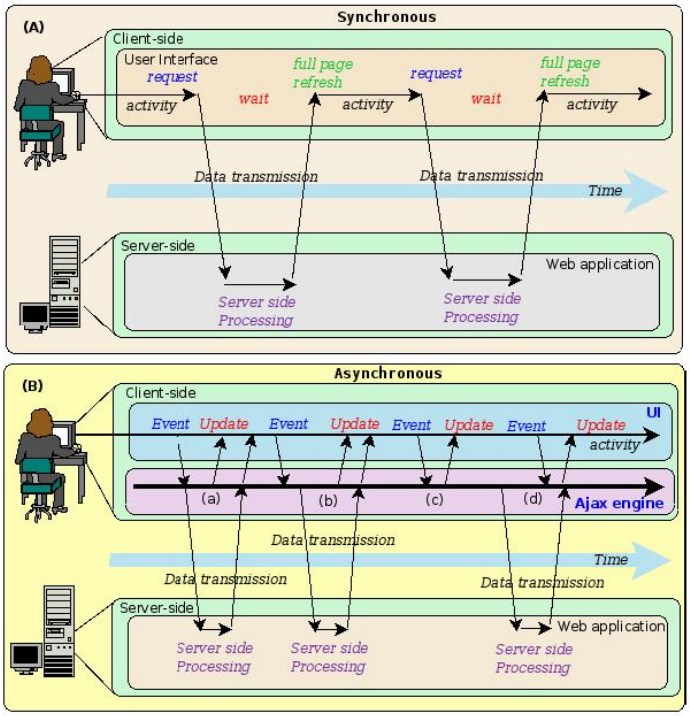

Fig. 1. The synchronous interaction vs the asynchronous interaction.

III. Communication

A. Synchronous Communication

The synchronous communication is enough for Web in the majority of cases. The publication of information in static pages is a good example. But, in some Web Applications where more interactivity and fast responses with the users are necessary, it is not a good solution.
Classical Web applications are based on a multi-page interface model [8], in which interactions are based on a page sequence paradigm [7].
The classic Web applications usually require that the user sends a request to the server through a link or form, as the figure 1 (A) is shown in synchronous case. We need to wait for the server-side processing; for some cases a dynamic creation of a new page is necessary [8], after; the server returns a new page to the client browser and it is update with full page refresh and new results. The figure 1 shows, how the user activity is broken constantly, and the update requires reload completely the Web page for every piece of new data [2]. The classic model creates many problems in Web applications due to slow performance, loss of states, excessive bandwidth, limited interactivity, and the data transmission have redundant code which is unnecessary in each page.
This request-wait-response-wait pattern is extremely disruptive and lowers productivity [1]. Also, regularly the pages have HTML, CSS and JavaScript mixed in a single file, it can generate troubles when some modifications are necessaries.

B. Asynchronous Communication

This new model consists of a single page web interface. This interface is composed of individual and independent components, which can be created, updated, deleted and replaced independently [6].The access to these components is with DOM specification. The update can be realize with small parts of XML code through an Ajax engine, without reloading a new web page. The figure 1 (B) shows, the asynchronous communication for Ajax engine, this section is separate in four cases:

- a) When user generates an event for the Ajax engine and it sends directly a request to the server side, but some updates can be realized while the server gives a response. When the server returns the new data, the independent component is updated.
- b) When user generates an event for the Ajax engine, and it sends indirectly a request to the server. After, the process it is the same as the case (a).
- c) Here, the user generates events, but is not necessary to send a request to the server. JavaScript can update the components with CSS, DOM and data of the same page through Javascript functions, without that Ajax engine sends some request.
- d) Before the user generates a specific event, the Ajax engine sends a request to the server, when the user gives a specific event, the information will be immediately updated.

The Ajax engine allows the communication in background with the server of asynchronous way, it sends and retrieves data for processing from server-side.

1-4244-1166-1/07/$25.00 ©2007 IEEE.
IEEE Catalog Number: 07EX1762C
ISBN: 1-4244-1166-1
Library of Congress: 2007923398

In this way it is possible to give an answer immediately to the client side. It seems like adding a layer to the application would make it less responsive, but the opposite is true [3].

Ajax eliminates the constant reloading of entire Web pages on each user interaction, such as in the classical Web applications model. With asynchronous communication, the data transmission is more efficient for some Web applications, because the update is only with a few data in specific components. Therefore, there are more rich interactivity, more responsive and faster user interfaces with less waiting. All the user interactions are directly within the browser.

With this new way to create Web applications, enable to the pages to behave more like desktop applications.

The designers and developers need to change the way of think in Web applications with a single page interface. The main idea in this model is the interaction very similar to desktop application, but the programming is with Web Applications technologies.

IV. Towards the Inside of Ajax Web Applications

In this section, we discuss in detail how the technologies work together in a new way. Ajax application in the client-side is formed for three layers. The layer 1 is the user interface, the layer 2 is JavaScript code and inside of it, the layer 3 with Ajax engine.

The layer user interface involve to XHTML, CSS and DOM. XHTML has the content of the page through tags. CSS offers the separation of content and presentation in the page through style documents with elimination of redundancy. The CSS rules are in different files to achieve the separation. DOM specification offers the way of dynamic access, update the content, structure, style, and elements of a page. When all is loaded in client-side, EventListener and XHR are created and active in the web page. The user interaction generates events, these events are caught with a JavaScript function called EventListener. The function catches the events, and if the events have special code to execute, then the events are processed with their respective JavaScript function. The figure 2 shows two cases. In the first case; EventListener sends the data to update functions, these are functions previously programmed to update the content, presentation, or structure of XHTML through DOM and CSS. In this case, there is not a external request, the processing is only in client-side, and the changes are shown immediately.

In the second case, EventListener sends the data to XHR object. The XHR object is created in different way depending of the browser. Internet Explorer uses a special class called ActiveXObject, while Firefox and Safari use the XMLHttpRequest object. Although these classes have different constructors, their behavior is the same.

The XHR object sends a request of asynchronous way, this request is through HTTP or HTTPS protocol.

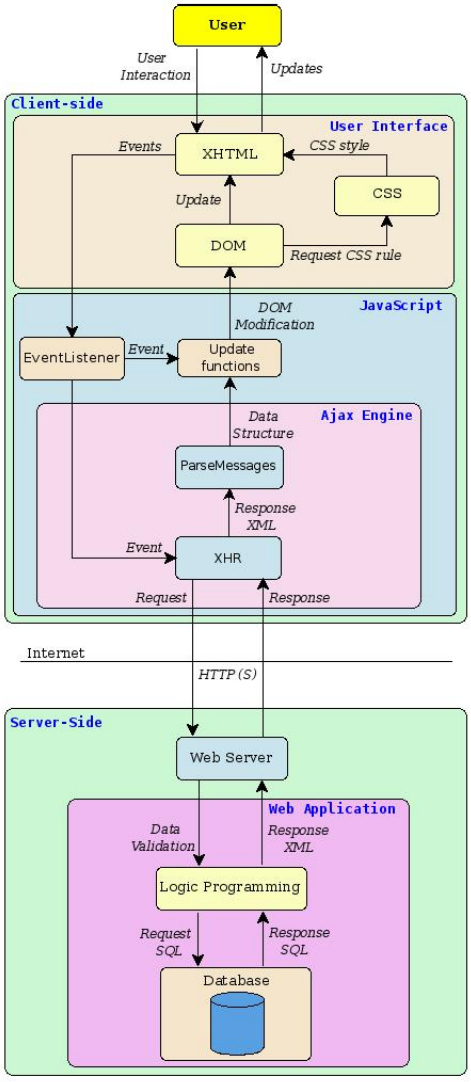

Fig. 2. Ajax Web Application Model

While HTTP request is processed in the server-side, its progress is indicated by changes in the readyState property. The property has five states: *uninitialized, loading, loaded, interactive and completed* [9]. The XHR object tells us about each change in the state with an *readystatechange* event.

When the server-side is processing the request of web application the data validation is necessary. This is very important when the request needs to access the database. The web applications and databases can be protected by realms of access, user roles, passwords, etc.

The database returns new data to Web application, the data are transformed in a specific format (XML, HTML, plain text, etc). The Web application sends the response to XHR object in client-side. When the response is over, the readystate property changes to *completed* state. Now, the XHR returns the response to JavaScript functions to parse the content.

IEEE Catalog Number: 07EX1762C
ISBN: 1-4244-1166-1
Library of Congress: 2007923398

The data transformation is with specific structure to be incorporate to the page. The update functions add the new data through DOM and CSS. In this model it is easy to have an optimized code, and it can be separated in small JavaScript files with specific functionality.

V. Features

A. Features

Ajax changes the way of development in Web applications. Google maps presents huge interest by web developers, because it shows that Ajax is practical for real-world applications. Recently, there are several researches about Ajax, but the more interesting is the application to research as in [6], and [9]. Now, we show some advantages and disadvantages of Ajax Web applications.

B. Advantages

- Cross-Browser and Cross-Platform Compatibility.
- Ajax application can be build with open standards technologies, is not tied to proprietary software.
- Lower cost of development.
- The code can be optimized and separated.
- More rich interaction with the user, by example: autocomplete, drag and drop, transparency, shading, Z-ordering, etc.
- More speed, and less wait time for screen updated.
- Widespread adoption of Ajax by industry leaders.
- Integration with others proprietaries technologies.

C. Disadvantages

- Same security problems as in classic Web applications.
- Think of different way in the development and design.
- The use of frameworks can generate complexity.
- More interaction through JavaScript.
- More code for old browser.
- Special code for some functions of Internet Explorer.

D. Security

The security in Ajax is an interesting topic for many developers. The main risk such as in all the Web applications are common mistakes in design and develop. The necessity of knowing each programming languages used in the develop is essential to avoid these mistakes.
The security in Ajax is part of new topics for research [10], [11], as well as its integration with Web Services [13]. The object XHR is the heart of Ajax Applications, but it is created with JavaScript. For some developers it is a big problem, because the code JavaScript is visible in the browsers. In some bad develops, JavaScript is used in validation of data only in client-side. The big problem is in the server-side when there is no validation on data. This way there are exposition of vulnerabilities like a SQL injection. Other problem with inexperience programmers is when they make code for client-side and they write the user, password, server, port of access to the server-side. Some applications collects of different servers special information to add to its page, but if the access to the servers is from the client-side instead of server, the code can be changed with malicious code. These are some the common mistakes in the programming. The request sends by XHR should be of the same way as a common Web application. The validation and analysis of all data receive in server-side is essential to avoid holes in the security.

VI. Conclusions

E-mails, calendars, spreadsheets, information managers, and word processors, are migrating of the traditional installation of programs in local desktop, towards Ajax Web applications. These Ajax applications have the similar interactivity and speed as desktop programs. The Ajax Web applications are producing radically changes towards a new Internet age. New, better, useful and successful applications based on Ajax, are near. It is necessary to understand and have some research how Ajax is working inside its new paradigms. Undoubtedly, there are many question to solve, but this is only the beginning of a new kind of Web applications.

VII. Acknowledgments

We thank Mridul Nandi for his proofreading and valuable comments.

References

[1] Keith Smith. Simplifying Ajax Style Web Development. *Computer*, Vol.39, no.5, pp. 98-102, May. 2006.
[2] Linda Dailey Paulson. Building Rich Web Applications with Ajax. *Computer*, Vol.38, issue 10, pp. 14-17, Oct. 2005.
[3] Jesse James Garret. Ajax: A New Approach to Web Applications. *http://www.adaptivepath.com/publications/essays/archives/000385.php*.
[4] Eric J. Bruno. Ajax: Asynchronous JavaScript and XML. *Dr. Dobb's Journal*, Vol.31 (2), pp. 32-35. Feb. 2006.
[5] Joe Hewitt. Ajax Debugging with Firebug. *Dr. Dobb's Journal*, no.393, pp. 22-26, Feb. 2007.
[6] Ali Mesbah, Arie van Deursen. Migrating Multi-page Web Applications to Single-page Ajax Interfaces. *11th European Conference on Software Maintenance and Reengineering*, pp. 181-190, March. 2007.
[7] Ali Mesbah, Arie van Deursen. An Architectural Style for Ajax. *Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA'07)*, pp. 44-53, Jan. 07.
[8] Jimm Challenger, Paul Dantzing, Arun Iyengar, Karen Wittig. A Fragment-Based Approach for Efficiently Creating Dynamic Web Content. *ACM Transactions on Internet Technology*, Vol.5, no.2, pp. 359-389, May. 2005.
[9] David Perelman Hall. Ajax and Record Locking. *Dr. Dobb's Journal*, Vol. 31, no.10, pp. 45-51, Oct. 2006.
[10] Michael Sonntag. Ajax Security in Groupware. *Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 472-479, Aug. 2006.
[11] James F. Ryan, Blair L. Reid. Usable Encryption Enabled by Ajax. *Proceedings of the International conference on Networking and Services*, 116, July, 2006.
[12] Conor Seabrook. Bringing the Desktop Application to the Web. *Dr. Dobb's Journal*, pp. 46-49, March. 2007.
[13] Ahmet Sayar, Marlon Pierce, Geoffrey Fox. Integrating Ajax Approach into GIS Visualization Web Services. *International Conference on Internet and Web Applications and Services*, AICT/ICIW 2006.
[14] Danny Ayers. From Here to There. *IEEE Internet Computing*, Vol.11, Issue 1, pp. 85-89, Jan.-Feb. 2007.

IEEE Catalog Number: 07EX1762C
ISBN: 1-4244-1166-1
Library of Congress: 2007923398