

Use of asynchronous JavaScript and XML for Comparative Market Analysis

Germán Cáseres¹, Lisandro Delia¹, Pablo Thomas², Hugo Ramón², Rodolfo Bertone²

{gcaseres, ldelia, pthomas, hramon, pbertone}@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática (III-LIDI)³ - Facultad de Informática - UNLP.

Abstract

Generally, the interactivity provided by web applications is carried out through surfing, as a result of a server's response at the request of a workstation. This is an essential characteristic of this type of software, although in some circumstances it turns into a functional and operative constraint. This paper solves the specific problem of a Real Estate CRM (Customer Relationship Management) by means of a web application, in which, in order to fulfill certain functional requirements, a similar desktop system interactivity is provided. To do this, the use of standard technology (PHP-MySQL) with asynchronous JavaScript and XML is provided, thus overcoming the stakeholders' expectations to carry out two functional requirements: first, the application must be web oriented and second, it must offer desktop system interactivity.

Key words:

Web application - AJAX (Asynchronous JavaScript and XML) - CMA (Comparative Market Analysis)

1. Introduction

When the outcome of a Requirement Process [6] of a Software System leads to the development of a distributed application [1], it is important to decide whether this application will be a desktop or web oriented application. A key issue in this decision, although not the only one, has to do with the physical distribution of the equipments with access to it. Generally, a Desktop system will be used, if these equipments are within the same building. However, the use of WEB oriented applications is not only convenient for customers located in distant places, but also for customers with different equipments or operative systems which are incompatible among them.

A WEB application is a system with access through a browser connected to a WEB server through Internet or Intranet, fulfilling a user request [8].

The main advantages of these types of applications are the installation, updating and maintenance usefulness, since its core is located in the server, and clients only have a WEB browser who is able to receive and to display information quickly.

The graphic interface of a WEB application is restrained by the browser. That is the reason why it provides a lesser degree of interactivity than desktop applications. Nowadays, however, the majority of the browsers use interpreted languages to execute certain routines in the client making the interface more functional without the need of the server to see the results.

¹ Becario III-LIDI. Facultad de Informática. UNLP

² Profesor Adjunto. Dedicación Exclusiva. Facultad de Informática. UNLP

³ Instituto de Investigación en Informática. Calle 50 y 115 1º Piso La Plata .Facultad de Informática. UNLP

These types of applications are divided at least into three layers, some of them optional according to the type of system to be developed. [Picture 1]:

- **WEB browser (first layer):** It is executed in the client, sending requests to the second layer and waiting for a response to show the result on the screen.
- **WEB server / Dynamic WEB engine (second layer):** WEB server receives a requirement from the first layer and responds with the required content. In case the content is dynamic, the engine (PHP, CGI, etc.) generates such content taking into account information from consultations with the database.
- **Database engine (third layer):** It executes the consultations made in the second layer and returns the information; it can operate in the same server than the Web engine.

Programming languages used for the implementation of WEB applications are varied, among others PHP, ASP/ASP.NET (more than a language it is a development architecture), JSP, Per, Ruby, Python are found.

As regards the most used database engines SQL Server, Oracle, DB2, MySQL, PostgreSQL are found.

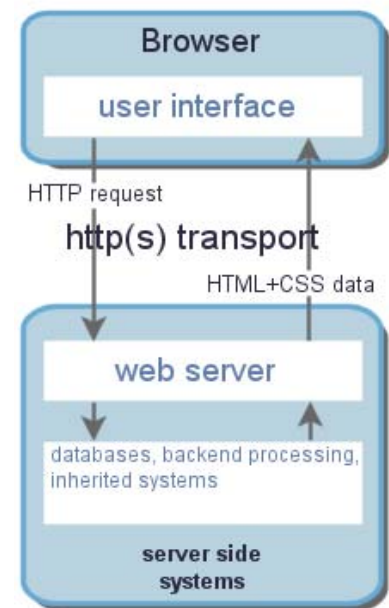
It is important to consider that in the first layer, executed in the client, the most used browsers should be able to represent correctly the WEB content which is important to display. On the other hand, it should be considered that nowadays every WEB browser supports a scripting language that executes code fragments in the client performing certain operations that add more dynamism to a WEB page without performing new requirements for the server. Among them JavaScript, ECMAScript and Jscript are found.

As regards Database language and engines above mentioned, it is possible to select different combinations to develop an application. A much used combination is PHP – MySQL – JavaScript which derives from the traditional LAMP (Linux-Apache-MySql-Php). The main reason for this election is the Free and Open Source nature of both, PHP [10] [5] and MySQL [2] [12], as well as the support given by the amount of applications developed through this technology [11]. Another advantage is the portability of both implementations, since they can operate within Linux or Windows servers, among others, without complications of any kind.

Nowadays, steady versions of PHP and MySQL have the necessary assistance to perform a complete WEB application without resorting other development tools. Besides, the huge amount of available data and support makes them a valid option.

In the case of JavaScript [7], the language was developed by Netscape, it was later adopted as ECMA Standard (European Computer Manufacture's Associations) in 1997 [17] under the name of ECMAScript. Finally it turned into ISO standard [22].

Microsoft has its own implementation of the standard called JScript [18]. Although JavaScript and JScript in certain cases become incompatible due to differences with the ISO standard, it is necessary to adapt to them so that compatibility does not diminish with two of the most widespread browsers in the market: Microsoft Internet Explorer and Mozilla Firefox (ex Netscape).



Standard WEB Application

Picture 1

This work is divided into seven sections, being the first an introduction to standard WEB applications and different implementation technologies. On the second section there is a description of a real WEB oriented application and a summary of its main purposes. The third section describes the application's main technical characteristics and an overview of its implementation using standard techniques. The constraints in the use of standard implementation methods are represented in the fourth section and an alternative solution is presented in the fifth section motivating a different way to develop web applications promoting a new architecture design. The last two sections are a conclusion about the most important decisions taken in the developing process of the presented WEB application, and a summary of future improvements that will be applied to the application.

2. Presentation of a real case: Area6 Real Estate Agent

Area6 is a web-oriented Real Estate CRM (Customer Relationship Management) [13], which operates in Spain. It controls the administrative circuit, from the entry of a real estate for sale to the sale operation, recording all the activities performed during the process. The system allows recording a detailed report of all the characteristics of a real estate (plot size, built area, rooms size, among others), since such information will be later used with statistical and comparative ends.

Due to the fact that Area6 is a Spanish CRM, all screen captures of this application are in that language since the requirements for its development did not involve any kind of internationalization.

The real estates data entry to the system is not only executed when the client wants to sell a property. This process is executed in every moment, so a detailed record of all the existing real estates in the market is found. Given that Area6 is a CRM, its purpose is not only optimizing the sale process but also offers a service for the client to make it valid. Besides the information of real estates on sale and their prices, this service also estimates, as objectively as possible, the sale price of any property for sale. This process is called CMA (Comparative Market Analysis) [19].

Generally, a client contracts an assessor if s/he wants to sell her/his property to calculate its price. In this particular case this process is not necessary, since the system performs the task "objectively".

Basically, the production of a CMA consists of the selection of a property ("property in consideration") and then the selection of a set of properties ("marker properties"), which, according to the real estate agent's opinion share similar characteristics and location. This is represented in pictures 2 and 3.

Propiedades

>>

<<

Número de Página: 1 de 29

Mostrar 10 filas

Administrar habitaciones

<input type="checkbox"/>	Provincia	Ciudad	Calle	Número	Tipología	Estado	Fecha	Estado	Precio Actual	Sup. Total	Const	Acción
<input type="checkbox"/>	Madrid	Galapagar	Encina	9B	Aislado	Activa	01/06/2004		590000.00	383.06		
<input type="checkbox"/>	Madrid	Torrelodones	Emilia Alarcos	17	Adosado	Activa	30/09/2005		615000.00	266.67		
<input type="checkbox"/>	Cádiz	Tarifa	Helechos	15	Aislado	Prospectiva	01/12/2005		75000.00	0.00		
<input type="checkbox"/>	Madrid	Torrelodones	Estanque	8	Aislado	Pendiente	25/11/2005		790000.00	327.15		
<input type="checkbox"/>	Madrid	Torrelodones	Loma	53	Aislado	Activa	04/08/2003		596900.00	238.07		
<input type="checkbox"/>	Madrid	Las Rozas	II	14	Aislado	Activa	24/05/2005		580900.00	256.74		

Picture 2: Selection of the property in consideration

Resultados de la búsqueda			
<< Volver Cantidad de propiedades: 235			
<input type="checkbox"/>	Estado	Domicilio	Barrio
<input checked="" type="checkbox"/>	Expirada	Tampico N° 08, Barrio El Gasco	El Gasco
<input type="checkbox"/>	Expirada	Los Pozos N° 08, Barrio Las Minas	Las Minas
<input type="checkbox"/>	Activa	Emilia Alarcos N° 17, Barrio Las Marias	Las Marias
<input type="checkbox"/>	Activa	Loma N° 53, Barrio Los Peñascales	Los Peñascales
<input checked="" type="checkbox"/>	Activa	II N° 14, Barrio Las Matas	Las Matas
<input type="checkbox"/>	Activa	Nalon N° 64, Barrio La Berzosa	La Berzosa
<input type="checkbox"/>	Activa	Las Minas N° 15, Barrio Las Minas	Las Minas
<input type="checkbox"/>	Activa	Covadonga N° 27, Barrio La Berzosa	La Berzosa
<input type="checkbox"/>	Activa	Alcotán N° 09, Barrio Molino de la Hoz	Molino de la Hoz

Picture 3: Selection of marker properties

Then each marker property is analyzed in relation with the property in consideration by means of two tasks:

- Comparison of the surface area of each floor.
- Comparison of other characteristics (property location, maintenance, proximity to certain places, among others)

-	Encina N° 9B, Barrio Los Jarales	757	603,900.00	-133,490.63	470,409.37	1,576.52	Activa
-	Ajustes por metros cuadrados						
Planta	Sup. Estudio	Sup. Testigo	Diferencia	K	Ajuste	Acción	
1 - Planta de Piso	98.40	0.00	-98.40	1.00	155,129.12		
0 - Planta de Piso	132.22	157.04	24.82	1.00	-39,129.11		
Total Sobre Rasante	230.62	157.04	-73.58	1.00	116,000.01		
-1 - Semisótano	36.05	226.02	189.97	1.00	-299,490.64		
Total Bajo Rasante	36.05	226.02	189.97		-299,490.64		
Total construida	266.67	383.06	116.39		-183,490.63		

Picture 4: Adjustments per square meter

-	Ajustes por condiciones estándar	Nuevo +
Condicion Estándar	Valor	Acción
Acabados Peor en Comparable	50,000.00	
Mantenimiento Mejor en comparable	-50,000.00	
-	Ajustes por condiciones no estándar	Nuevo +
Condicion no estándar	Valor	Acción
Seguridad Privada	2,500.00	

Picture 5: Adjustments per standard/non-standard condition

In the first comparison shown in **picture 4**, differences per square meter are calculated, the value being calculated according to the sale price of the marker property. The agent can later adjust the value using a “homogenizing factor” that corrects other differences among the properties.

In the second comparison shown in **picture 5**, it is the agent who calculates to what extent other characteristics will affect the sale price. This step is necessary due to the fact that properties can be similar but not identical.

Finally, the sale price of a property in consideration is calculated finding the average of the **adjusted** sale price of each marker properties.

Therefore, it is possible to calculate the sale price of a property by means of a comparative analysis with other available properties in the market. What is important at this point is that the sale price obtained is well supported, being then an “objective” market price.

Summing up, a CMA can be divided into three stages:

- Establishment of a property to study (property in consideration)
- Selection of a set of “similar” properties with similar characteristics with those of the property in consideration (marker properties).
- Comparison of each marker property, making the necessary adjustments to get the most objective comparison (**Picture 6**)

Propiedad Objeto del Estudio:

Domicilio:	Emilia Alarcos Nº 17, Barrio Las Marias
Tipología:	Adosado
Categoría:	Residencial Unifamiliar
Superficie total:	266.67
Superficie sobre rasante:	230.62
Superficie bajo rasante:	36.05

Propiedades testigo:

Propiedad	DEM	Precio	Ajuste	Precio Ajustado	Importe por M2	Estado
- Encina Nº 9B, Barrio Los Jarales	757	603,900.00	-133,490.63	470,409.37	1,576.52	Activa
Ajustes por metros cuadrados						
Planta	Sup. Estudio	Sup. Testigo	Diferencia	K	Ajuste	Acción
1 - Planta de Piso	98.40	0.00	-98.40	1.00	155,129.12	
0 - Planta de Piso	132.22	157.04	24.82	1.00	-39,129.11	
Total Sobre Rasante	230.62	157.04	-73.58	1.00	116,000.01	
-1 - Semisótano	36.05	226.02	189.97	1.00	-299,490.64	
Total Bajo Rasante	36.05	226.02	189.97		-299,490.64	
Total construida	266.67	383.06	116.39		-183,490.63	
Ajustes por condiciones estándar Nuevo +						
Condicion Estándar					Valor	Acción
Acabados Peor en Comparable					50,000.00	
Ajustes por condiciones no estándar Nuevo +						
Condicion no estándar					Valor	Acción
- Loma Nº 53, Barrio Los Peñascales	1,059	596,900.00	71,707.23	668,607.23	2,507.25	Activa
Ajustes por metros cuadrados						
Planta	Sup. Estudio	Sup. Testigo	Diferencia	K	Ajuste	Acción
1 - Planta de Piso	98.40	0.00	-98.40	1.00	246,712.98	
0 - Planta de Piso	132.22	185.04	52.82	1.00	-132,432.72	
Total Sobre Rasante	230.62	185.04	-45.58	1.00	114,280.26	
-1 - Semisótano	36.05	53.03	16.98	1.00	-42,573.03	
Total Bajo Rasante	36.05	53.03	16.98		-42,573.03	
Total construida	266.67	238.07	-28.60		71,707.23	
Ajustes por condiciones estándar Nuevo +						
Condicion Estándar					Valor	Acción
Ajustes por condiciones no estándar Nuevo +						
Condicion no estándar					Valor	Acción

Tipo de cálculo:

☒ Por M² ☒ Por condiciones estándar ☒ Por condiciones no estándar

Porcentaje de margen: %

Estado	Nº	DEM	Precio		
Activa	2	908	668,607	420,409	544,508
	Total	Promedio	Más cara	Más barata	Promedio Gral
Resumen	2	908	668,607	420,409	544,508

Picture 6: General view of a CMA

3. Implementation of Area6 Real Estate Agent

The development of Area6 was a very particular process, in particular the requirement elicitation [6], because the stakeholder was distant. Some of these special characteristics were connected with the stakeholder availability, being in contact through Skype version 1.2.0.48 (20). Besides, his wide experience over the problem domain, added to the use of AgenOffice software version 7.0 (21) as a model to requirement reuse, allowed elaborating a SRS (Specification Requirement Software) based on consistent IEEE/ANSI 830 (3) (4).

As for the development itself, a combination of PHP - MySQL - Java technologies was adopted due to their above mentioned advantages.

Most of the system is based on operation Add/Delete/Modify combination known as ABM (Alta/Baja/Modificación) unit (**picture 7**).

Filtrado de datos

País: C. Autónoma: Provincia: Ciudad: Barrio: Calle:

Ap Dueño: Nom Dueño: Categoría Propiedad: Tipología: Estado:

Fecha Estado: Entre -- y --

Detallada: ☐

Propiedades

Número de Página: 1 de 20 Mostrar 10 filas

Administrar habitaciones

<input type="checkbox"/>	Provincia	Ciudad	Calle	Nro	Tipología	Estado	Fecha	Estado	Precio Actual	Sup.	Total	Const	Acción
<input type="checkbox"/>	Madrid	Galapagar	Encina	98	Aislado	Activa	01/06/2004		603900.00	383.06			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Torrelodones	Emilia Alarcos	17	Adosado	Activa	30/09/2005		615000.00	266.67			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Cádiz	Tarifa	Helechos	15	Aislado	Prospectiva	01/12/2005		75000.00	250.00			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Torrelodones	Estanque 8		Aislado	Pendiente	25/11/2005		790000.00	327.15			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Torrelodones	Loma	53	Aislado	Activa	04/08/2003		596900.00	238.07			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Las Rozas	II	14	Aislado	Activa	24/05/2005		580900.00	256.74			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Galapagar	Las Minas	15	Aislado	Activa	19/01/2005		750000.00	414.09			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Las Rozas	Alcotán	09	Aislado	Activa	21/06/2005		900000.00	434.91			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Hoyo de Manzanares	Asturias	70	Aislado	Activa	18/12/2002		630000.00	362.49			<input type="button" value="X"/> <input type="button" value="CMA"/>
<input type="checkbox"/>	Madrid	Torrelodones	Emilia Alarcos	09	Adosado Esquina	Activa	31/12/2005		620000.00	269.98			<input type="button" value="X"/> <input type="button" value="CMA"/>

Número de Página: 1 de 20 Mostrar 10 filas

Administrar habitaciones

Picture 7: ABM unit of properties

Each time the user chooses an operation from ABM unit, the system displays a new web page that allows performing that operation, and after confirming it or not, the system leads again to the initial page of ABM unit.

These transitions among web pages do not correspond to a simple interface effect that organizes operations that the user can execute and avoids confusion, but rather they represent the moment in which the web server receives a new requirement from the PC client, giving back a new interface as a response.

The operation of each unit and the overall system is very much alike, and follows the “client’s requirement-server’s response” model, showing a new page each time the client wants to have access to a new operation.

Although this model operates for the overall system, the CMA execution displays special characteristics which require a different functionality.

As it was earlier explained, after selecting comparable characteristics for the production of CMA, a web page with details of the comparisons and their adjustments is displayed.

This adjustment assignment page shows three conditions:

- **It includes several ABM units.**
- **Do not lose track of the rest of the data.**
- **A CMA must be stored only when finished.**

3.1 ABM Units

Analyzing the different parts of a CMA, it is important to highlight the inclusion of:

- **1** Property in consideration
- **n** Comparable properties (marker)
- **n** Property in consideration – marker property comparisons
- **i** Adjustments per square meter assigned to each comparison
- **i** Adjustments per standard condition assigned to each comparison
- **k** Adjustments per non-standard condition assigned to each comparison

The information about the property in consideration and marker properties is revealed once stages 1 and 2 of the comparative analysis are finished.

In stage 3 the user has an overall view of all the comparisons and will be able to add/delete/modify the corresponding adjustments. Thus, for each comparison there are three ABM subunits, each of them for each type of adjustment.

3.2 Do not lose track of the rest of the data

Given that the analysis is *comparative*, it is not only necessary to have visual access to the comparison to which adjustments are being made, but also it is necessary to analyze the rest of the comparisons, that is to say, the possibility to compare the adjustment values among different marker properties.

That is the reason why it is important that during the CMA production, be it through adding, modifying or deleting an adjustment to any comparison, the user never loses track of the rest of the comparison as well as of the adjustments already made. Thus, it is not valid to use standard technology to gain access to a new page each time an adjustment operation is made.

3.3 A CMA must store only when finished

This stage involves a very important aspect of CMA unit since unlike a simple ABM, a CMA requires an elaboration process, so that data which are part of it, are not definitive until the user confirms that s/he has finished the CMA and wants to store it.

So far, CMA stage 3 has been analyzed, where the user has access to all the information to make the necessary adjustments.

At first sight, it seems that the implementation of this stage is not complicated since it is a set of ABM units such as those which form part of the system. However, the need to fulfill with all the mentioned conditions turns the implementation different from the standard ABM model.

4. Restraints in the CMA development using standard techniques

One of the major restraints that the CMA development has is that its storage must be done only when the user confirms that the selected marker properties and the adjustments are correct.

During the elaboration process the user faces the possibility of adding or deleting an adjustment. Therefore, it is necessary that the adjustment of each operation should be stored temporally until the CMA is confirmed or cancelled.

Implementation in PHP using a standard technique, where each operation implies sending the server a requirement and waiting that a new interface allows a new operation, consists of the following steps:

- The user must insert an adjustment, thus a requirement to the server is made.
- The server sends a new page as a response with an interface that allows adding data to the adjustments.
- The user completes a form and confirms the operation, sending a new requirement to the server again.
- The server inserts the adjustment to the database and responds with a new interface where updated data are displayed.

It can be noticed that using this technique the condition “Do not lose track of the rest of the data” is not being fulfilled. However, there is still a more complicated problem that arises if the user decides to close the browser to cancel the CMA production, or if s/he simply decides to open a new URL.

In this case, a server does not have any means to know if the user has decided to abort the CMA production. Consequently, data added to the database will remain when they should be deleted, given that a CMA has been aborted. Therefore, the third condition “A CMA must store only when finished” is not fulfilled either.

The need to have an overall view of the information each time a new operation is made, represents, basically an interface problem. As a new web page must be produced each time a requirement to the server is made, all the information of the previous page is lost.

A solution to this problem would be that the new WEB interface received should contain the previous information added to the necessary fields to perform the operation chosen by the user. The disadvantage of this alternative is that there is a lot of redundant information in the web, that is to say, the user is constantly receiving information already available at hand.

Assuming that the above mentioned solution is valid though its disadvantages, only one of the two last conditions of CMA are being fulfilled. It is still a question to solve the need to store CMA data definitively only if the customer confirms it.

Achieving the fact that the server knows the decision to abort a CMA production implies that it must receive a requirement that indicates that precise event.

The main reason why a server should “know” that the CMA production has been aborted is that the temporal information stored while processing should be eliminated.

The cases in which such event could occur can be summarized in the following way:

- The user clicks some button to cancel the CMA production.
- The user closes the browser.
- The user opens a new URL.
- The PC turns off or disconnects from the web.

The solution to the first problem is easy, since it involves performing the requirement to the user when the clicking event has been produced from a button of the web page.

The following two cases share an important characteristic: although they are different, both depend on the browser, not on an action produced over the WEB interface itself. The only means to detect any of these two events is by means of a scripting language “in the client” (in this particular case JavaScript). However, the notification to the server that this event has been produced is still a problem to solve.

A not very advisable option would be to create a new browser window using JavaScript, so that it shows the outcome of having performed a request to the “cancellation of CMA production” server, when capturing the event. Obviously, the fact of another window appearing when the user closes the browser, which has to be closed, is neither useful nor practical option.

A more practical alternative which covers the fourth item (the client system turns off or disconnects from the web) would be that CMA data are not stored in the database while the user performs each operation, on the contrary, this storage is performed locally in the client, and is only transferred to the server database when the user confirms the CMA.

Again, the use of JavaScript is important, since in this case information is being stored in the memory or in the disk. Its main advantage is that in case any of these events occur, junk information will not be stored in the database.

Although this solution is more useful, again, there are reasons to believe that it is not the most advisable one:

- If the equipment is turned off abruptly or does not work, it is impossible to recover the data in memory.
- Transferring all the CMA information from the client to the Server in the very moment that its production is confirmed requires performing in one request 150 adding consultations to the database, if an average of 10 comparisons with 15 adjustments each is considered. Transference cost is increased, and it should be considered that the amount of information that can be sent as a parameter in one requirement is limited.

Analyzing the particular characteristics of a CMA unit, different solutions have been proposed. However, none of them covers the problem entirely, with more than one issue to solve.

Summing up, it is important to highlight that the different alternatives have respected the use of standard web techniques in development (“requirement→response” pattern in which each response involves gaining access to a new page with required information).

Now, an alternative of WEB development will be shown to implement a complete solution to the CMA problem, using the same tools (PHP – MySQL –JavaScript) but with another work methodology.

5. Use of asynchronous JavaScript and XML

AJAX (Asynchronous JavaScript and XML) [14] [15] [16] is a WEB development technique which combines the following technology:

- **HTML (or XHTML) and CSS (Cascade Style Sheet):** These technologies are part of the standard used nowadays for the publication of hypertext in the WEB and its representation. A WEB page is made up of a HTML document specifying its content and a CSS specifying the way in which it will be represented.
- **DOM (Document Object Model):** Object oriented model which allows the access to each of the elements that are part of a HTML document by means of scripting languages.
- **XMLHttpRequest:** Object provided by languages such as JavaScript for data interchange with WEB server.
- **XML (Extensible Markup Language):** Extensible label language designed as a standard for information interchange between different platforms.

The essential characteristic of AJAX is that it provides the possibility of executing an application in the client keeping an underlying connection with the server asynchronously. (**Picture 8**).

This behavior gives a web development a similar functionality of that of desktop application, in which the user has a higher interactivity without establishing requirements to the server continually.

Basically, the use of this development technique consists of the following steps:

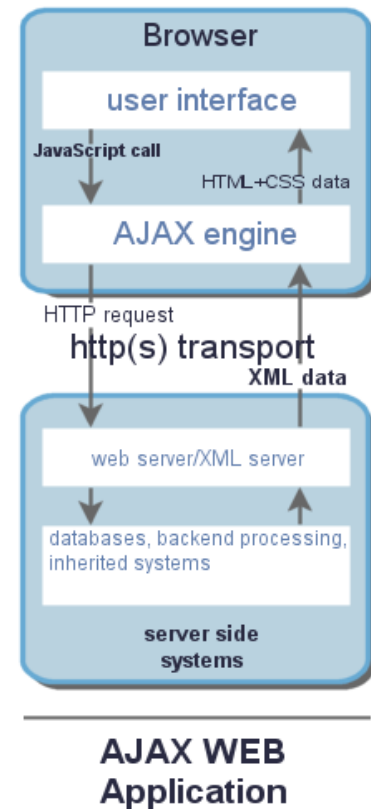
- Detecting an event made by a user by means of JavaScript.
- Sending an asynchronous requirement to the server resorting to the XMLHttpRequest object from JavaScript.
- In the moment a response is received (through a function call), modify the current HTML document content using DOM. The response information is generally found in XML format and the access is met by means of XMLHttpRequest.

It can be noticed, thus, that AJAX does not require “refreshing” or “reloading” a web page each time a request to the server is done.

This characteristic is, at first sight, an improvement as regards the interactivity level between a WEB application and the user. However, in the CMA case, it seems to be a new alternative to the early analyzed solutions.

As regards the above mentioned conditions:

- **Do not lose track of the rest of the data:** this technique allows generating a CMA without losing access to the information obtained during the elaboration process. It is neither necessary updating pages nor receiving redundant information, only what is updated.
- **A CMA must be stored only when finished:** The solution to this problem using AJAX consists of a combination among possible implementations early analyzed. So far, the main problem using conventional development techniques was to notify the server when an event was produced. Using AJAX, however, it is possible to execute a requirement to the server each time an event is produced, without the need for the browser to gain access to a new URL.



Picture 8

This new approach motivated the design of a new architecture as a solution to every drawback that early alternatives had:

- Each time the user makes an ABM operation over an adjustment, an asynchronous requirement must be performed to the server to reflect it in the BD.
- Once the updated information is received, a DOM must be used to modify the interface and display the changes without updating the entire page.
- In case the user decides canceling the CMA production, be it the cancellation through the interface, the access to another URL or closing of the browser, it is possible to detect any of these events by means of JavaScript. Once the event has been detected, it is only necessary to send a requirement to AJAX so that the server is in charge of eliminating the CMA under production.
- In case PC accidentally shutdown or any problem in the network appear, the temporary data are stored in the database with recovery possibility to continue the CMA.

Analyzing this implementation, it can be concluded that JavaScript manages to determine when any updating is necessary, thus replacing PHP server. However, the lack of need for the server to update the interface does not mean that the information is only in the client. All modifications are stored in the database each time the user makes a modification.

Although the main problem when storing the CMA production data occurred since it was necessary to notify the server before the cancellation of a process, it is possible, using AJAX, to notify the event so that the server deletes all the useless data.

Nowadays the Area6 project allows the complete CMA production manner using AJAX. The use of this development technique provided the possibility of fulfilling the application requirements, without resorting new technologies or programming languages.

AJAX avoids the sending of redundant information from the server to the client, transferring only the necessary information to update the interface. That is why the bandwidth used in these types of applications is lesser than a WEB application developed with the early mentioned techniques.

It must be noticed that the use of this development technique has allowed fulfilling the CMA implementation with its fundamental features, reducing information transfer costs.

Besides, it is also important the visual impact of an application developed under AJAX, since it is similar to desktop application, where it is possible to interact with the different parts of an interface without the need to access to a new web page to see the results.

However, one of the main disadvantages of using AJAX is the processing amount in the client. JavaScript is an interpreted language and its efficiency is determined by the hardware and software used. That is the reason why it is important to minimize the client's responsibility so that its function is to display information, other than processing it.

6. Conclusion

Analyzing the different solutions for the CMA production in Area6 project, it was clear that although it was possible to find a solution, a satisfactory application resorting standard WEB application was not fulfilled due to their limitations.

The use of AJAX as an alternative development technique provides a different approach over a WEB application as regards its interactivity. It reduces the need of a wider bandwidth and displays new options for the resolution of particular problems.

Nowadays, the use of AJAX is widespread due to WEB applications such as Gmail, able to provide new options to the user without the need to install plug-ins or other devices.

In the particular case of Area6 Real Estate Agents, it shows that it is possible to achieve new solutions changing implementation methodology only, instead of looking for different technologies or altering the type of application.

7. Future Works

Presently, CMA unit allows the production of a comparative market analysis for a particular real estate. The information produced by the CMA can be obtained in HTML or PDF format according to the user's need. It is also possible to consult already produced CMAs and, although not implemented, it is totally possible for the user to modify an existing CMA.

One of the main goals of future updating works is to implement the possibility of making comparisons among CMAs already produced so that it allows a wider analysis.

Finally, another characteristic to implement is the possibility to delay the production of a CMA so that it can be finished in another moment. It also can be possible if the user equipment stops working or is turned off abruptly.

8. References

- [1] Coulouris G., **Distributed Systems – Concepts and Design**, Addison-Wesley, 1994.
- [2] Dubois Paul, **Edición Especial MySQL**, Prentice Hall, 2000.
- [3] IEEE, **IEEE Recommended Practice for Software Requirements Specifications**, IEEE Std 830-1998 (Revision of IEEE Std 830-1993)
- [4] IEEE, **“An American National Standard. IEEE Guide to Software Requirements Specifications. ANSI/IEEE Std 830-1984”**, en Dorfman, M., Thayer, R.H., Standards, Guidelines, and Examples on Systems and Software Requirements Engineering, IEEE Computer Society Press, Los Alamitos, 1990
- [5] Holzner Steven, **PHP 5 Manual Avanzado**, Anaya Multimedia, 2005.
- [6] Loucopoulos P. Karakostas V., **System Requirements Engineering**, McGraw-Hill International series in Software Engineering, 1995.
- [7] McFedries Paul, **Edición Especial JavaScript**, Pearson Educación, 2002.
- [8] Rodriguez de la Puente Santiago, **Programación de aplicaciones WEB**, Paraninfo, 2003.
- [9] Sharp H., Finkelstein A., Galal G., **Stakeholder Identification in the Requirements Engineering Process**, Proceedings of the 10th International Workshop on Database & Expert Systems Applications , September 01 - 03, 1999 , Florence, Italy.
- [10] Sklar David, **Learning PHP 5**, O'Reilly, 2004.
- [11] Torchiano M. Morisio M., **Overlooked Aspects of COTS-Based Development**, IEEE Software, 2004.
- [12] Zawodny Jeremy D., **MySql Avanzado**, Anaya Multimedia, 2004.

- [13] Zikmund William G., **CRM Customer Relationship Management**, C.E.C.S.A., 2004.
- [14] <http://www.ajax.org/>
- [15] <http://es.wikipedia.org/wiki/AJAX>
- [16] <http://developer.mozilla.org/en/docs/AJAX>
- [17] <http://www.ecma-international.org/>
- [18] <http://msdn.microsoft.com/scripting/default.htm>
- [19] <http://homebuying.about.com/library/glossary/bldef4.htm>
- [20] <http://www.skype.com/>
- [21] <http://www.realestate.fnf.com/Products/AgentOffice.aspx>
- [22] <http://www.iso.org>