AJAX - Asynchronous Java Script and XML

Sudhir Batra

University of Applied Science and Technology - Salzburg
ITS - Information Technology and Systems Management
Supervisor: Dr. Dipl.-Ing. Thomas Heistracher
sbatra.tks2003@fh-salzburg.ac.at

Abstract. AJAX - Asynchronous Java Script and XML - is a new implementation of established web development technologies to gain interactivity between users and servers through multiple client sided server connections in the background. It allows to integrate web applications in browsers, as a consequence the borders to desktop applications are disappearing. AJAX is a kind of framework that leads Web programming into the direction of Web 2.0, the new generation of using and developing the internet. This paper discusses AJAX, Web 2.0 and gives an idea about the principles of the asynchronous Java Script and XML technology by implementing examples and evaluating the concepts.

Table of Contents

1 Introduction

The main topic in this paper is the evaluation and discussion of AJAX - asynchronous Java Script and XML. Technical aspects are described and an overview about the main concepts is given. This paper also discusses Web 2.0 and the history of Web Services, shows examples and gives an idea about the principles of the asynchronous Java Script and XML technology by implementing examples and evaluating the concepts. The first chapter will give a short common introduction to Web programming. After this introduction details about AJAX are described in chapter 2. Further details and more technical aspects are evaluated in chapter 3. Chapter 4 deals with a practical part, in which also source code examples are implemented. In chapter 5 a review is given and an outlook for the future is discussed.

1.1 History of Web Services

Similar to almost every IT - sector also in the part of Web services and programming the development und innovation of technologies in the last few decades was risen in a very significal way. As in 1989 Tim Berners Lee invented the Hypertext Markup Language, nobody knew what kind of rapid development it would lead to. In the first years HTML was only used for static websites and for layout purposes. But HTML is still, like nearly twenty years ago, also nowadays (today the XHTML 2.0 standard is common) hierarchically structured and assembled by so called tags. This is a very important aspect for the DOM - Document Object Model [15], which will be evaluated later in detail.

The more sites web designers and web programmers implemented, the more the demand for dynamic web sites increased. In 1998 first implementations of Dynamic HTML were published technically feasible with Java Script. These were the first foreriders of the new AJAX framework, which generally only uses existing technologies. But not everyone was affected with that hype, one of the problems was the Netscape Microsoft browser war. While Netscape invented the JavaScript object based language, Microsoft countered with its Jscript which had similar functionalities but for web programmers there were too many problems with compatibility. Also nowadays it is not easy to create Java Script applications compatible for every browser. While Microsoft uses the Active X support in its Internet Explorer, the Gecko browsers (Mozilla, Firefox, etc.) are not completely compatible to them.

Later websites gained interactivity and dynamic actions through Java applets and Flash applications which all use the common browser-server request: a user opens a browser, sends a request to a server, the server handles the request and gives it back to the browser where the user waits for the answer. With the help of AJAX a lot of these connections can be realized simultaneously while the user is working.

In 2000 the establishment of XML allowed the describing of data. XML, which is a metalanguage, forms the basis of many Web services and allows to exchange data in a standardized way. It also works with the use of tags, which can, in contrast to HTML, be self invented. For the last few years more end devices (mobiles, PDAs, etc.) have created a new challenge. Through these developments and evolutions the next step - building interactive Web applications was not very far away. Such interactive applications are the basis of the new generation of the Web - Web 2.0.

1.2 Web 2.0

The latest generation of the world wide Web is the so called Web 2.0. Through the development and the success of Web Services, information and several providers like Wikipedia.org or Google the kind of information exchange in the Web is changing and evolving. Interactive Web applications in which users can be important interactors or can play parts within lead to eliminate the border to desktop applications. An example for these developments is the online photo shop and information service 'flickr'. Users can upload their own photos and give them an XML-tag keyword they like. So others have the possibility to search for photos with these keywords. Table 1 shows the evolution of the Web. Another aspect of the new Web 2.0 generation is the trend away from personal Websites to blogging information. A reason for the hype of Web 2.0 are that the broadband has become mainstream and ubiquitous, resulting in an increased usage of the Internet for even small tasks on different devices and so more people go online for a variety of tasks and shopping-related activities. The trends go to [14]

- A social phenomenon referring to an approach to creating and distributing Web content itself, characterized by open communication, decentralization of authority, freedom to share and re-use, and "the market as a conversation"
- The transition of Websites from isolated information silos to sources of content and functionality, thus becoming a computing platform serving Web applications to end users
- A more organized and categorized content, with a far more developed deep linking Web architecture
- Web 2.0 is a marketing term to differentiate new Web businesses from those of the dot com boom, which due to the bust now seem discredited
- The resurgence of excitement around the possibilities of innovative Web applications and services that gained a lot of momentum around mid 2005.

[14,9]

Web 1.0	Web 2.0
Britannica Online	Wikipedia
personal Websites	blogging
domain name speculation	search engine optimization
content management systems	wikis
publishing	participation
directories (taxonomy)	tagging
screen scraping	Webservices
stickiness	syndication

Table 1: Comparision Web 1.0 - Web 2.0

2 AJAX

This chapter describes the main concepts of AJAX - asynchronous Java Script and XML - framework. The benefits and disadvantages are discussed and some examples are given.

2.1 Concepts of Ajax

AJAX - asynchronous JavaScript and XML - is no programming or script language, no new invention and no seperate Web service, module or plug-in. In common it is a marketing term for 'Remote Scripting with JavaScript, CSS and DOM'. It is an algorithm with 'old' technologies similar to the Dynamic HTML. Ajax allows to create server connections in the background while a user is interacting with a Web front-end. These connections can be created asynchronously, which means that the user need not wait until the server replies. They are usually created as a consequence of events, realized in JavaScript which offers easy event handling. XML is used to exchange data between server and client (browser). For the user no complete reloading of the Website is necessary. E.g. when a user types an email address into an input form, it is possible via AJAX to create a server connection in the background, check if the address is valid or not and give the information back to the user via an output [8].

2.2 Benefits

There are a lot of advantages of the AJAX technology. No pushing on a submit button and reloading of a complete Website are needed. So the interactivity and the speed for the users are more efficient. A service can be adopted on a persons need and gain more information if the users decides for a certain step. AJAX is not a general solution for all Web development problems, but it can be used in a very needful und actual way creating a user friendly application. On the developer's side it is possible to create database connections or script executions during interacting with users.

On the other hand there are also some disadvantages. As AJAX is a very new combination of old technologies, no one can be sure if it is only a marketing hype now or if it will really be established in some years. So there are no best practices. There are a lot of applications which use it, but perhaps a better technology can challenge it. One big problem is the compatibility. There are some problems with Microsoft Internet Explorer, which can be avoided with a little bit of programming but some IT-technicians beware of that. Another point is that JavaScript can be switched off in browsers, because of security reasons. Without the support of JavaScript no event handling and server connections on the client side are possible. The next disadvantage as regards surfing comfort is that no back button in AJAX applications is available (as in Gmail). Because of the asynchronous generated code the browser has no former page in the cache and cannot reload it exactly. To establish AJAX connections a little bit client-server Web knowledge like parameter passing is needed (get,post,put,etc.). Some of these problems can be solved with AJAX frameworks, which help developers to create AJAX requests and services more easily [1]. These frameworks are discussed in chapter 4.1.

2.3 Server/Browser Model

As already mentioned prior AJAX allows multiple server browser connections in the background. It's possible to create two kinds of connections: synchronous and asynchronous. For interactivity and actuality it is necessary to use the second, so the browser does not wait for a reply of the server script. So the Website need not to be completely reloaded, the focus is on the - for the user - informative part. Fig.1 and Fig.2 show the differences between the usual server-browser and the AJAX server-browser model where a AJAX-engine is needed to send and receive requests asynchronously.

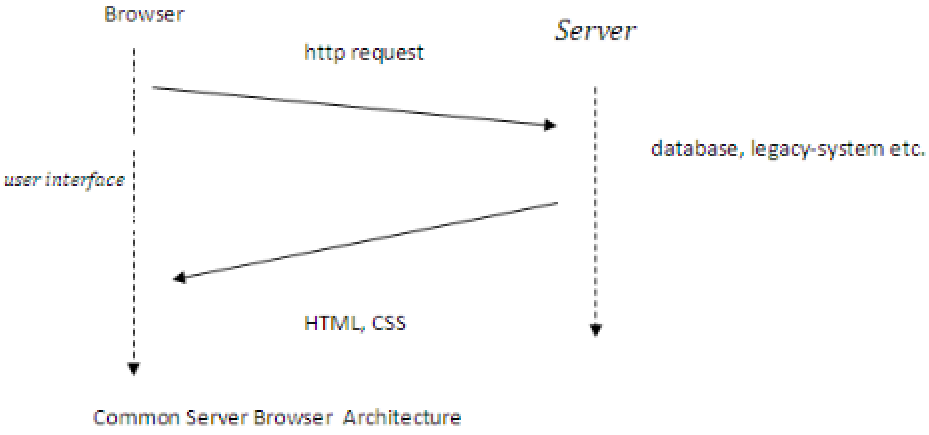
Browser
Server
http request
database, legacy-system etc.
user interface
HTML, CSS
Common Server Browser Architecture

Fig. 1. Common Server Browser Request

2.4 Existing Ajax Tools

There are already several existing AJAX applications. One of the biggest supporters of this technology is Google. In the following part of this paper, some interesting interactive and AJAX-based applications are described and evaluated.

Google Suggest [5]- is an extension of the common Google Search engine in Beta state. As soon as letters are typed into the form, a pull down menu with search results and combined keywords pops up. The user is immediately able to see possible results and can choose his desired combination or even combinations he has not known before.

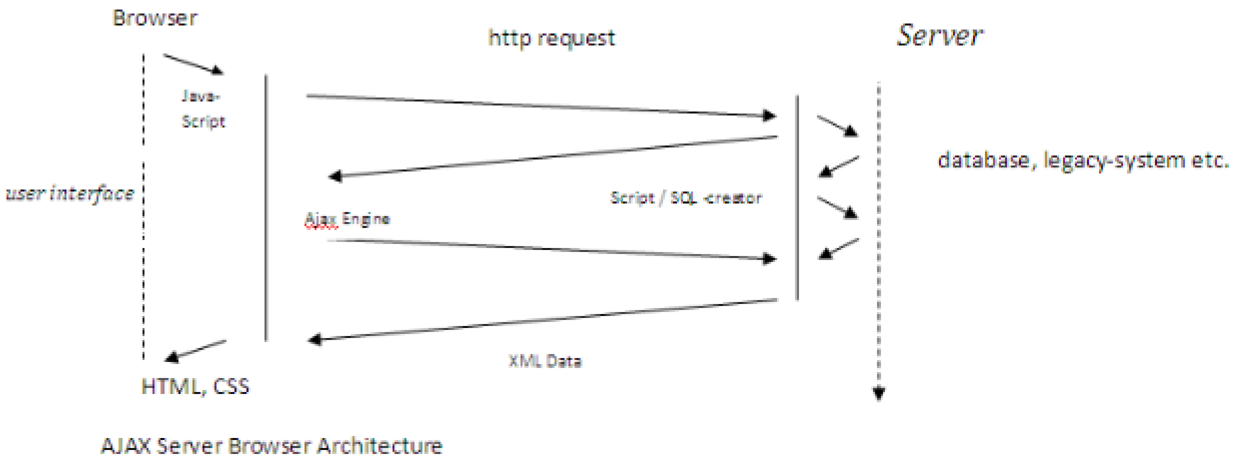

Fig. 2. Ajax Server Browser Request

Google Maps and Google Earth [6] - is a world wide map built up with satellite pictures. Users have even the possibility to see three dimensional views of landscapes and famous cities. There is also additional information for tourisms places and geographical values. Not the complete information is loaded when the program is started, through AJAX only the needed and wanted information - chosen by a mouse click by the user - is reloaded and zoomed. It is even possible to create route maps with Google Earth.
Writely [13]is an online browser-dependent word processing software which is by some people seen as an opponent to Microsoft Word. It is not necessary to install or buy an expensive Office distribution, you only need a standard browser like the Microsoft Internet Explorer, type in the URL and you are ready to use the program. The complete style is very similar to Microsoft Word. AJAX handles the written characters and saves them in the background.
Gmail- [4] The Google Mail System is also based on AJAX Services. The complete menu to administrate your received and sent emails is not reloaded if you delete or disarrange a letter. A request to the database in the background is sent.

3 Technical Aspects

This chapter describes the functions and concepts of AJAX in more detail and in technical aspects. It describes the basics, the data exchange and manipulation and also the creation of requests.

3.1 Technical Overview

AJAX is implemented with the client sided JavaScript programming language. JavaScript provides easy event handling and is almost integrated in HTML. It can be used as an object based language and allows easy manipulation of data with the Document Object Model (DOM) and Cascading Style Sheets (CSS). These two technologies support the use of the object based feature of JS. The syntax is similar to Java but the use is completely different. Because of compatibility reasons and the possibility in browsers to switch it off, JavaScript lacks in popularity in the IT - environment. But also in spite of its bad failure recognition it is available on every browser and offers very simple and various techniques which are necessary to know for all Web programmers and designers [3].

3.2 XmlhttpRequest

The XmlHTTPRequest is the heart of all AJAX applications. It is a JavaScript object which can usually be simply instantiated.

```
function createXMLHttpRequest(){
var req = null;
try { req = new ActiveXObject(MSXML2.XMLHTTP);}
catch (err_MSXML2){ req = new ActiveXObject(Microsoft.XMLHTTP) ;}
catch (err_Microsoft){
if (typeof XMLHTTPRequest != undefined)
req = new XMLHTTPRequest;
}
}
return req;
}
req.onreadystatechange = handleStateChange;
req.open(GET,http://w3.org/,true);
req.send(anything);
```

After instantiating an XmlHttpRequest the response of the server can easily be derived with help of the handlestateChange function and the readystate variable.

```
function handleStateChange(){
switch (req.readystate){
case 0: //UNINITIALISED
case 1: //LOADING
case 2: //LOADED
case 3: //INTERACTIVE
break;
case 4: //COMPLETED
handleResponse(req.status , req.responseText);
break;
default : ;// failure state}};
```

With its open method the XmlhttpRequest object offers connections with http-requests (get,post,put,etc.) and the possibility to choose between synchronous (false-parameter) and asynchronous (true) connections. Some frameworks and libraries offer the encapsulation of the XmlHTTPRequest which ease the use and handling for programmers. Asynchrony allows user to work during code generation.

3.3 Data transformation and Exchange

Servers and browsers have to communicate after an opened request. This can be done with the readystate variable, which gives information about the current connection status. But if more information is needed - which is the common case - like database information or other data, the general data transformation and exchange format is XML. Logically seen you do not need to use it, e.g. if you only wait for a one word reply or a parameter (a inserted primary key value which was generated automatically etc.) but with the use of a big amount of data it makes sense. With the help of XML it is possible to structure and describe data logically and script languages also support or even integrate XML - parsers which transforms XML in usable data.

3.4 Representation and Manipulation of Data

The representation and manipulation of data are managed in a way of handling objects. The Document Object Model (DOM) of Javascript allows to create objects of a HTML document. These objects are instantiated hierarchically structured like the tags in the code. That is why it important to create a well formed HTML document. If not DOM cannot recognize the tag soup. The Document Object Models then creates child nodes and parent nodes which can easily be altered or appended. This is done by supported functions in JavaScript. Example:

```
<div class="testclass" id="testid" onclick="change()";>Hello World!</div>
<script language="javascript">
function change{
var testclass = document.getElementbyId("testid");
testclass.innerHTML="New World";
}
</script>
// other functions:
// getelementbytagname(), getelementbyid, haschildnode(), appendchildnode();
```

DOM can be used with Cascading Style Sheets (CSS) to represent data. That allows splitting code and adding events (onmouseover(),onmouseclick(),etc.) to several classes. The Document Object Model is supported by all new browsers where JavaScript is turned on [9,15].

4 Practical Part

This chapter tries to represent the application levels of AJAX and gives some examples of how to use it. For that reason some frameworks and also server sided AJAX solutions are evaluated. For the TEC project a drag and drop front end with an AJAX implementation in the background was developed.

4.1 Frameworks for AJAX

Frameworks ease the work and the programming of Web applications. They are a very mighty tool and can be very useful when using AJAX technology. The following sections deals with some important AJAX frameworks. **SAJAX** [12] is a framework for server sided implementation of the AJAX algorithm and a possibility to execute server sided functions through the use of browser sided Javascript. It is an interface to integrate client code on servers and offers modules for PHP, ASP, Perl, Python and also Coldfusion. There are 5 steps which have to be implemented [8]:

1. integrate library – *require(sajax.php);*
2. define functions on server script
3. initialise – *sajax-init();*
4. export functions to make them available to clients – *sajax-export(functionname);*
5. handle client requests – *functions are available for clients with x-functionname*

Sarissa [10] is a huge JavaScript library which helps programmers and developers to ease the use of AJAX. Although there is no help for connections there are needful functions which extend DOM and XMLHttpRequest. Sarissa is often used with the Prototype library which is very useful in implementing design features and visual effects.
ATLAS [7] is a Microsoft supported AJAX solution which is compatible to all .NET applications and frameworks. It is usually used with Microsoft's ASP.NET script language.
wiki.script.aculo.us [11] is an online JavaScript library implemented by two Austrians and which provides a lot of useful information and encapsulated AJAX functions. It provides also a lot of documentation and helpful demos. The basics of these functions where also used in the TIGS [2] project.

4.2 Sourcecode Example

The TIGS [2] project is an online syndication portal similar to a content management system for tourism providers (attractions and institutions) and tourism disposer (hotels or tourism institutions) implemented by the Forschung Urstein together with the Fachhochschule Salzburg. For the selection of the provider information the tourism disposers have to use an interactive drag and drop front end with AJAX technology in the background. It is implemented with a server sided PHP-PEAR engine and the template system SMARTY where the HTML and JavaScript code is generated using the model-view-controller architecture. The important parts of the source code can be found in the appendix.

4.3 Alternatives

Remote scripting can also be implemented with other technologies. For example a simple Inline Frame could also be able to reload server - sided dynamic generated Websites. But the disadvantage of iframes are security and compatibility reasons. Iframes are neither dynamically producible. Other alternatives are Live-Connect functionality of Flash applications or JAVA implementations, but these technologies are manufacturer specific and it's difficult to implement extensions compared to AJAX. Another very new alternative is the DOM 3.0 load save specification which also allows several browser - server connections in the background. But the disadvantage is that it's not yet compatible in all browsers, only Opera supports it. AJAX combines the advantages of these alternatives without having their disadvantages.

5 Review and Summary

Web developers and programmers have to decide if AJAX will really be a common used standard in the next decade. Critics say there is nothing new with AJAX, it's only a marketing term for old dynamic HTML combined with JavaScript. Through the common use of broadband internet connections the more server connections in the background won't be that slow, other criticism is security and compatibility aspects because of Microsoft's Active X standards. It is not possible to use AJAX in every problem or with every Website but it is a useful way to gain interactivity and it is easy to learn and extend. Not everything can be done with AJAX but it is a new challenge and a step further for disappearing the border between desktop applications and Web applications and services. With the help of frameworks AJAX is a very powerful technique, no matter if it's only a term for well established technologies.

References

1. Christian Wenz. Ajax. Entwicklerpress (2006), 2006.
2. Forschung Urstein. Tourimus Info Gate (TEC/TIGS). http://www.tourimus-info-gate.at, 2006.
3. Gerhild Maier. Ajax von A bis X. http://krottmaier.cgv.tugraz.at/docs/seminar/sem2005_ajax.pdf, year = 2006.
4. Google. Google Gmail. http://gmail.google.com, year = 2005.
5. Google. Google Suggest. http://www.google.com/webhp?hl=de, year = 2006.
6. Google. Google Maps. http://maps.google.com, 2006.
7. Microsoft. ATLAS. http://www.microsoft.com/germany/msdn/library/web/AJAXUndASPNET.mspx?mfr=true, year = 2006.
8. Olaf Bergmann, Carsten Bormann. Ajax - Web 2.0. SPC Teija Verlag (16.11.2005), 2005.
9. Paul Miller. Web 2.0: Building the New Library. http://www.ariadne.ac.uk/issue45/miller/, year = 2006.
10. Sarissa. Javascript Library. http://swik.net/Sarissa/, year = 2006.
11. Scriptacolous. script.aculo.us.
12. Simple Ajax Toolkit. SAJAX. http://www.modernmethod.com/sajax/, year = 2006.
13. The Web Word Processor. Writely. http://www2.writely.com/info/WritelyOverflowWelcome.htm, year = 2006.
14. Tim O'Reilly. What is Web 2.0 ? http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (23.05.2006), 2006.
15. W3C Community. DOM 3.0. http://www.w3.org/TR/2004/NOTE-DOM-Level-3-XPath-20040226/, year = 2006.

A Appendix

```
File editabo.php (controller):
<? include "intro.php";
   check_auth();
$db->setFetchMode(DB_FETCHMODE_OBJECT, "paket");
  $sql = "SELECT distinct * FROM paket
  LEFT JOIN user USING ( USERID) ORDER BY KURZTITEL";
    $result =& $db->query($sql);
    $pakete = array();
while($result->fetchInto($paket)){
$pakete[] = $paket;
}
$smarty->assign('pakete', $pakete);
$smarty->assign('titel', "Neues Abo erstellen");
$smarty->assign('view',  "editabo"  );
$smarty->display('index.tpl');
?>

File editabo.tpl (view):

<script src="scriptaculous/lib/prototype.js" type="text/javascript"></script>
<script src="scriptaculous/src/scriptaculous.js" type="text/javascript"></script>
<script src="scriptaculous/src/dragdrop.js" type="text/javascript"></script>

<div align="center"><strong>Warenkorb:</strong></div><br />
<div id="korb" class="korb" >
<p align="center">Ziehen Sie hier mittels Drag and Drop Ihre gew&uuml;nschten
 Pakete hinein</p></div>
<div id="anzahl" align="center">0</div> <div align="center">Pakete gesamt</div>
<div id="loeschen" class="loeschen">
<p align="center">Ziehen Sie Pakete aus dem Warenkorb hier rein, um sie zu
l&ouml;schen<p></div>

<div id="alle-pakete">
{foreach from=$pakete item=paket}
<div id="{$paket->PAKETID}" class="paket">
<span class="zeile">{$paket->NAME}</span>
<span class="zeile"><b>{$paket->KURZTITEL}</b></span>

<script type="text/javascript" language="javascript">
 new Draggable('{$paket->PAKETID}' ,{revert:true});
    </script>
  {/foreach}</script>
</div>
```

```
<script type="text/javascript" language="javascript">
      Droppables.add('loeschen', {
    accept: 'waren',
     onDrop: function(element) {
      //Layout change

  var handlerFunc = function(t) {
      aboid.innerHTML = t.responseText;
}

  var errFunc = function(t) {
     alert('Error ' + t.status + ' -- ' + t.statusText);
}

new Ajax.Request('pakettoabo.php',
    {parameters:'delete=true&paketid='+element.id+'&aboid='+abo_id, onSuccess:handlerFunc, 
 }})

Droppables.add('korb', {
   accept: 'paket',
  onDrop: function(element) {

// Layout change

var errFunc = function(t) {
     alert('Error ' + t.status + ' -- ' + t.statusText);
}

new Ajax.Request('pakettoabo.php',
{parameters:'paketid='+element.id+'&aboid='+abo_id,
  onSuccess:handlerFunc, onFailure:errFunc});
    }})</script>
```