

A Study on Ajax in Web Applications with Latest Trends

S. Rajesh Kumar, R. Aravazhi

Department Of Computer Application
Bharath College of Science and Management, Thanjavur
sivanesanrajesh@gmail.com, aravazhi_r@rediffmail.com

Abstract: The Web was an interesting but not particularly exciting tool used by scientific researches-until 1993. In olden days the web technology used for creating web pages is HTML (Hyper Text Markup Language). Now a days the most efficient and advanced Web technologies using are XML (extensive Markup Language), DOTNET and ASP.NET. A Web browser is a software application designed to find hypertext documents on the Web and then open the documents on the user's computer. The Web browser displays a Web page as specified by the page's underlying HTML.

Keywords: Evolutions of Web, Ajax, XML, Ajax Updates, Conclusion

1. EVOLUTIONS OF WEB

Generally, each individual web page is delivered to the client as a static document, but the sequence of pages can provide an interactive experience, as user input is returned through web form elements embedded in the page markup. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application.

- In 1995, Netscape introduced client-side scripting called JavaScript, allowed programmers to add some dynamic elements to the user interface ran on the client side.
- In 1996, Macromedia introduced Flash, a vector animation player that could be added to browsers as a plug-in to embed animations on the web pages.
- In 1999, the "web application" concept was introduced in the Java language in the

Servlet Specification version 2.2. [2.1?]. At that time both JavaScript and XML had already been developed, but AJAX had not still been coined and the XMLHttpRequest object had only been recently introduced on Internet Explorer 5 as an Active X object.

- In 2004, Web 2.0 was coined, and it was proposed to use the Web as platform, as opposed to simply developing applications used Internet to communicate with the server.
- In 2005, AJAX was coined, and applications like Gmail started to make their client sides more and more interactive

2. INTRODUCTION TO AJAX

Ajax, or **AJAX** (Asynchronous JavaScript and XML), is a group of interrelated web development techniques used to create interactive web applications or rich Internet applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and

563

behavior of the existing page. The use of Ajax has led to an increase in interactive animation on web pages. Data is retrieved using the XMLHttpRequest object or through the use of Remote Scripting in browsers that do not support it. Despite the name, the use of JavaScript and XML is not actually required, nor do the requests need to be asynchronous.

3. HISTORY OF AJAX

While the term Ajax was coined in 2005, alternative techniques for the asynchronous loading of content date back to the mid 1990s. Java applets were introduced in the first version of the Java language in 1995. These allow compiled client-side code to load data asynchronously from the web server after a web page is loaded. In 1996, Internet Explorer introduced the I Frame element to HTML, which also enables this to be achieved. In 1999, Microsoft created the XMLHttpRequest object as an ActiveX control in Internet Explorer 5. This is now supported by Mozilla and Safari as native versions of the object. On April 5, 2006 the World Wide Web Consortium (W3C) released the first draft specification for the object in an attempt to create an official web standard. Ajax is Asynchronous JavaScript and XML

4. XML

XML is a language defined by the World Wide Web Consortium the body that sets the standards for the Web. XML supersedes other markup language such as Hypertext Markup Language (HTML). In HTML all the HTML elements you can use are predefined and there are not enough of them. In fact, XML is a meta markup language because it lets to create our own markup language.

FEATURES OF XML

- Easy data exchange
- Customize markup language
- Self-describing data

TECHNOLOGY AND DEVELOPMENT

The term Ajax has come to represent a broad group of web technologies that can be used to implement a web application that communicates with a server in the background, without interfering with the current state of the page. In the article that coined the term Ajax, Jesse James Garrett explained that it refers specifically to these technologies:

- XHTML and CSS for **presentation**
- The Document Object Model for dynamic display of and interaction with data
- XML and XSLT for the interchange, manipulation and display of data, respectively
- The XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

Since then, however, there have been a number of developments in the technologies used in an Ajax application, and the definition of the term Ajax. In particular, it has been noted that:

- JavaScript is not the only client-side scripting language that can be used for implementing an Ajax application. Other languages such as VBScript and EGL Programming Language are also capable of the required functionality.
- XML is not required for data interchange and therefore XSLT is not required for the manipulation of data. JavaScript Object Notation (JSON) is often used as an alternative format for data interchange, although other

formats such as preformatted HTML or plain text can also be used.

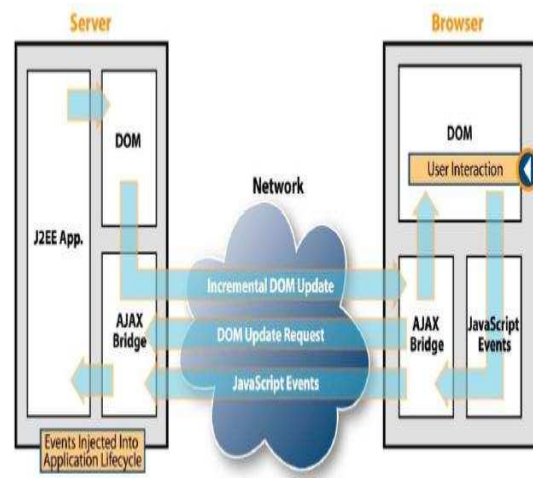
5. APPLICATIONS

- ❖ The AJAX Bridge overcomes most of the general concerns that we voiced earlier with regard to ad hoc AJAX. Specifically, the AJAX bridge:
 - ❖ Is small and well contained, deals with browser idiosyncrasies and eliminates JavaScript memory leaks.
 - ❖ Eliminates data and business logic replication at the browser, thus eliminating security issues within the application. We have now addressed most of the issues related to improving the presentation element of the Application, but we need to address user interaction with that presentation. To achieve this we hook the JavaScript event model back into the application event model via the same AJAX Bridge. The AJAX bridge
 - ❖ Is small and well contained, deals with browser idiosyncrasies and eliminates JavaScript memory leaks.
 - ❖ Eliminates data and business logic replication at the browser, thus eliminating security issues within the application.

BASIC AJAX ARCHITECTURE

We have now addressed most of the issues related to improving the presentation element of the Application, but we need to address user interaction with that presentation. To achieve this we hook the JavaScript event model back into the application event model via the same AJAX Bridge. This means that our application can react to user interaction with the presentation in real time and effect changes in the back end application data model.

Ultimately, these state changes are reflected in the application's presentation, and the incremental presentation changes can be propagated back to the client for presentation to the user. We now have an architecture that facilitates rich web applications and leverages server-side, industrial-strength, standards-based J2EE technologies to achieve it. This basic architecture is



THE DOCUMENT OBJECT MODEL (DOM)

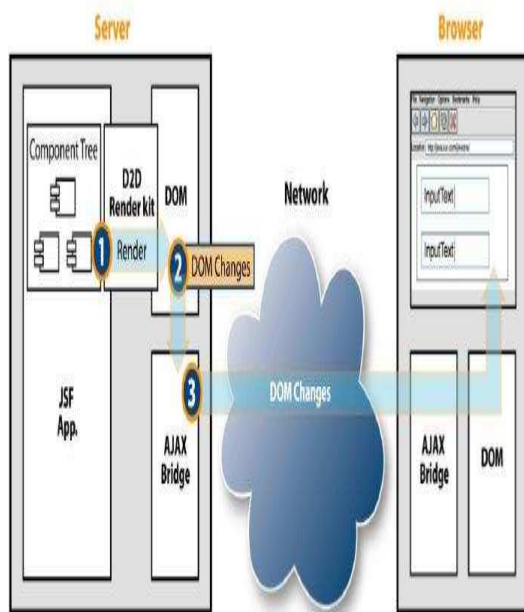
The DOM is a representation of the Web page as a hierarchy or tree structure, where every part of the page (the graphics, the text boxes, the buttons, and the text itself) is modeled by the browser.

JAVA SERVER FACES (JSF) WITH DIRECT-TO-DOM RENDERING AND INCREMENTAL UPDATE

One of the profound concepts in JSF is the Render Kit architecture. It provides separation between JSF component behavior, and the markup that represents those components in the presentation. The Render Kit architecture facilitates plugging different Render Kits into the same application in support of different presentation environments; so you might render HTML to a desktop browser and WML to a mobile phone, but maintains

common server-side application logic. We leverage this Render Kit architecture in the implementation of the JSF/AJAX architecture illustrated above, and introduce a Revolutionary technology called Direct-to-DOM rendering to achieve it. Direct-to-DOM rendering is just what it sounds like, the ability to render a JSF component tree directly into a W3C DOM data structure. During a standard JSF render pass, the component tree is traversed and each component rendered produces its output. Direct-to-DOM renderers produce their output into the server-side DOM. The DOM mutations that result are packaged up and delivered to the browser via AJAX bridge and reassembled to create the presentation for the application. The basic process is

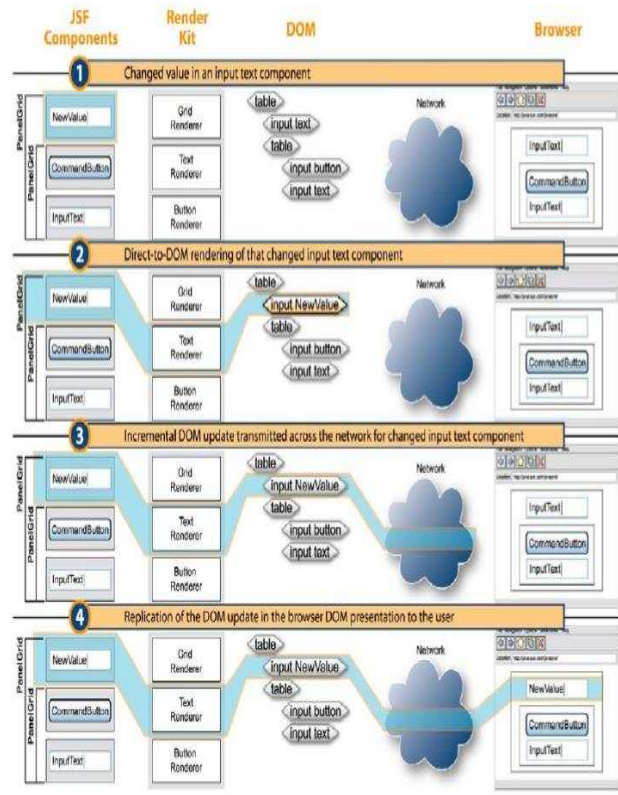
DIRECT-TO-DOM RENDERING



So, if we equip the JSF framework with a Direct-To-DOM Render Kit, we have a mechanism for efficiently generating and maintaining a server-side DOM, and if we

render only components that have changed, we minimize the number of DOM mutations that must be replicated at the client-side DOM. The following sequence of diagrams illustrates an incremental presentation layer update using Direct-to-DOM rendering.

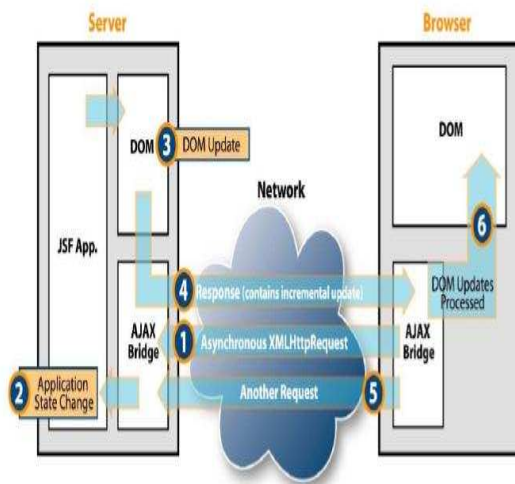
DIRECT-TO-DOM INCREMENTAL UPDATE



The combination of Direct-to-DOM rendering and incremental update completely change how we think about web application page design and opens the door to a myriad of rich, interactive features that are not shackled to the standard page refresh model. Now we not only want smooth incremental updates for presentation changes, but we want to be able to drive those changes to the client in an asynchronous fashion when the application state changes. To achieve this, we prime an ongoing update request loop from the client-side AJAX Bridge with an

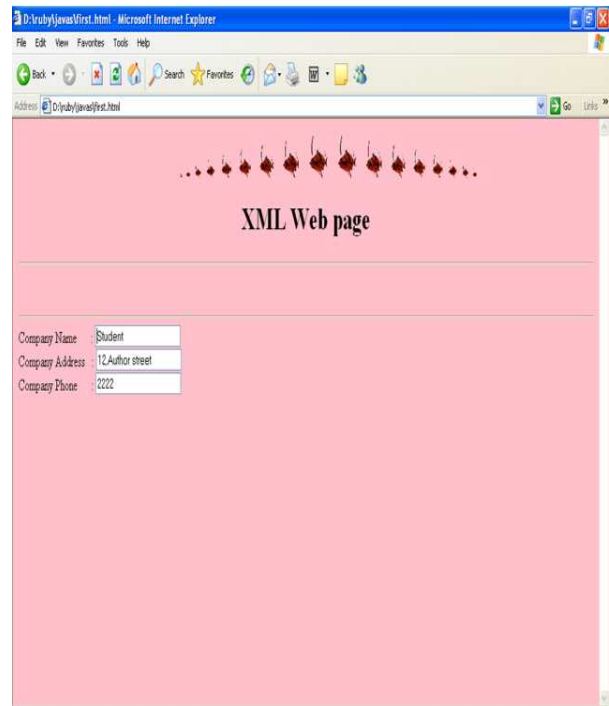
asynchronous XMLHttpRequest for incremental presentation changes. This request gets fulfilled at the server when the next set of incremental DOM updates is prepared. The response is transmitted back to the browser and a new request is issued prior to processing the DOM updates, thus facilitating ongoing asynchronous presentation updates, as

AJAX PRESENTATION UPDATES



Within the backing application, logic trigger points can be identified that cause presentation updates and these trigger points can force a JSF render pass to generate the required changes to the presentation. For example, a simple clock bean could produce and consume events once per second and use those events to trigger an update of the clock presentation on a page. With Direct-to-DOM rendering the clock will tick smoothly even as the user interacts with the page.

6. IMPLEMENTATION



SIMPLE XMLDOM CODE :

```

<!--
    xmlhttp object - readyState property

    uninitialized  0
    loading        1
    loaded         2
    interactive    3
    complete       4
-->
<html>
<head>
<script language=javascript>
<!--
var root1;
var root2;

var xmlDoc1=new ActiveXObject("microsoft.xmlhttp");
xmlDoc1.load("fl1.xml");
    
```

XML CODE :

```
<?xml version="1.0"?>
<message>
<cname>Student</cname>
<caddress>12,Author
street</caddress>
<phone> 2343 </phone>
<fax> 2222 </fax>
</message>
```

7. ADVANTAGES

- In many cases, related pages on a website consist of much content that is common between them. Using traditional methods, that content would have to be reloaded on every request. However, using Ajax, a web application can request only the content that needs to be updated, thus drastically reducing bandwidth usage and load time.
- The use of asynchronous requests allows the client's Web browser UI to be more interactive and to respond quickly to inputs, and sections of pages can also be Browser applications typically require little or no disk space on the client, upgrade and searching. They also provide cross-platform compatibility (i.e., Windows, Mac, Linux automatically with new features, integrate easily into other web procedures, such as email reloaded individually.

- Users may perceive the application to be faster or more responsive, even if the application has not changed on the server side.
- The use of Ajax can reduce connections to the server, since scripts and only have to be requested once.

8. CONCLUSION

"Asynchronous JavaScript and XML", is a web development technique for creating interactive web applications. The intent is to make web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire web page does not have to be reloaded each time the user requests a change. This is intended to increase the web page's interactivity, speed, and usability.

REFERENCE

1. www.w3schools.com/ajax
2. www.tutorialspoint.com/ajax/
3. www.tizag.com/ajaxTutorial/
4. www.courses.coreservlets.com/Course-Materials/ajax.html
5. www.oracle.com/technetwork/java/javaeetutorial-jsp-140089.html
6. www.xml-sitemaps.com/
7. www.webopedia.com/TERM/X/XML.html