More on AJAX Internals and Their Related Issues

Introduction

We have known that AJAX consists of the following technologies/components [1,2,3,4]:

1. Stylesheet language: is used to style the appearance of the page. CSS, XSLT, or combination of both can be used by AJAX application.
2. Javascript, a scripting language. One element of Javascript that is key to AJAX is XMLHttpRequest (XHR)
3. DOM (Document Object Model). DOM represents the structure of a web page as a programmable tree structure. Scripting the DOM allows an AJAX application to modify the UI on the fly, effectively redrawing parts of a page.
4. XML, the format for sending data from web server to the client. However, other formats can also be used, like HTML, JSON, or plain text.

In this paper, we will discuss the above components by highlighting each key feature and problem related to its implementation.

Stylesheet Language

In old web application, there is no such visible separation within content or data and the page layout. The HTML W3C specification makes it possible to style the content by modifying tag attributes. However, observing how the web technology evolve – especially design patterns in web development– it becomes clearer that each entity has its own functional purpose and should be processed in accordance with its specific inherences. Hence, data and layout should be handled separately.

CSS

CSS is a stylesheet language which is standardized by W3C[5]. This language is used as mechanism to modify data presentation, for e.g: fonts, color, padding, spacing, etc, in web documents. Each browser is equipped with engine to read and interpret CSS files and properties loaded into a web document.

CSS Problems

There are 3 levels of specifications, named CSS level 1, 2, and 3 respectively. Because CSS is interpreted by browser, the main problem is browser compatibility. Different browsers support different level of compatibility. An example of the list of CSS compatibility problems among browsers can be found in Quirksmode[6].

XSL

XSL (Extensible Stylesheet Language) is a language used to format document written in XML. A subset of this language called XSLT (XSL Transformations) is usually used to convert XML document into HTML or XHTML document for webpage. Standard for XSL is established by W3C[7].

XSL Problems

Because XSL transformation requires specific parser/processor, problem related with XSL is commonly related with parsing. However, there is also performance problem when transforming big document as written by Schafer[8].

Javascript

Originally built by Netscape and later standardized by ECMA under ECMA-262 specification[9]. This language is also known by several nicknames[10]. Javascript is running at client side and is interpreted by browser.

XMLHttpRequest Object

XMLHttpRequest (XHR) object is the essence of Ajax technology. This object allows web pages to request information from a server by using a set of functions written in javascript. This object is first implemented by Microsoft as ActiveX object in IE5 for Windows. Support for XHR in other browsers is added in Mozilla 1.0 and Safari 1.2. It can be said that these days, modern browsers have already supported XHR. W3C as an institution which stipulates web standards has also written the specifications for XmlHttpRequest[11].

Standard methods in XHR object can be seen from the table below[11,12]:

Method	Description
`abort()`	The current request.
`getAllResponseHeaders()`	Returns all the response headers for the HTTP request as key/value pairs.
`getResponseHeader("header")`	Returns the string value of the specified header.
`open("method", "url")`	Sets the stage for a call to the server. The method argument can be either `GET`, `POST`, or `PUT`. The `url` argument can be relative or absolute. This method includes three optional arguments.
`send(content)`	Sends the request to the server.
`setRequestHeader("header", "value")`	Sets the specified header to the supplied value. `open()` must be called before attempting to set any headers.

Above methods in details:
void open (string method, string url, boolean asynch, string username, string password)
This method sets up the call to the server which is represented in the url. Available methods are GET, POST, and PUT. The third parameter synch is used to hint whether the call is asynchronous (default value) or not. The last two parameters are used if the call requires user authentication.

void send (content)
This method makes the request to the server. If the request was declared as asynchronous, this method returns immediately, otherwise it will wait until response is received.

void setRequestHeader (string header, string value)
This method sets the value for a given header value in the HTTP request. If this method is used, it has to be called after *open()*.

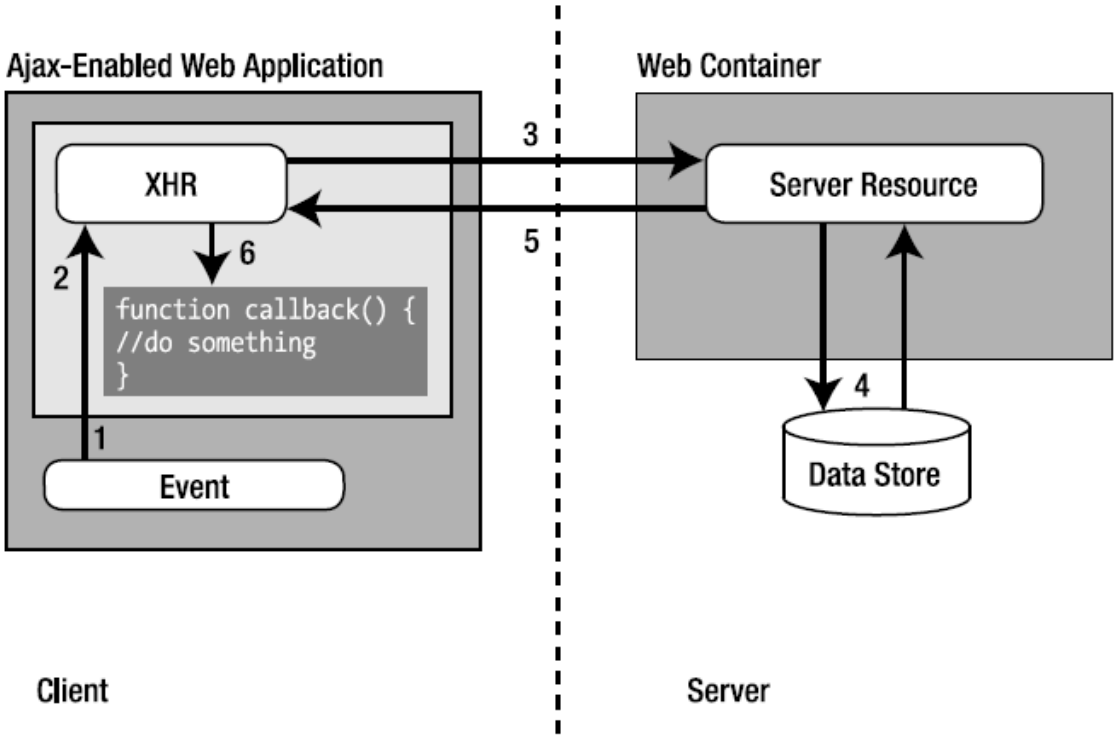
string getAllResponseHeaders()
This method will return a string containing response headers from the HTTP request. Headers include Content-Length, Date, and URI.

string getResponseHeader(string header)
This method is similar with previous method. The difference is it will only return header specified in the parameter.

In addition to the methods, XHR also has object properties as follows:

Property	Description
onreadystatechange	The event handler that fires at every state change, typically a call to a JavaScript function.
readyState	The state of the request. The five possible values are 0 = uninitialized, 1 = loading, 2 = loaded, 3 = interactive, and 4 = complete.
responseText	The response from the server as a string.
responseXML	The response from the server as XML. This object can be parsed and examined as a DOM object.
status	The HTTP status code from the server (that is, 200 for OK, 404 for Not Found, and so on).
statusText	The text version of the HTTP status code (that is, OK or Not Found, and so on).

XHR problems correlate with how AJAX-enabled application invokes the commands, i.e. sending request and processing the data. The diagram below provides interaction diagrams in an Ajax application[12]:



1. A client-side event triggers an Ajax event.

2. An instance of XHR object is created. The call is set up by the *open()* method and triggered by *send()* method.
3. A request is made to the server, which will involve server-side processing.
4. The server can do anything, for example accessing data store
5. The request is returned to the browser. Default Content-Type header is set to text/xml. However for more complex instance, it can be different.
6. XHR object is configured to call the function *callback()* when processing returned data. The function checks the *readyState* property on the XHR object and then looks at the status code returned from the server.

Performance Issues with XHR

Since XHR is a part of Javascript, it inherits Javascript problems. There are two main problems[4] regarding Javascript performance problems: 1)javascript execution speed and 2) javascript memory footprint. The first problem is classic since javascript is interpreted. For the second problem, Javascript is a memory-managed language which means that garbage-collection process automatically handles the allocation and deallocation of memory. Although this feature is time-saver or software architect, it can bring problem especially when a program is badly architected. A memory leak may occur during the operation and this can affect the stability of program and user convenience.

This propagates into the necessity of better application design. Profiling process is commonly added into the development stage in order to debug the application and search where memory leak can potentially occur. Profilers like Venkman and Firebug are example tools used for profiling AJAX-based applications.

Document Object Model (DOM)

HTML DOM represents the structure of a web page (the HTML elements) as a programmable tree structure, i.e. a series of related objects within a web document. W3C has released a recommendation[13] that provides for three different levels of DOM support, numbered 1 to 3. The higher the DOM level, the larger the feature set that is supported. A brief comparison can be seen below[13,14]:

- DOM level 1 : focuses on XML and HTML documents
- DOM level 2 : add stylesheet support to DOM level 1 and provides mechanism for applications to manipulate style information programmatically. This level also supports XML namespaces and defines an event model
- DOM level 3 : add supoorts for DTD and schemas on top of level 2

W3C DOM treats data as a tree of nodes, where each node has properties and methods. The recommendation is platform-and programming language- independent, hence only contains interfaces for manipulating the web documents instead of language specific methods.

DOM Issues

Problem around DOM mainly relates with browser compatibility. Because different browsers support different level of DOM specifications (for their Javascript parser), compatibility check and some workaround needs to be done in order to make sure the

document is properly presented across various browsers. Quirksmode also lists DOM compatibility problems among browsers[15].

On the other side, manipulating DOM can also result in performance issues. DOM implementation using Javascript for example, consumes significant memory and time for a large set of elements.

Extensible Markup Language (XML)

XML is a markup language which is also used to transport data across multi-platform environments. In regards to Ajax implementation, XML is used as an option for data format sent from server.

Issues with XML

There is no specific issue with compatibility. Main problem is related with data size. Many Ajax-driven web applications use other formats for transporting data from the server like the URL-encoded, raw text, SOAP, or JSON. When it comes to bandwidth usage and parsing speed, JSON usually outperforms other format[16].

References

[1] Ort, E., Basler, M. *AJAX Design Strategies*. http://java.sun.com/developer/technicalArticles/J2EE/AJAX/DesignStrategies/design-strategies.pdf

[2] Forum Systems Inc. *Ajax Application Security and Performance Considerations*. http://forumsys.com/papers/ForumSystems_AJAXwp-spring2006.pdf

[3] Zakas, Nicholas C., McPeak, J., Fawcett, J. *Professional Ajax*. Wrox Press. 2006

[4] Crane, D., Pascarello, E., James, D. *Ajax in Action*. Manning Publications Co. 2006

[5] *W3C CSS Specifications*. http://www.w3.org/Style/CSS/#specs

[6] *CSS – Contents and Compatibility*. http://www.quirksmode.org/css/contents.html

[7] *The Extensible Stylesheet Language Family (XSL)*. http://www.w3.org/Style/XSL/

[8] Schafer, U. *WHAT: An XSLT-based Infrastructure for the Integration of Natural Language Processing Components*. Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS). 2003

[9] *ECMA-262 Specifications*. http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf

[10] *Wikipedia Entry for ECMA Script*. http://en.wikipedia.org/wiki/ECMAScript

[11] *W3C Spefications for XMLHttpRequest Object*. http://www.w3.org/TR/XMLHttpRequest/

[12] Asleson, R., Schutta, Nathaniel T. *Foundations of Ajax*. Apress. 2006

[13] *W3C Document Object Model Specifications*. http://www.w3.org/DOM/DOMTR

[14] Jacobs, S. *Beginning XML with DOM and Ajax*. Apress. 2006

[15] *W3C DOM Compatibility Tables*. http://www.quirksmode.org/dom/compatibility.html

[16] Lauriat, Shawn M. *Advanced Ajax: Architecture and Best Practices*. Prentice Hall. 2008