AJAX: A NEW TWIST ON EXISTING TECHNOLOGIES

Mike Gilorma
Software Engineer
Apogee Labs, Inc.
North Wales, Pa USA

ABSTRACT

Asynchronous JavaScript and XML (AJAX) has improved web applications in a way that has enhanced performance and made the user experience more like that of a desktop application. As the performance of PCs increases and broadband Internet access is more prevalent, switching between web pages is less painful than ever. One of the biggest advantages of AJAX is the ability for a web application to update only a small piece of data without refreshing the whole page. AJAX also allows for piecewise validation of user entry as opposed to the standard form entry with which we have become so accustomed.

This paper describes how AJAX enabled applications are different from classic web applications and shows the advantages and disadvantages from both client and server sides of an AJAX enabled application. AJAX is not a new technology, but rather a new approach to web applications that uses standards already in place for XHTML, CSS, DOM, XML, and JavaScript. It is this new approach that eliminates the full page refresh that was so commonplace and now gives web applications the ability to look and feel more like desktop applications.

KEY WORDS

AJAX, XML, JavaScript, Asynchronous

INTRODUCTION

AJAX is a buzzword in the world of web applications. Just about everybody who has been on the web within the last couple of years has already been exposed to this technology, and most don't even know it. Anyone who uses Google to search the web for information has been using AJAX for quite some time. Both Google Suggest and Google Maps use this technology. If you haven't tried these tools, try Google Suggest and notice that it updates almost as fast as you can type. Look up your neighborhood on Google Maps and notice that you can zoom in, zoom out, grab and drag the map all around without having to refresh the page. That is AJAX at work. The biggest benefit of using an AJAX enabled application is the ability to update only pieces of information relevant to the web page without having to refresh the whole page from the server. It is this fact about AJAX that allows web pages to be more user friendly, quicker, and gives a web application the look and feel of a desktop application.

WHAT IS AJAX AND HOW DOES IT WORK?

The term AJAX (or Ajax) is an acronym that stands for Asynchronous JavaScript and XML. AJAX was first publicly used by Jesse James Garnet in February 2005[1]. He used the term to describe a suite of technologies he was proposing to a client. Similar technology was around nearly a decade before that, and it is that idea which has grown into AJAX and will continue to evolve for years to come. AJAX incorporates a combination of the following technologies:

- XHTML and CSS – for the look and feel of the Web Application
- The DOM for dynamic display and interactions
- The XMLHttpRequest object to Asynchronously exchange data with a web server
- XML, JSON, preformatted HTML, or even plain text as the format for transferring data between the server and client
- JavaScript to bind all the technologies together

In a standard web application, most user interactions trigger a request to the server where the server performs processing. After the server has completed its processing, it returns an HTML page to the client which is then displayed on the user's web browser. We have all experienced the joy of filling out a long form on line and clicking submit to send it to the server only to find that you have an error somewhere on the form and must go back to fix it.

In a world of instant gratification, such as ours, why would users ever want to wait for a server to do what it needs to do when they could be doing other things? In my example above, the user would wait for the server to check out the form and report any errors, then go back and fix those errors. From a user experience standpoint, it would make sense to check the information one piece at a time without reloading the whole page so that the user doesn't have to wait for the server to check all of the data at one time. AJAX allows servers and clients to communicate through an intermediate layer that is invisible to the user and results in a much more fluid experience.

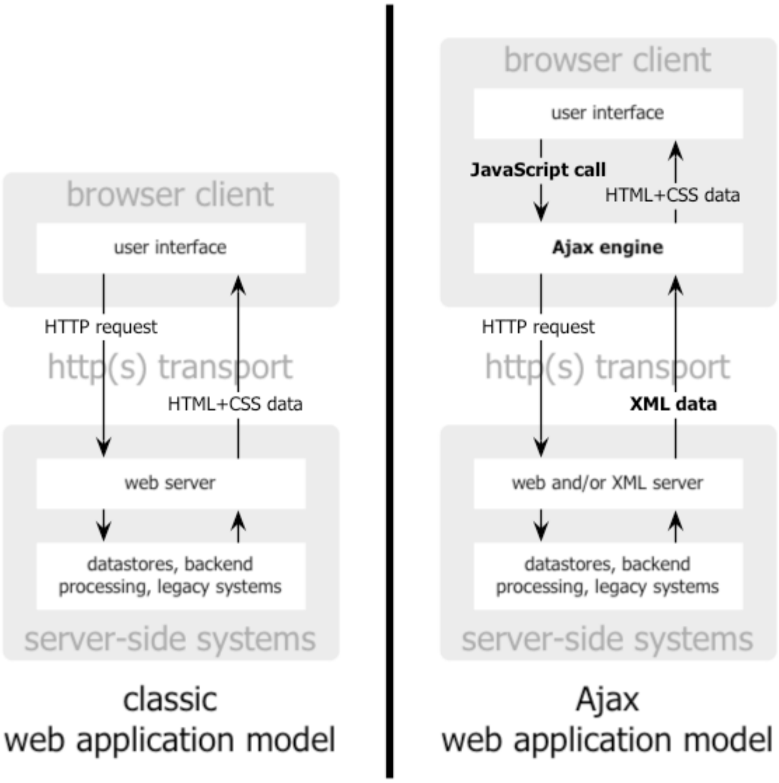

Figure 1: Classic web model vs. Ajax web application model [2]

In Figure 1 it can be seen how an AJAX web application is different than a classic web application. AJAX is responsible for being the intermediary between the client and the server. Instead of reloading the web page after each user interaction, an AJAX enabled web page is loaded only once at the start of the session. AJAX changes the interaction between the client and the server in terms of what data is actually exchanged and how it is exchanged.

When AJAX is the intermediary between the client and the server it allows the server to do work without the user noticing. AJAX allows the interaction to become asynchronous, whereby the user's interaction is independent of communication with the server. In the example of filling out a large form on line, each entry made by the user would be validated immediately instead of all of the data being verified when the user clicks submit. This means that every user action that would normally be a call to the server is now a call through JavaScript to AJAX. AJAX then determines whether it needs to talk to the server, or in the case of simple data validation, AJAX does the work itself. If AJAX does need to talk to the server, it does so asynchronously using a type of formatting (usually XML) to move data between the server and client.

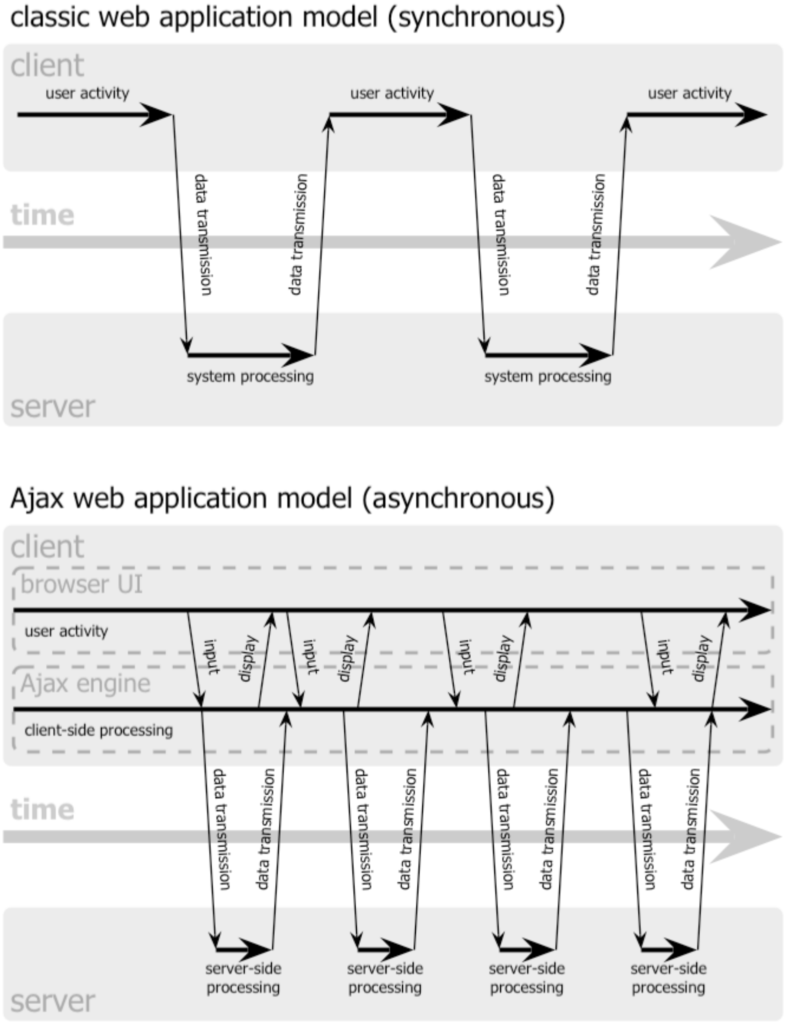

Figure 2: classic synchronous model vs Ajax asynchronous model [2]

In Figure 2 it can be seen that, unlike a classic web application where the user must wait for the server's response, an AJAX enabled application eliminates user wait time due the asynchronous communication between the user, AJAX, and the server.

ADVANTAGES WITH AJAX

Probably the biggest advantage of AJAX is the improved user experience that it offers. It allows users to have an overall smoother experience, much like a traditional desktop application. In fact the best thing about an AJAX enabled application is, that as a user, AJAX is invisible and it makes the server seem more responsive. Another advantage of AJAX is that it is supported by most major web browsers, thus providing a cross platform application that can be run on most Operating Systems.

From the server side of an AJAX enabled application, an advantage includes the ability to update information without refreshing the whole page which directly lowers the bandwidth usage. This is due simply to the fact that AJAX allows the server to update small pieces of data to the client without having to refresh the whole HTML page and reload all the data. For instance, if a user is watching a status page (50 kB in size) that refreshes twice a second for an hour, that is 6 Mb worth of data transferred in an hour. However, that page might only require 1 kB worth of data to update the status fields (100 character buffers of 10 characters each), for a total of 120 kB worth of data transferred in an hour. In this very general example, AJAX would yield a 98% reduction in bandwidth usage.

DISADVANTAGES WITH AJAX

One major advantage of an AJAX enabled application, the support of most major web browsers, comes with the disadvantage that it does not operate exactly the same across all browsers. Since AJAX relies on JavaScript, it is not uncommon to see JavaScript code that is written twice (once for IE and once for Mozilla compatibles). This is a consideration with any web based application and requires careful programing and testing for optimum operation across platforms. Another issue resides with the operation of the back button in regards to a dynamically loaded page. When clicked, the back button could take the user to an unexpected page or clear the current state of the page they were on. Again, these issues can be solved using careful programming and solid testing practices.

Another possible issue with using AJAX resides with the asynchronous nature of this technology. Network latency needs to be considered when developing an AJAX enabled web application due to the possibility of poor preloading of data and/or handling of the XMLHttpRequest object which may result in delays in the interface of the application.

DEVELOPING WITH AJAX

When faced with developing a new product and the challenge of writing a remote control interface for that product, there are always a number of questions to take into consideration. What OS is the application going to be run on? Does the application need to run on different platforms? What are the minimum requirements of that system to run the application? Would the users have to download extra software to run my application? What kind of system latency is acceptable? These and other questions all need to be answered when designing any remote control interface. Because AJAX is supported on most web browsers, most users wouldn't need to do anything but type in the IP Address for the server that they were looking for and they would be up and running. Since the web application is completely hosted on the server, it does not require a user with multiple computers in different locations to install software on each computer. All that is required is the computer have a web browser. Another added benefit of running a web application to control a remote device is the tabbing feature. Many browsers provide this feature which allows you to control multiple devices by simply opening more tabs in the browser. All of these benefits are available to any web application. However, the desktop feel and piece-wise updating of an AJAX enabled web application make it an attractive option, not only from the users standpoint, but from a development standpoint as well. Unlike desktop applications which usually are started from scratch for each new product, an AJAX enabled web application and its standard set of web tools enable a rapid development environment that is easily built upon and tested.

AJAX AND TELEMETRY SYSTEMS

In the world of telemetry, users have a variety of operating systems which do not run the same software. This is why many telemetry systems have begun to implement HTML servers. This ensures that the client software is a standard, installed software which is compatible with the server. The user does not have to install any custom software to remotely communicate with the telemetry system. Without AJAX, when changing parameters of the system via the web browser, the user would have to click the submit button to send the entire setup on that page to the server. An AJAX enabled application would allow the user to have each parameter verified as they changed them. AJAX would verify each parameter on the client side, then send the change to the server after it passed initial validation. AJAX can also report to the user through a visual cue, such as the parameter flashing green for one second or a pop up message, that the command was successfully set on the server. If the user was monitoring a status page without

AJAX, they would either have to manually refresh the page or wait for the refresh timeout to happen. In either case the server would have to resend the entire page of information across the network. Most of these refreshes can only occur once per second. However, with an AJAX enabled application, because it only updates small pieces of information at a time, the status fields can be updated multiple times per second. This gives the application more real time updates which is a necessity in telemetry applications.

CONCLUSION

While this paper was more of an overview of the theory of AJAX, and the advantages and disadvantages of using an AJAX enabled web application, I hope I have presented enough information to give an idea of how AJAX is a useful technology. That isn't to say that it is the best solution to all your programming needs, however it does provide a great set of tools for improving web applications in general. It also opens up web applications to a wider variety of remote control solutions that, in my opinion, would not be as attractive without the benefits of AJAX. There is a wealth of AJAX examples for all platforms on line. To find examples, simply search for “Ajax examples”.

REFERENCES

1. Mozilla developer homepage for AJAX: http://developer.mozilla.org/en/docs/AJAX
2. Jesse James Garnet's article introducing AJAX: http://www.adaptivepath.com/publications/essays/archives/000385.php
3. Wikipedia: The Free Encyclopedia: http://en.wikipedia.org/wiki/Ajax_(programming)

GLOSSARY OF TERMS

AJAX	Asynchronous JavaScript and XML
XML	Extensible Markup Language – a general purpose markup language that allows users to define their own tags, primarily used for data sharing across different systems
JavaScript	a scripting language mostly used for client side web development
HTML	HyperText Markup Language – main markup language used for the creation of web pages

XHTML Extensible HyperText Markup Language – has the same expressions and abilities as HTML, but conforms to XML syntax

CSS Cascading Style Sheets – a style sheet language used to describe the presentation of documents written in a markup language

DOM Document Object Model – an object model used for representing XML and HTML related formats

JSON JavaScript Object Notation – a lightweight computer data interchange format used for representing objects and other data structures

XMLHttpRequest The key component of AJAX that allows JavaScript or another scripting language to transfer XML data from a web server by independently communicating between the server and client