



AJAX: What, When and Why

White Paper

Abstract

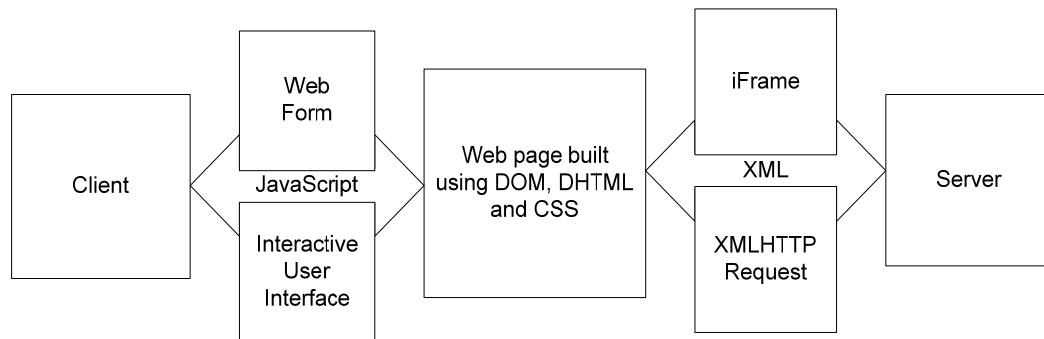
This paper describes the new upcoming technology called AJAX and compares it to other similar technologies.

AJAX is a relatively new method to create rich internet applications with responsive interfaces that allow the designer to take a lot of the tedium out of using web-based mediums for day-to-day data interaction. The scope of this paper will be to provide information about what AJAX is and what it does, when it should and shouldn't be used and what the future looks like for AJAX in comparison to the present and future alternatives. This will help developers determine the value of using AJAX in their projects and provide information about the proper use of this technology. This report will assume a basic knowledge with web forms and the design of web applications.

AJAX is an acronym for Asynchronous JavaScript and XML. It is not a technology in itself, but rather a suite of technologies. The combination of JavaScript and XML allows for a responsive user interface. It requires more development time than a traditional web solution; however a well-designed AJAX user interface makes the traditional form-based web application seem almost archaic.

What is AJAX and what does it do?

As mentioned in the introduction, AJAX is a suite of technologies used to create rich internet applications. It uses JavaScript to gather and show data to the user. It uses XML transmitted either in the XMLHttpRequest or IFrame objects to communicate with the server. Together they allow the user to do a lot of things interactively with one web page. The diagram illustrates on which side of the data transaction XML and JavaScript function.



As opposed to traditional form-based interaction, in an AJAX application, JavaScript provides a lot of the interaction. While elements of web-forms are often used, in well-designed AJAX, JavaScript becomes the medium for interaction and the web page only provides the layout for the data and the way it will be displayed. JavaScript is in charge of the input and output events and displaying the data associated with those events. This allows one web page to display and manipulate a lot of data without having to be reloaded.

While JavaScript can manage the user interface, it cannot communicate with the server. This is where XML comes into play. This paper will discuss XML

but any method of conveying data can be used if the situation requires it. XML serves as the communication between the web page and the server. This makes a compact response object because the server only has to send an XML object representing the data that will be displayed in the web page already in the browser. By creating a smaller response object and putting some of the processing associated with display on the client computer, resources from the server are saved. Finally, since the client doesn't need to load a new page, the data is displayed much quicker, creating a much faster interface that is much more enriching for the user.

It's not hard to see how many applications this would have. A great example would be a form that allows users to look at their email. In the traditional web form model, a user would have to view a new web page every time they wanted to cycle to a new email. Using an AJAX web application, the user could cycle through emails without ever having to load a new web page. This is an example of a simple AJAX application although it could go so far as to implement a menu to select which email to view, a way to create and send emails, and a method of instantly notifying a user of new emails into the same web page, never having to be refreshed. Obviously, this would create a faster and more responsive web experience.

A time and a place...

With all of this power, why not use AJAX for everything a webpage does? Since all AJAX provides is a method of synchronizing and transferring data on request from the user and then manipulating it into the display, it shouldn't be used for everything. While it may be tempting to create a web site that uses AJAX to provide on-the-fly search functionality and customizable navigation, AJAX development takes a lot of time, and the charm it provides will quickly wear off. Here are some situations where the development of an AJAX solution will provide the maximum return from development costs.

Forms: There is nothing more tedious than having to use the traditional web-form method for data entry. By applying AJAX to web forms, an amazing amount of functionality is added. Imagine being able to create a Windows application style interface to receive information from users on the web. When compared to a standard web interface, this seems much better. AJAX in forms has an obvious use anywhere multiple instances of the same data type will be received or displayed. It can also be used where modifying information will change information somewhere else to provide real-time updates.

Rapid Interaction: AJAX allows a web page to be used to create rapid interaction with the user. AJAX can make instant messaging on a web page a reality. AJAX can be used to allow a web page to serve as an event-based alarm, such as a stock ticker that warns the user when a price is reached. AJAX can be used in any situation where a browser is typically auto-refreshed which is

another way AJAX moves the functionality in creating web applications closer to the freedom had when creating Windows applications.

Avoiding Refreshing: AJAX should be used in any situation that requires a lot of browser refreshing. Loading large lists for several small manipulations or working through a long tree view on a forum are tedious examples of situations where traditional web solutions spend a lot of time refreshing. Filling out small request or voting form on the side of a larger feature is so slow and distracting users avoid doing it. AJAX provides an elegant solution to all of these problems by being able completely avoid browser refreshing.

Auto Complete: This is tricky situation. AJAX can provide a great solution here when it provides auto completion of things that are similar between many users or a situation where users would use different language to describe the same thing. However, the time spent developing something like this for a website may just end up overlapping functionality already provided by most major browsers. If done correctly however, using AJAX to provide a well placed auto completion text box is a great idea, the key is to make sure that major browser's auto-complete functionality isn't duplicated and that the auto-completion being provided is worth the time to develop it.

AJAX has a fairly slow development time and it is in a developer's best interest to use it only where it will provide a benefit that is worth the extra development time. For example, using AJAX to manage the navigation on a website is overkill. Using the DOM, CSS and XHTML is an easier and faster way to develop this. As outlined before, typically AJAX shouldn't be used to provide the shape of the interface, but should be used to handle the interaction between the Client and the Server with the web page as the middle ground.

AJAX's Role Now and in the Future

While AJAX provides the most compatible solution to the problem of how to provide fast and flexible interaction with the user, it is not the only solution available. Macromedia's Flash plug-in allows a lot of freedom in user interface creation. So do Java applets. Java applets also allow for complex server communications. Comparing AJAX to other solutions, which this section attempts to do, shows the major strength of AJAX is compatibility, but it isn't as flexible as other solutions.

The first solution typically used to create an enhanced user interface is Macromedia Flash. Flash requires a browser plug-in distributed by Macromedia who is now a subsidiary of Adobe and is in the process of re-branding. The fact a plug-in needs to be installed immediately creates some distrust from a substantial amount of users. Coupled with the very distinct look Flash sites have, it has lead to Flash being almost taboo to use in corporate web design. Some users also describe it as tacky or cliché. Asking whether or not to use Flash is bound to elicit a strong opinion from nearly any web designer.

A good Flash designer is usually hard to find. While it is easy to learn to create Flash sites, it isn't always an intuitive process. It takes a big skill set, consisting of animation, graphics design, web editing, object, artistic creativity and script programming in a proprietary language. To add to the grab-bag of skills required, Flash can be used in tandem with XML in a technique known as AFLAX. This provides the same server communication advantage that AJAX provides, but also allows the expanded user interface of Flash. The new Flash development suite, Flex, should allow for this to be done more easily. Ultimately, Flash is a good product that some would say is good at lights and smoke, while others would say is great at providing the stage to perform on. It is up to the designer to decide if it's right for the job.

Another solution that certainly provides all the functionality AJAX does is Java applets. Java applets provide desktop functionality in a web page. They are basically an executable that is ran and displayed in a web page container. While making them is similar to creating a desktop application, there are some differences. The program is ultimately an application on the client machine but much of the data comes from the server. It requires a weighty framework to operate and it cannot readily access many of the client system's resources. Java applets are slow to load usually, but once they are running they're very fast. They are probably overkill in situations where AJAX can be used, but they provide one-of-a-kind functionality for programs that display a lot of variables in real-time. They are also often used in creating interactive games on the web. Java applets also are a security concern to most IT professionals therefore they are not allowed on a significant number of workstations, making compatibility a large concern.

Those are the two major existing alternatives to AJAX. There are several alternatives that aren't in wide-implementation yet, because of browser dependence or because they are still in development.

XUL is Mozilla's attempt to create a highly functional markup language. It allows for fairly sophisticated user interfaces, but its lack of flexibility has some developers finding insurmountable obstacles in designing complex web applications. It only operates on Mozilla based browsers.

XAML is Microsoft's attempt to create a highly functional markup language. It is still in development and is slated to be released with Windows Vista. While it is functionally an answer to AJAX it is exciting with Microsoft's current trend toward rapid application development tools to think that AJAX's functionality could be provided with lower development costs. While it will certainly be more functional and configurable than XUL, it will also be dependent on WinFX meaning it won't work on browsers that don't use Avalon.

SVG stands for Scalable Vector Graphics. The standard for SVG is being developed by the W3C but is very slowly being integrated into browsers. SVG uses XML to describe 2-dimensional vector graphics. If SVG becomes integrated like other W3C technologies have, then it will provide a default standard for complex graphics in web browsers and in tandem with JavaScript it will provide all of the functionality that Flash does. Since it is an addition to the

DOM it will be more of a partner to AJAX than a replacement, but is important to look at regardless.

That gives a lot of options for expanded user interfaces in web applications. Choosing the method used isn't usually a tough decision. It's often based on the preference of the designer. However, some critical thinking should be made before a decision is made based simply on the preference of the designer.

AJAX requires a lot of time to develop however the skills used to develop AJAX are common place amongst web developers. Learning to apply JavaScript and XML together doesn't take a tremendous amount of time to learn. Java Applets and Flash require specialized developers but developing the solution in them can be cheaper. XUL, XAML and SVG require an even rarer skill set, but XUL and XAML provide the fastest development time. After considering development costs, compatibility becomes the main concern.

AJAX is native to most modern browsers. Technically, Internet Explorer requires a plug-in to use AJAX, but its installation and use is transparent to the novice end user. Flash and java applets both require plug-ins to operate. Some users don't want to install plug-ins or some administrators don't allow them. XUL and XAML are both browser dependent. While XAML will have the larger market share, XUL has a small but vocal user base that is substantial to commercial sites. SVG currently has a very tiny user base.

It is likely that AJAX is going to be with us for a while. While alternatives exist today, nothing can really do what it does. Developing for AJAX now is a good idea. Flash is currently the most viable alternative, but with new, promising technologies being released, Flash stands to see some of its market share cut. Java applets are becoming less and less common place. Out of the newest crop of alternatives, XUL, XAML and SVG, XAML appears to be the most promising in functionality and will likely have the broadest user base, but as SVG is implemented into major browsers, its potential is being show. XUL, only being compatible with a small share of browsers and lacking any significant development base in web applications, falls behind and shouldn't be used in the development of web applications under normal circumstances.

Comparison of Rich Internet Application Technologies			
Technology	Compatibility	User Base	Development Time
AJAX	Native to most browsers	Largest	Slow
Flash	Requires plug-in	Large	Medium
Java Applet	Requires plug-in	Medium	Medium
XUL	Native to Mozilla browsers	Small	Fast
XAML	Native to Windows Vista, Requires WinFX SDK on XP	Tiny	Fastest
SVG	Most browsers require plug-in	Tiny	Slow

The chart restates that AJAX is the most compatible solution available to developers today however it often takes the most time to develop. In the near future, combined with SVG, it will provide W3C-DOM compliant browsers the

ability to create powerful web applications without the use of plug-ins or proprietary software.

Conclusion

AJAX is one of the most valuable tools in a web developer's toolbox. With AJAX, a developer is able to move away from the traditional submit/refresh cycle that traditional web forms tend to lock design into. AJAX allows functionality similar to that which used to have to be provided by browser plug-ins. It is reliable to develop in because it is compatible and nearly everyone using the web can use AJAX. Finally, it is poised to be an important part of web development for a long time. AJAX is the most compatible web application solution available today. For intranet applications, something like XAML or XUL may be a better idea, but for standard internet applications, the only reasonable choices currently are AJAX and Flash.

This document is for informational purposes only. Tometa Software, Inc. MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2006 Tometa Software, Inc. All rights reserved.