Building Rich Web Applications with Ajax

Linda Dailey Paulson

As the Internet has become more mature, rich applications featuring responsive user interfaces and interactive capabilities have become increasingly popular. The capabilities represent a way to make programs easier to use and more functional, thus enhancing the user experience.

Developers have used a variety of applications from companies such as Macromedia, Microsoft, and Sun Microsystems to add these capabilities in the past, as discussed in the "Developing Large-Scale Rich Web Applications" sidebar.

However, Web applications have generally exhibited problems such as slow performance and limited interactivity, particularly when compared to typical desktop applications, noted Nate Root, research director for Forrester Research, a market analysis firm.

Now, developers are going back to the future by building Web applications using Ajax (Asynchronous JavaScript and XML), a set of technologies mostly developed in the 1990s. A key advantage of Ajax applications is that they look and act more like desktop applications, according to Root.

Proponents argue that Ajax applications perform better than traditional Web programs. As an example, Ajax applications can add or retrieve new data for a page it is working with and the page will update immediately without reloading. For instance, when users
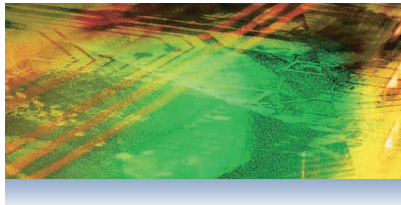
hold down the left mouse button and slide the cursor over an image on the Ajax-based Google Maps beta site (http://maps.google.com) to retrieve a part of the map not shown on the screen, the updates occur smoothly and the image appears to move and change immediately. With typical Web applications, users must spend time waiting for entire pages to reload, even for small changes.

When companies began working with the technology several years ago, before the approach even had the name Ajax, they used it for smaller, less important applications. However, as the component technologies have improved, Google and a number of other companies have started using Ajax for more important enterprise applications.

In addition to its map site, Google has worked with Ajax to build applications such as Gmail and Google Groups, a community and discussion service, said Bret Taylor, Google Maps product manager.

Flickr uses Ajax in some parts of its Web site, on which users post and share photographs. For example, Ajax

enables the site to let users add and view photo annotations. Expedia has produced features such as pop-up calendars on its travel site via Ajax.

All major browsers now support the technology. Thus, Ajax could pose a threat to Microsoft, Macromedia, and Sun. However, while some companies may decide Ajax is particularly useful for certain kinds of applications, industry observers say it won't be suitable for all types. And in some cases, companies may use Ajax to complement other Web-application approaches.

Meanwhile, Ajax still faces several technical challenges, such as usage complexity and security.

WHAT AJAX IS

Developers use Ajax technologies to build Web applications with improved performance and interactivity, as well as responsive user interfaces.

The applications offer functionality generally available in desktop software but not on the Web, which was designed for communications simplicity, not to enable the development of programs with enhanced capabilities.

Component technologies

Most of Ajax's component Web technologies were developed and standardized during the past 10 years. These technologies have improved recently, making them more suitable for enterprise use.

Dynamic HTML. Ajax applications take advantage of dynamic HTML, which consists of HTML, cascading stylesheets, and JavaScript glued together with the document object model.

The technology describes HTML extensions that designers can use to develop dynamic Web pages that are more animated than those using previous HTML versions. For example, when a cursor passes over a DHTML page, a color might change or text might get bigger. Also, a user could drag and drop images to different places.

XML. Ajax uses XML to encode data for transfer between a server and a

browser or client application. The W3C started work on XML in 1996 to enable cross-platform data interoperability over the Internet. The consortium approved the standard's first version in 1998.

XML is a markup metalanguage that can define a set of languages for use with structured data in online documents. Any organization can develop an XML-based language with its own set of markup tags.

Cascading stylesheets. A W3C standard since 1996, CSS gives Web site developers and users more control over how browsers display pages. Developers use CSS to create stylesheets that define how different page elements, such as headers and links, appear. Multiple stylesheets can apply to the same Web page.

Document object model. The DOM, a W3C standard since 1998, is a programming interface that lets developers create and modify HTML and XML documents as sets of program objects, which makes it easier to design Web pages that users can manipulate.

The DOM defines the attributes associated with each object, as well as the ways in which users can interact with objects. DHTML works with the DOM to dynamically change the appearance of Web pages.

Working with the DOM makes Ajax applications particularly responsive for users.

JavaScript. Released in 1995 by Netscape and Sun, JavaScript interacts with HTML code and makes Web pages and Ajax applications more active. For example, the technology can cause a linked page to appear automatically in a popup window or let a mouse rollover change text or images. Developers can embed JavaScript, which is openly and freely available, in HTML pages.

Ajax uses asynchronous JavaScript, which an HTML page can use to make calls asynchronously to the server from which it was loaded to fetch XML documents. This capability lets an application make a server call,

retrieve new data, and simultaneously update the Web page without having to reload all the contents, all while the user continues interacting with the program.
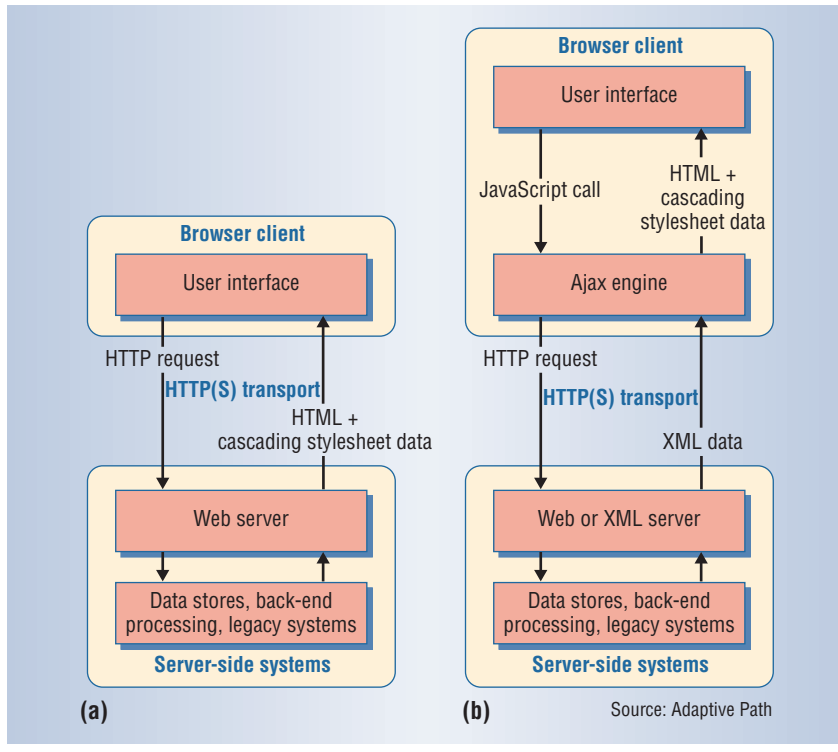
Enterprise application developers have become more interested in working with JavaScript because users have removed some of the technology's bugs and developed workarounds for various problems.

Because JavaScript is a cross-platform scripting language, Ajax applications require no plug-ins, unlike Macromedia

Flash and other proprietary Web application technologies.

XMLHttpRequest. Systems can use JavaScript-based XMLHttpRequest objects to make HTTP requests and receive responses quickly and in the background, without the user experiencing any visual interruptions. Thus, Web pages can get new information from servers instantly without having to completely reload.

For example, users of an application with XMLHttpRequest objects could type in a centigrade amount in one box

---

Developing Large-Scale Rich Web Applications

Companies are beginning to use Ajax and its component technologies to develop large-scale, Web-based enterprise applications. Traditionally, they have used Ajax for smaller programs and have developed more important software with technologies by Microsoft, Macromedia, and Sun Microsystems.

Microsoft

Microsoft is reportedly trying to simplify the development of rich Web applications via a project code-named Atlas. Atlas will provide tools to be used with the company's ASP.NET, which developers use to create Web pages whose elements are treated as objects.

Microsoft declined to comment for this article.

Greg DeMichillie, lead analyst for Directions on Microsoft, a market research firm, said there is little information about Atlas except that "it will work by providing much of the boilerplate code that an Ajax developer would otherwise have to write, such as determining which browser is being used and adjusting the JavaScript sent to the client accordingly. That makes Ajax applications easier to write because developers can focus on code specific to their application."

Macromedia's Flash

Macromedia's Flash is a popular type of Web authoring software that creates vector-graphics-based animation programs.

According to Kevin Lynch, Macromedia's chief software architect, Ajax won't supplant Flash because the technologies don't do all the same things. For example, he noted, unlike Ajax, Flash supports audio and video. Also, he said, Flash is more widely available.

Sun's Java

Java-based applications offer some advantages over Ajax-based programs, according to Tor Norbye, a senior staff engineer with Sun. For example, he explained, there are reusable Java components and toolkits, which is not yet the case for Ajax.

In many cases, Norbye noted, Java supports Ajax, as in the June 2005 release of Java Studio Creator. And, he contended, "The ideal architecture for Ajax today is Java on the server, where the interesting processing happens, and JavaScript in the browser."

Thus, he concluded, "I think there's room for both."

Figure 1. In (a) a traditional Web application, user actions trigger an HTTP request to a Web server, which processes the request and returns an HTML page to the client. Additional requests lock up the application until the system updates the page. (b) Ajax applications create a JavaScript-based engine that runs on the browser. The engine intercepts user inputs, displays requested material, and handles many interactions on the client side. If the engine needs more data, it requests material from the server in the background, while letting the user continue to interact with the application.

of a temperature-conversion application and have the Fahrenheit amount appear instantly in another box.

Various browsers—including recent versions of Internet Explorer, Mozilla's Firefox, Netscape, and Apple Computer's Safari—work with XMLHttp-Request.

Working with Ajax

In the classic Web application model, user actions trigger an HTTP request to a Web server, which processes the request and returns an HTML page to the client. This makes technical sense but doesn't always provide a great user experience because, for example, it limits interactivity and requires Web pages to reload for every piece of new data.

Ajax applications create a JavaScript-based engine that runs on the browser, explained Forrester's Root. Instead of loading a traditional Web page, the browser loads the engine, which then displays the requested material, as Figure 1 shows.

The engine intercepts user inputs and handles many interactions, such as simple data validation, on the client side, Root said. If the engine needs more data, it requests the material from the server in the background without locking up the user interface. The engine thus lets users interact with an application independently of server communication, reducing server response wait times.

THE UPSIDE

Proponents say that Ajax applications perform better than today's Web applications.

Generally this is true because Ajax applications are more responsive to user actions and the programs don't experience page-reloading-related interruptions, explained Jesse James Garrett, cofounder and director of user experience strategy for the Adaptive Path consultancy. Garrett coined the term Ajax earlier this year.

Also, Garrett said, Ajax applications are usually fast because the approach minimizes traffic to the server by sending and requesting just the minimum amount of data needed.

Proponents say another benefit of working with Ajax is that more developers have experience with its component technologies than with other Web-application-development approaches.

In addition, said Håkon Wium Lie, chief technology officer for browser maker Opera Software, because Ajax works across platforms, developers generally can write applications once to run via the Web on many users' systems, regardless of platform.

THE DOWNSIDE

According to analyst Ray Valdes with market research firm Gartner Inc., developers can add Ajax techniques piecemeal to an existing system, one code snippet at a time. However, he added, trying to implement all the techniques at once in a complex project "approaches the rocket-science level of difficulty."

"Your average developer is not going to be able to figure it out," Valdes said. "Only a small, elite group has the smarts to do it in a comprehensive, complete way."

In general, though, Garrett said, Ajax's learning curve for professional designers is not steep. However, he noted, the technology is immature and still needs toolkits and frameworks.

In addition, he said, "Because there are so few prebuilt Ajax interface components, developers will have to custom build most interfaces for each application. Some will be successful, some won't."

Ajax also has raised some security concerns. For example, Forrester's Root said, its component technologies have been around for years, but they are now being used in unproven ways that might make them more vulnerable to security breaches.

"The two major compatibility issues with Ajax," said Garrett, "are differences in JavaScript implementations across browsers and providing alternate means of accessing the applications with older browsers that don't fully support modern Ajax features. In both cases, additional development effort is required."

Another challenge, noted Sandy Leung, Yahoo's product manager for next-generation interfaces, is that users must get accustomed to Ajax applications that don't perform like traditional Web applications.

And because Ajax doesn't reload entire pages to add new material, he said, search engines might not index some important information.

Moreover, Ajax isn't useful for some applications. For example, it can't be used for audio and video streaming, as neither HTML nor JavaScript have an audio or video API, explained Sun senior staff engineer Tor Norbye.

WHAT'S NEXT

Some sources say the recent attention to Ajax has also brought attention to rich Web applications, which will help vendors using other development approaches, Garrett said.

According to Norbye, better browsers, tools, and network performance will improve Ajax's capabilities in the future.

Ajax could find various uses. For example, vendors could use it to build Web-based versions of desktop applications. This way, companies could make software widely available to employees via a network and thus avoid spending the time and money required to install applications on every computer. Ajax also could be useful for the growing number of Web applications for mobile devices.

However, predicted Root, while Ajax may prove interesting to developers now, they may turn to versions of Flash and other technologies in the future because, for example, Flash supports audio, video, advanced vector graphics, and other capabilities that Ajax can't offer.

Because they find it useful, companies will create more Ajax-based applications in the near future, predicted Kevin Lynch, Macromedia's chief software architect.

"We're now entering a period of experimentation," said Adaptive Path's Garrett. "A lot of people in the past six months became aware of the possibilities that Ajax opens up for them. Developers are pushing at the boundaries of what they can do with it."

Ajax will do well as long as it is competitive with other approaches. For example, Google's Taylor said, his company will use Ajax as long as it likes what the technology offers. He explained, "We will use whatever technology platform provides the richest user experience possible." ■

Linda Dailey Paulson is a freelance technology writer who is based in Ventura, California. Contact her at ldpaulson@yahoo.com.

Editor: Lee Garber, *Computer*, l.garber@computer.org