

EMERGING TECHNOLOGIES

Ajax and Firefox: New Web Applications and Browsers

Robert Godwin-Jones
Virginia Commonwealth University

Five years ago (May, 2000), this column dealt with "[Web Browser Trends and Technologies](#)." Ten years ago marks the arrival of several major Web players including [Yahoo](#), [eBay](#), and [Amazon](#). It was also in 1995 that the red-hot [Netscape](#) IPO ignited the start of the tech bubble in the U.S. Stock Exchange that burst in 2000. Given the anniversaries, it seems like a good time to revisit the topic of Web technologies, especially since some significant changes are underway. Alternative browsers are gaining significant market share, and both Apple and Microsoft are releasing OS upgrades which portend some interesting changes in Web development. Of particular interest for language learning professionals may be new developments in the area of Web browser based applications, particularly using an approach dubbed "Ajax."

New Browser Wars?

In the spring of 2000, Netscape had just released the beta of Netscape 6, to a largely indifferent Web community. This spring, Netscape has again released a beta, this time of [Netscape 8](#), which uses a dramatically different approach, based largely on a popular new browser, [Firefox](#). In 2000, several browsers had been newly released: [Opera](#) and iCab. Today [Safari](#) (from Apple) and the open-source [Firefox](#) (from Mozilla) are the new players and are attracting considerable interest. The increasing popularity of Apple's Safari is understandable, given Microsoft's decision not to update the aging IE for the Mac, while Apple regularly updates its browser with new features and capabilities. It is Firefox, however, that has made the biggest splash in the media, with a [cover story](#), for example, in a recent issue of Wired Magazine. This is largely because for the first time the number of users of Microsoft's überbrowser, [Internet Explorer](#) (IE), has begun to decline, in response to wider use of alternative browsers. One [statistical count](#) points to a current browser market share of IE at 84%, down from a high of 94%.

The decline in IE use does not just derive from the availability and attractiveness of new browsers but perhaps more from problems users have encountered in using the Windows version of IE. A new development since 2000, which users would gladly forego, is the increasing annoyance of non-requested and unwanted information and programs invading ones computer in the form of pop-up advertisements, automatically installed mini-programs commonly called spyware or adware, and installed scripts that redirect the browser to pornography or other undesirable sites. Microsoft has been slow to upgrade IE to deal with these problems, although the recent Windows XP [Service Pack 2](#) upgrade has added a number of security fixes to IE and Windows generally.

Firefox has features (like not supporting [activeX controls](#)) which make it [more secure](#) than Microsoft's browser (although it, too, has had security [issues](#)), and this has been a major ingredient in its popularity. But its attractiveness is also due to [features](#) not found in IE such as integrated newsfeeds, advanced tabbed browsing, and customizable contextual menus. One of the more popular features of Firefox is its flexibility; many [extensions](#) are available to expand its functionality, such as controlling iTunes playback, playing card games, or integration of a variety of dictionaries. Firefox has also garnered publicity due to its origins. It was begun by teenage prodigy [Blake Ross](#), who started writing code for the open-source [Mozilla](#) project at age 14. He and others felt that Mozilla (and Netscape) were suffering from feature bloat and started a new development path that featured a stripped-down and faster, yet expandable, browser. The first release of Firefox came in the Fall, 2004, with updates continuing to be released on a regular basis. Microsoft has [responded](#) to the sudden popularity of Firefox. Originally IE was not scheduled to be

upgraded until the release of the next version of Windows, code-named [Longhorn](#), scheduled for release in 2006, but recently Microsoft [announced](#) that it will release [IE 7](#) before Longhorn. If it is indeed the case that significant numbers of users are viewing Web pages in non-IE browsers, this presents an interesting new situation for Web authors, many of whom have been coding exclusively for optimal display in IE.

Standards and Compatibility

One of the [complaints](#) about Microsoft's approach in IE is the lack of support for Internet standards (as propagated by the [W3C](#)) and its insistence on the use of proprietary implementations of others. Firefox and other alternative browsers generally have better support for standards, such as [CSS 2](#) (cascading style sheets, level 2), which is notoriously quirky in IE. However, since IE has been the dominant browser by far, many Web sites have been designed to display well in IE, standards or no standards. This has raised a hotly debated issue in the Web browser community, over whether the alternative browsers should support official Web standards only or what David Hyatt of Apple [calls](#) "real-world standards," that is, the de facto standards imposed by IE. Hyatt argues that one should not have a purist approach to standards, but take into consideration how the vast majority of Web sites on the Internet are coded, namely using some of Microsoft's proprietary tags or non-standard implementation of CSS. This is how Hyatt explains, for example, Apple's decision to support the proprietary Microsoft "[contenteditable](#)" property (which allows in-browser editing of text). Understandably, Microsoft also [argues](#) for "real-world" standards (i.e., what is in IE) versus official standards.

Both Apple and Microsoft have been criticized over their plans to include proprietary approaches to Web development in their operating system updates. Apple's recently released 10.4 version of Mac OS X (code-named Tiger) includes Safari 2.0, which supports new proprietary HTML tags. The most attention has been directed at "[Dashboard](#)," which allows creation of mini-applications (e.g., Sticky Notes, Calendar, Calculator, Airline Scheduler). Dashboard "widgets" are written in HTML, Javascript, and CSS and can be viewed on the desktop. To accomplish this, Apple has introduced new HTML tags, including "canvas" (for graphics display) and "slider"/"search" (for forms). Meanwhile, Microsoft is endorsing an implementation of XML called [XAML](#) (eXtensible Application Markup Language), which will be incorporated into Longhorn, along with the new graphics system called [Avalon](#). Microsoft is touting XAML/Avalon as a way to create more sophisticated Web applications, but it is also designed to create stand-alone Windows applications. In fact, Longhorn is designed (as is, to some extent, Mac OS 10.4) to blur the distinction between Web and desktop applications. At least initially, both Apple's Dashboard and Microsoft's XAML apps will only run in their respective operating systems, not a hopeful direction for those favoring a universal approach to Web authoring.

One of the interesting issues related to browser development has been the use of XML. In early 2000, [XHTML](#) had just been released by the W3C as an XML successor to HTML, while [HTML 4](#) was slated to fade away from existence. The vast majority of pages today, however, are still coded in ordinary (and often badly written) HTML, not in valid XHTML (which mandates a "doctype" declaration, end tags, lower case). Browser makers have seen themselves obliged to be backwards compatible enough to display existing Web pages. In fact, Microsoft has used this situation as a [reason](#) for not having updated their browser to be more standards-compliant. Many Web developers have wanted Microsoft to improve style sheet support and its implementation of the document object model ([DOM](#)), the way in which objects on the page such as text or images can be accessed and changed through scripting. Microsoft has responded that changed and improved implementations of CSS and DOM in IE would break many existing Web sites. It has been [pointed out](#), however, that Microsoft could improve standards support for pages having a valid "doctype" declarations while maintaining backwards compatibility with the 80 to 90% of Web pages that are not valid XHTML. This "[doctype switching](#)" (first introduced in IE for Mac 5.5) allows Web developers to take full advantage of advanced Web design standards while allowing plain and quirky

HTML to be continued to be used. This is, in fact, the case for alternative browsers such as Opera, Safari and Firefox. The recent beta of Netscape 8.0 (for Windows) goes even further in striving for compatibility. It is based on Firefox, which uses the "Gecko" rendering system created by Mozilla, but it also includes IE mode which displays the page as it would appear in Microsoft's browser.

In 2000, a draft specification had just been released for a new way to use forms on the Web, [Xforms](#), a re-writing of form tags in XML (to be used in XHTML). Today, however, Xforms is not supported in current browsers and there has been intense debate about whether it should be implemented as written. An alternative approach called [Web Forms 2.0](#), has been endorsed by a group called [WhatWG](#) (Web Hypertext Application Technology Working Group) which is comprised of representatives of browser makers (except Microsoft). This approach is designed to be backwards compatible with existing forms, which Xforms is not. It also relies on field validation (whether a user has entered a legitimate value in a form field) through Javascript (as in existing browsers) rather than through checking for XML declarations. For the time being, the W3C is [staying](#) with Xforms, but it is too early to know which approach will be implemented in browsers. There is much riding on this development, as forms are vitally important on the Web, supporting as they do all kinds of transactions and interactions.

The reservations over Xforms signals a general unease with proceeding to the vision embraced in 2000 by the W3C of a Web based entirely on XML. Some argue that it is in fact the ease of creating HTML and its forgiving nature that has led to the Web's explosive popularity and that XML threatens the populist and democratic essence of the Web. In fact, there have even been calls for an extension of life for HTML, possibly even a new [HTML 5](#). This indicates an unwillingness to move away from the error-forgiving nature of HTML to a standard, rule-based XML with its insistence on well-formed code. In fact, Apple's "Dashboard" widgets, implemented in the new OS 10.4, use HTML, not XML, in large part, [according to Apple](#), to allow non-technical folks to create widgets. On the other hand, one XML implementation just getting started in 2000, has become a run-away success. [RSS](#) or "Rich Site Summary" is a way to create XML descriptions of Web content which become subscribable "channels" in RSS readers or in browsers with RSS support (such as Firefox or Safari 2.0). Another implementation of XML, [SVG](#) (Scalable Vector Graphics) has also garnered considerable support, although it is not yet universally supported in browsers. On the whole, however, XML has unfortunately not proved to be the Web technology that leads to universal standards in all browsers and platforms.

Ajax and Web Applications

One of the more promising uses of XML is the ability to connect Web pages to back-end databases and applications. While Microsoft's announced approach of using XML for Web applications will be restricted to Windows user running Microsoft software, there has been intense interest recent in a different approach which is cross-platform and cross-browser. This has been named [Ajax](#) by Jesse Garrett (of [adaptive path](#)), which stands for Asynchronous Javascript + XML. It describes a way to write dynamic Web pages that interact in the background with resources located on a server. This has been possible for a while now using an approach usually called "[Remote Scripting](#)," mostly using a Javascript library written by Brent Ashley called [JSRS](#) (Javascript Remote Scripting) or [Microsoft's approach](#), which uses [ASP](#) (Active Server Pages) and Java.

What is new in the Ajax approach is the addition of XML, pages being encoded in XHTML and data exchanged through XML formatted files. The core of Ajax, however, is something with the ungainly name of [XMLHttpRequest](#). This is a function first created by Microsoft in IE 5 for Windows as an ActiveX object. It was subsequently [implemented](#) as a Javascript object in Mozilla/Firefox and Safari. Similar functionality to XMLHttpRequest is in the proposed "Load and Save" DOM [specification](#) from W3C, which means this functionality is not likely to go away anytime soon. What it does is to allow retrieval of XML-formatted data in the background, which is converted to HTML objects that can be displayed through CSS and the DOM to the reader, updating information on the page without the need for

reloading. This is different from the traditional way in which server resources are retrieved, namely through returning an entire HTML page. To get an idea of how this works, one can take a look at how the zoom feature works in [Google Maps](#). Scrolling around within a map happens almost instantly because the data has been pre-loaded from the server and is waiting and available to be displayed to the user.

Google in fact has been making extensive use of Ajax to create Web applications such as [Gmail](#) (email service), Google [Suggest](#) (suggest as you type search function), [Orkut](#) (social networking service), and Google [Groups](#). It is also used in [Flickr](#) (photo sharing service), Amazon's [A9.com](#) (search service) and in the queue function in [Netflix](#) (DVD rental service). Apple has an interesting [demo](#), using XML data from iTunes RSS feeds. The reason for the strong developer interest in this approach is that it uses non-proprietary technologies that are well established and is fully cross-platform and cross-browser. Developers have been searching for some time for a good way to maintain a connection to a server which would allow constant, but hidden, updating of content. This has been a challenge given the stateless nature of HTTP server requests (i.e., no connection "state" maintained between server and browser). Awkward solutions such as hidden frames and faceless Java applets have been used, but Ajax works much more smoothly. It also seems to scale well, so that it can accommodate large and small Web applications. Obviously, it is most appropriate in contexts where there is XML data residing on a server (or data that can be transformed into XML) which a user might want to exchange for information already loaded on a Web page. Clearly there are compelling opportunities in Ajax for language learning applications. One could envision, for example, text annotations or other comprehension aids pre-loaded on a page and ready for when they might be requested by the user. Another example is the construction of vocabulary flashcards (Chinese-English [example](#)) or [quizzes](#) from a list of items kept in an XML file. [Examples](#) of other uses of Ajax are likely to suggest many more possibilities for language learning.

RESOURCE LIST

Web Browsers and Trends

- [OneStat.com](#) browser usage statistics
- [Once upon a time](#) 10 Year Anniversary of Yahoo, Netscape (Newsweek)
- [Yahoo 10th Anniversary](#)
- [WebSideStory: Firefox Gains Beginning to Slow](#) browser trends and statistics
- [Mozilla Chairman Mitch Kapor Speaks Out](#) discussion of browser competition
- [Browser battle shakes Net apps](#) concerning different standards for new Web forms
- [Rise of the Underdog Browser](#)
- [How Firefox works](#) from Howstuffworks
- [Firefox Extensions](#) from Mozilla
- [Browser Wars: Who's Winning, Who's Losing](#)
- [Netscape 8](#) information from Slashdot
- [Review of Netscape 8](#) from blogspot.com
- [IE7](#) From IEBlog
- [The Gillmor Gang](#) conversations about Web browsers (audio)

Standards and Compatibility

- [Surfin' Safari](#) Safari news and updates -- blog by David Hyatt
- [Web Hypertext Application Technology](#) unofficial group advocating a different new Web forms standard
- [The Web Standards Project](#) project promoting adoption of Web standards
- [Extending HTML](#) about Apple's new tags
- [Surfin' Safari](#) cComments from David Hyatt on Apple's proprietary tags
- [Dashboard](#) information about new feature in Mac OS 10.4 (Tiger)

- [Dashboard: Apple's ActiveX?](#) by Todd Dominey
- [Dashboard: Benevolent Pragmatism](#) about Safari's new proprietary Dashboard feature
- [IE and Standards](#) from Chris Wilson, one of the IE developers at Microsoft
- [Position is Everything](#) site dedicated to dealing with CSS bugs in browsers
- [A Standards-based Look at XAML's Features](#) from devX.com
- [Doctype switching and standards compliance: An overview](#) good explanation of standard versus "quirks" mode in browsers

Ajax and Web Applications

- [Remote Scripting](#) good overview
- [JSRS Select Box Filling Demo](#) example of remote scripting
- [Remote Scripting with IFRAME](#) from Apple Developer
- [CSS3](#) good introduction with examples
- [State zip demo](#) example of remote scripting using iframe
- [Guide to using XMLHttpRequest](#) tutorial by Bill Bercik
- [Ajax: A New Approach to Web Applications](#) by Jesse James Garrett
- [Dynamic HTML and XML: The XMLHttpRequest Object](#) from the Apple Developer Connection
- [Map of Switzerland](#) using Ajax
- [Weblications](#) about using Ajax to build Web applications