January 2004, Volume 8, Number 1
pp. 8-12

EMERGING TECHNOLOGIES
Making the Web Dynamic: DOM and DAV

Robert Godwin-Jones
Virginia Comonwealth University

Five years ago, in the January, 1998, issue of LLT, I wrote a column on Dynamic Web Page Creation, discussing options for Web interactivity and bemoaning incompatibilities among browsers. In the current column we will explore what has changed since 1998, new options that have arrived, and where we are with standards implementation.

Scripting Transformations: CSS and the DOM

Five years ago, "Cascading Style Sheets" (CSS) were just beginning to be used in designing Web pages; the specifications for CSS 1 were at that point about a year old. Since then, CSS Level 2 (May, 1998) is an approved recommendation (by the W3C), and CSS Level 3 modules have been issued as working drafts or candidate recommendations. More significant than the W3C activity is the fact that CSS has become the most widely used method for formatting Web pages. This development has only been possible because of CSS support in Web browsers. Beginning with the 5th generation browsers (Internet Explorer 5, Netscape 6, Opera 5), support for CSS 1 has been sufficiently robust and consistent to encourage developers of HTML authoring tools to incorporate CSS support. A helpful development that has served to encourage more wide-spread deployment of CSS has been the push to provide more accessible Web pages. Accessibility to users with special needs is much easier to code in a consistent and machine-readable fashion using CSS than in traditional HTML formatting, often built around the use of tables for formatting.

CSS use is an important step towards creation of a semantic Web, that is, Web pages which have content clearly separated from formatting and thereby ease document search and retrieval. The use of styles to format Web content brings another benefit: It allows on-the-fly transformation of content displayed on a page. This provides a powerful means to change instantly the "look and feel" of Web pages. It is a fairly straightforward process, for example, to allow the viewers of this page, assuming they are using 5th generation browsers or later, to change the font style (to Palatino or Comic Sans MS) or font size (to 16 or 10) [script from Apple Developer page]. With CSS this is done immediately, without the necessity of reloading the page or fetching fresh content from the server. As the Web becomes the central means for delivering all kinds of information, it makes sense to accommodate user preferences as is done in traditional software programs.

The real magic of CSS is evident only when combined with scripting access to the properties of a page's DOM or "Document Object Model." The DOM provides a standard way to represent and access all the items on a Web page, conceptualized in a hierarchical fashion, like a tree with branches. The tree describes the entire HTML document, with each item being a branch representing an HTML tag or a textual entry inside a tag. The relationships of the items to each other (parent, child, or sibling) are represented in the tree. A "node" of a page's DOM tree is an object (such as an HTML tag like <p> or <body> or a string of text) that has a set of properties that can be accessed and changed. DOM Level 2 became a W3C recommendation only a year ago (January, 2003) but has been around as working draft since 2000. DOM Level 3 is under development. The acceptance of the DOM and its support in current browsers goes quite a ways in solving the lack of standards in DHTML ("dynamic HTML") discussed 5 years ago.

The problem of DHTML in 4th generation browsers was not just incompatibility, but also limitations on access to Web page contents. The ubiquitous image swapping (image changes as cursor passes over it) was made possible because images were one element which could be manipulated. But that did not hold

Copyright © 2004, ISSN 1094-3501

true for all parts of a page. While under DHTML individual objects -- but not all of them -- could be accessed and transformed, under DOM the whole document is accessible. Moreover, under DHTML each tag object had a different set of properties. Under DOM, there is a standard set of properties for each of the two types of nodes (tags or text), as well as a standard way to access any mode (most commonly by using "getElementByID"). Nodes can be created, changed or deleted on the fly. That means an individual string of text like "**I think, therefore I am**." can be instantly changed to French, German, Spanish, back to English or to anything else we want. Nodes can also be hidden and then displayed in consequence of a user action. This provides some interesting annotation options for language texts or for allowing a larger document to be displayed with less screen real estate, such as in this lesson plan creator. In fact, the potential for using the DOM to create dynamic mini-applications for language learning is enormous.

Remote Scripting and Web Services

Manipulating the DOM provides one means of overcoming the statelessness of the Web. A user retrieving a Web page does not have any long-term connection to the Web site hosting that document, but rather simply contacts the site for a fraction of a second to request transmission of a Web page, along with any accompanying images, multimedia, scripts or style sheets. Identifying information about the user and the nature of the connection to a particular Web server might be saved in a "cookie," but this information is stored locally. To retrieve new information from an already loaded Web page, the user must make a new connection to the Web host requesting a refreshed copy of the document. As Web pages become applications in their own right, this single page retrieval system becomes awkward, cumbersome, and severely limiting. The DOM allows re-display of content without re-loading a page but it does not update content from the server, but rather just transforms the content already loaded.

There are now, however, methods available for seamlessly updating information on Web pages. "Remote scripting" refers to a process of updating information on a page with data retrieved from a server (often from a database) without the necessity of reloading the page. The process is transparent to the user, who is likely unaware of (and uninterested in) where and how the information being viewed is stored and retrieved. Remote scripting is a form of "Remote Procedure Call" (RPC), a reference to long existing processes for passing data between remote computer systems. One means to achieve this goal is through the use of Java applets or ActiveX controls. Either can query a server and retrieve information, which can be passed through JavaScript variables to a Web page's memory, then dynamically updated through the DOM. Ironically, the use of this approach, recently enabled in 1998, is problematic today because of lack of browser support (in all but Microsoft Windows browsers) of the underlying technology, usually called LiveConnect, and originally developed by Netscape.

A more universally deployable technique for achieving this goal is to use a non-displaying browser container such as an "iframe" (first developed by Microsoft) or a hidden layer element. An example (in French) of this approach is a coloring map of France, which combines locally loaded information with data retrieved from a server. It is easy to understand how this technology could be used to retrieve and display dictionary or encyclopedia entries to assist in a student's understanding of an online text. This is described in a presentation by Michael Leventhal. An on-line version of a German fairy tale uses this approach to display either a local gloss or, if no gloss exists for the search term, to fetch a dictionary definition from a remote server. To the user, it is not immediately apparent (or of importance) whether the gloss is stored locally through JavaScript in the browser memory or retrieved from a hard drive on a Web server in Germany.

One of the major developments on the Web since 1998 is the advent of XML. Increasingly XML is being used to encode documents of all types but especially in situations in which the data may need to be accessed by a variety of clients and displayed in multiple ways. Data encoded in XML is readable by both humans and machines and can be easily transformed into HTML, plain text, or other formats. XML is used today as a lingua franca between different computer applications. The import/export functions in

Blackboard, for example, write all the data in XML. Steve Cushion (London Metropolitan University) has demonstrated how to use XML to develop a common format among quiz/exercise building software programs such as "Hot Potatoes" or "Welts." This allows the potential of creating a large pool of exercises for practice or assessment that is independent of any delivery system.

The exchange of XML-encoded data between remote computer systems is often refereed to with the term "Web services," although this could also involve mining traditional database systems. This is an area of immense interest currently, especially in the area of business-to-business transactions. But the universality and transparency of this approach make it of more general interest and potential, especially since more and more information of interest to language teachers and learners is likely to be stored in XML format. Major Web services/companies such as Google, Amazon, or eBay use XML encoding, but so too do publishers and media centers worldwide. A project under development at the University of Limerick (E-RAM for "Electronic Resources for task design using Authentic Materials") by Freda Mishan uses XML to create interactive teacher resources with a range of authentic language materials such as newspapers, advertisements, broadcasts, and other Web documents.

One of the first protocols developed to enable Web services was XML:RPC by Dave Winer. He is also one of the architects of a new protocol, SOAP ("Simple Object Access Protocol"), which has gained wide acceptance among Web developers. It provides a system for passing structured (i.e., XML) data between computer systems. As such it is a server-side technology. However, the Netscape (and Mozilla) browsers have built-in client support for SOAP services, and other browsers will likely follow. Netscape uses a client-side implementation in a JavaScript interface which can create, read, and receive SOAP messages. Creating a SOAP message follows the same process as creating any JavaScript object. An example of using this interface is to provide a more flexible way to search and display search results from a site such as Google. The results are retrieved behind the scenes and through the DOM and JavaScript can be integrated into a Web page in a variety of ways.

DAV and the Collaborative Web

A different approach to interfacing with servers is represented by the technology called WebDAV or simply DAV ("Web-Based Distributed Authoring and Versioning"). DAV is an open standard (RFC 2518 from the W3C) that supplements the traditional http protocols of "GET" (to retrieve a Web page) and "PUT" (to send data to a CGI script) with a variety of additional actions (DELETE, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK). It is integrated into current versions of the most popular Web server, Apache (through mod_dav). In a sense, DAV is like a new and improved FTP service. It enables viewing and updating of files and folders stored on a Web server. If you are using WindowXP or MacOS X, DAV is built into your operating system. In both systems, folders on remote servers can be mounted on the desktop and manipulated as if they were folders on the local hard drive. This assumes you are a registered user of that system with the requisite username and password.

DAV is ideal for collaborative work, being fully cross-platform and application independent. It allows for groups of users to access and edit the same files, with mechanisms providing security and version control. One of these features is file locking, which prevents group members from overwriting changes made by each other. When several people are working on the same file, DAV requires that the files be compared or merged before saving. By keeping track of changes and using metadata to record other information about a file, DAV aids in identifying the contents of files. The property values stored with documents also enable more efficient searching.

DAV is built into the current version of the learning management system WebCT, allowing users of that system complete and easy access to course content. It is also supported in recent versions of Microsoft Office as well as by Web authoring tools such as Macromedia's Dreamweaver and Adobe's GoLive. Several open source content management projects such as Zope are built around DAV. It is also being

used a means of sharing calendars, currently enabled among users of the Mozilla calendaring system or Apple's iCal.

Outlook for Language Learning

As the Web has grown in volume and use over the past 5 years, Web pages have become more sophisticated and complex. While there are still lots of plain Jane sites, the trend is clearly in the direction of graphically attractive pages with dynamic elements such as layered menus or at least image swapping in buttons. Often Web design is separated form content creation, a process facilitated by the use of CSS. Much of the coding necessary to enable popular uses of DOM interactivity is automatically created by authoring tools. This enables a low threshold access to creating fairly professional looking and functioning Web sites.

This is as available to language teachers as it is to anyone else. While this is a positive development, it doesn't necessarily lead to the creation of better Web sites for language learning. Of course, many language teachers today are using a learning management system, so that any Web pages created for teaching may be generated automatically in the forms and formats supplied by the system. In any case it is likely that fewer language professionals are exploring new technologies such as the DOM or Web services to investigate potential benefits to language learning. Yet the technologies are too powerful not to offer immense potential. This is the case as well for DAV, which provides an easy-to-use content management tool. Management of digital content will become of increasing importance as language teachers accumulate more materials and documents in digital formats. The Web has become a more dynamic -- and more standards-based -- environment since 1998 and one which could yield significant new creative opportunities for Web-assisted language learning.

Resource List

CSS and DOM

- Robin Cover's DOM pages good overview
- Hide/Show Layer from Apple Developer site
- Dynamic Content with DOM-2
- The Semantic Web is closer than you think from O'Reilly
- Taylor's DHTML Tutorial from WebMonkey

Remote Scripting and Web Services

- Web Services from O'Reilly
- Remote Scripting with IFRAME from Apple Developer site
- The Iframes Lowdown from WebMonkey
- A Gentle Introduction to SOAP by Sam Ruby
- A Busy Developer's Guide to SOAP Dave Winer and Jake Savin
- XML-RPC for Newbies by Dave Winer
- Remote Scripting Resources from Brent Ashley
- Coloring Example using remote scripting (in French)
- Get State data example using Remote Scripting
- MSRS Microsoft remote scripting
- IDIOMA-TIC XML-based language learning authoring tool (in French)
- Increasing accessibility by pooling digital resources using XML to convert between authoring tools (by Steve Cushion)
- E-RAM - an electronic resource for language learning task design using authentic materials uses XML, by Freda Mishan

DAV

- DAV FAQ
- Zope open source content management system using WebDav
- WebDAV Calendar Sharing
- Collaborative Authoring on the Web: Introducing WebDAV by E. James Whitehead
- WebDAV protocol comes of age from InfoWorld