Nifty Technology and Nonconformance: The Web in Crisis

Nate Zelnick, Allaire Corp.

The near parity between Microsoft's Internet Explorer and Netscape's Navigator is bringing what had been merely a developers' annoyance firmly into the realm of disaster.

The problem is a relatively familiar one: Nonconformance or incomplete implementation of standards makes building sophisticated Web pages that work across browsers a difficult and expensive job. Because Web developers have to spend extra time and effort working around quirks in different implementations, they almost always have to privilege general functionality over technological enhancements.

Just trying to get pages to look the same across browsers by following the Cascading Style Sheet Level 1 (CSS1) standard—a W3C recommendation finished nearly two years ago—requires intimate knowledge of the peculiarities of each version of each browser on each platform. IE 4 on Windows, for example, supports about 80 percent of CSS1 and about the same amount on Macintosh, but not the same 80 percent. Navigator 4 supports a little less of CSS1 on both Windows and Macintosh, but again not the same subset and not in the same way.

And CSS is just the tip of the iceberg.

Editor: Charles Severance, Michigan State University, Department of Computer Science, 1338 Engineering Bldg., East Lansing, MI 48824; voice (517) 353-2268; fax (517) 355-7516; crs@egr.msu.edu; http://www.egr.msu.edu/~crs

It seemed for a short time that tragedy would be averted and the much heralded "browser war" would settle into détente.

DHTML DEBACLE

Both Netscape and Microsoft introduced the concept of Dynamic HTML in their 4.0 releases. But while DHTML generally meant the same thing to both companies, they differed dramatically in the details.

At the high level, DHTML is the combination of style sheets, scripting, and a document object model. These technologies together give developers the ability to control the location of elements on the screen and to manipulate objects through simple, declarative scripting.

Netscape based its DHTML initially on a set of proprietary extensions to HTML, JavaScript, and a proposed element—called Layer—that was soundly rejected by W3C. And the system that Navigator uses to associate style sheets with pages is JavaScript Accessible Stylesheets (JASS), another proposal rejected by W3C.

Microsoft, which had made its own forays into proprietary extensions of HTML, recanted somewhat in the IE 4 release and based its DHTML mostly on W3C standards. IE 4 handles presentation through CSS, document elements through an implementation of the Document Object Model (DOM), and, as in IE 3, scripting through the generic Internet Explorer Scripting Host. The problem with IE 4 is that it doesn't implement all the W3C specifications completely. What's more, it piled a whole set of IE-only technologies on top. While these technologies are interesting, they are tangential to the needs of developers and end users.

Netscape and Microsoft both agreed to support W3C standards when it became clear that their differing approaches were going to get them nowhere. Developers couldn't use the nifty new features without sacrificing a significant percentage of the potential audience for their pages.

It seemed for a short time that the tragedy of the Web being broken into separate, parallel worlds would be averted and that the much heralded "browser war" would settle into détente. But now that the 5.0 browsers are approaching release, it's becoming clearer that neither company is going to live up to its pledge.

STANDARD EFFECTS

The standards issue may seem arcane—important only to those who like to read IETF RFCs and W3C submissions—but standards matter to everyone who uses the Web, builds applications, or believes that universal access to information via the Internet is valuable.

Users

In your relationship to the Web as an information consumer, you confront the problem as the frustration of reaching a page that doesn't render properly in one or another browser. Maybe you're confronted with a banner that proclaims the site requires either Internet Explorer or Navigator. If you are using the wrong browser, then you have to decide whether the information you're trying to get is worth a 20-Mbyte download and a new browser installation.

Developers

If you're a developer building Web applications, the problem is much, much worse. With the installed Internet Explorer and Navigator bases being nearly equivalent, you are now faced with the decision to build a site that either excludes 50 percent of your potential audience or is less functional than it could be.

> Developers would love to use all the nifty technologies that Microsoft and Netscape introduce. But because they spend all of their time working around what should be there but isn't, they can't.

Because building a one-size-fits-all Web site means using older versions of HTML, past workarounds continue to be necessary, including using tables to do positioning instead of the more straightforward CSS positioning technique that was codeveloped by Netscape and Microsoft more than a year ago (and which the companies have yet to adopt completely). Since the implementation of HTML tables is slightly different in each browser—sometimes varying between point releases of the same browser—getting the look you want requires a lot of tinkering, even with older versions of HTML.

Believers

Finally, if you view the Web as a new medium that changes the way people deal with information, then the browser standards debacle is nothing less than a giant step backward. By building pages for particular browsers, developers are either building documents with limited lifespans or ensuring that the hacks and workarounds used today will have to continue for the foreseeable future.

A WAY OUT

When it became clear that the commercial interests and the general health of the Web were in conflict, the W3C sought to find a way to help move the Web forward without breaking it. The solution was to separate aspects of a document into clearly distinct layers.

By doing so, developers can delineate the relationship between the markup language used to define a page's structure, the means to manipulate that structure, and the method used to present information. Because these layers are separate, they can be used much more efficiently and according to what needs to be done with the basic information.

Core standards

The standards that make these layers possible are HTML 4.0, XML, Cascading Style Sheets, and the Document Object Model (DOM). Adding a version of Netscape's JavaScript—standardized by the European Computer Manufacturers Association (ECMA) as ECMAScript—makes the list complete. These are the standards that Microsoft and Netscape each agreed to support more than a year ago. And these are the standards that neither browser supports completely and will not support completely in the 5.0 releases.

You can think about how these technologies work together in this way:

- HTML defines what is what on a page.
- CSS describes how it's rendered.
- DOM defines how to change elements via scripting or some other programmatic method.

The advantage of this approach is profound.

Think about it this way: When PCs first hit the market, the most popular word processor was WordStar. When WordPerfect became the de facto standard, the most common document format changed. It changed again when Word overtook WordPerfect as the dominant word processor. But does Word know how to read WordStar documents? What if you need to read a document created in 1982 and stored in WordStar, but don't have the program? A painful process of stripping the binary formatting out of ASCII awaits you.

But what if that document was written in a general descriptive language like HTML or, even better, XML? Then all you'd need to know is how to parse the elements to understand the general structure of the document.

Endless possibilities

Since markup languages are a well-understood, standard, and predictable declarative format, the document can tell you what to do with it. Since the actual presentation of the document is described by a style sheet that exists as a separate file, it can be swapped out for a more appropriate presentation designed for a different kind of device—maybe a television screen or a PDA—without having to change anything in the document itself. In fact, once the information in a document is created and structured, you'll never have to touch it again unless you're making changes to its actual content.

And because the document exposes its structured elements in a standard and predictable manner via DOM, different applications can manipulate its structure automatically. While the immediate benefit is Web pages that can do more interesting things in a more predictable manner, the possibilities are much more interesting. The most obvious benefit is better search engines that don't waste time, cycles, and storage trying to catalog Font and Blink elements, empty GIF files, and extra table information.

But beyond that there are whole sets of applications that become possible when machines can read the structure of a document automatically and extract information in a useful manner without human intervention. Getting there, however, is not going to be easy.

Web Standards Project

The problem is a classic Catch-22 situation. Until the browsers fully support the basic standards, developers will be forced to create workarounds. As long as

Continued on page 119

Continued from page 116

developers are creating workarounds, the browser makers will feel no pressure to step up to the task of actually implementing everything they've promised.

To get out of this trap, a group of Web design professionals and other interested people formed the Web Standards Project. WSP (pronounced "wasp" and implying a sting) is designed to do what a consortium like the W3C—as a neutral meeting place for competitive companies—cannot: Make it clear that not having these basic standards supported in browsers is holding back the potential of the Web.

WSP estimates that about 25 percent of the Web development process is spent dealing with compatibility issues in the most basic standards. It is unfortunate that developers have to choose between reaching a broad audience or applying interesting technology, because they will almost always have to favor a broad audience, which slows the rate of innovation on the Web and robs it of its ultimate potential.

Innovation in browsers is great. Developers would love to use all the nifty technologies that Microsoft and Netscape introduce. But because they spend all of their time working around what should be there but isn't, they can't.

A Pyrrhic victory is one in which nobody wins. The expression derives from early Roman history wherein Pyrrhus, king of Epirus, beat the Romans on the battlefield but destroyed his kingdom in the effort. A browser war that is fought on the basis of core standards can do nothing but destroy the Web along the way.

Add your voice to the WSP (http://www.webstandards.org) so we can move on to real work. It's in everybody's best interest to make this issue disappear. ❖

Nate Zelnick is project manager for industry relations at Allaire Corp. and represents the company at W3C. He also writes the biweekly columns "Under Development" for Internet World *and "The XML Files" for the Web magazine* WebDeveloper.com. *Contact him at nzelnick@allaire.com*.