Language Learning & Technology
Vol. 5, No. 1, January 2001, pp. 11-19

EMERGING TECHNOLOGIES

Accessibility and Web Design
Why Does It Matter?

Bob Godwin-Jones
Virginia Commonwealth University

The passage of the American with Disabilities Act (ADA) in 1990 has had a major impact on the physical infrastructure of college campuses in the US. Less well-known is the growing ADA impact on educational applications of technology, particularly in the use of the Web. Parallel to the beginnings of ADA-inspired awareness of this issue has come the approval by the Federal Office of Management and Budget on December 21, 2000, of Section 508 (of the Rehabilitation Act) accessibility standards. Government agencies will have six months from this date to make their Web sites accessible to users with disabilities. After that date, federal agencies -- and presumably non-compliant institutions receiving federal funds -- will be subject to law suits. Many regions in the US have instituted accessibility guidelines as well.

This is not just a US issue; there are a number of countries, including Australia, Canada, Denmark, France, and Japan, which have recently issued policies relating to Web accessibility. But legality aside, it makes good sense to make instructional technology as accessible as possible. There are emerging standards for achieving this goal. Interestingly, new Web delivery options can benefit from following the same guidelines.

Why should language teachers be concerned with accessibility? Can't the small number of users with special needs be accommodated individually? In response, one could ask, why worry about the limited number of Internet users who don't know English? Just as we should be concerned with monolingualism on the Web and the social-economic "digital divide," so too should we be concerned about problematic access to learning materials on the part of even a small numbers of students. In fact, the number is probably larger than one might assume. In the US there are different estimates on the number of students with disabilities -- one recent analysis gives a count of 8%. Using the classroom retrofitting analogy, it's not a question of building new, separate classrooms, but of changing classroom configurations to accommodate all users. It turns out that the main-streaming approach ends up benefiting all users.

The Web Accessibility Initiative

What makes Web pages inaccessible? That all depends on the nature of the disability. Visually impaired users might need a much larger font, or a sharp contrast between background and foreground color. Color-blind users need to have color-transmitted information translated into distinguishable shades of gray or delineated in some other way. Blind users may be accessing Web pages using a screen reader, which uses speech synthesis to read the pages and may be confused by improperly coded pages. Physically impaired users might have difficulty in typing key combinations. Other users might need to navigate with a non-traditional input device.

The W3C, the standards-setting body for the World Wide Web, has addressed these issues through its Web Accessibility Initiative (WAI), which issued a set of Web Content Accessibility Guidelines (version 1.0) in May, 1999. They were followed in 2000 by WAI guidelines for user agents and authoring tools. The WAI Content Guidelines include a list of checkpoints for evaluating Web pages for their degree of accessibility to people with physical, visual, hearing and cognitive/neurological disabilities. Each checkpoint is assigned one of three priority levels. Priority One are checkpoints which must be met to prevent lack of access for some groups of users. Compliance to Priority One checkpoints is known as "Single-A" conformance. Priority Two ("Double-A" conformance) are checkpoints which should be met to prevent difficulties in access for some users, and Priority Three ("Triple-A" conformance) are checkpoints which authors may satisfy to ensure good access for all users. The fact that there are three ordered levels of conformance allows Web site developers to focus first on eliminating the most serious barriers to accessibility. The WAI Guidelines correspond closely to those for conformance with Section 508.

Best Practices for Accessibility

The list below discusses items drawn principally from Priority One checkpoints. One of the side benefits of following the practices suggested here is that the pages will also be better designed for access by users of palmtop computers (such as Palm or Windows PocketPC) or wireless devices (such as a WAP -- "Wireless Application Protocol" -- enabled cell phones). Code examples for most of the issues discussed here are given at the WebAIM site, an excellent resource for Web accessibility issues.

FRAMES

> Frames should be used cautiously, since they do not display well in text-only environments. When used, frames should always be titled (in the frameset document). The most popular use of frames is to provide a table of contents in one frame while displaying the content in another. A possible alternative (or addition) is to include at the bottom of each page (perhaps as a server-side include) main navigational links for the site. Alternatively, the "noframes" tag can be used to supply text or links for non-frame capable browsers.

TABLES

Tables are widely used for formatting of Web pages, not just for spreadsheet-like data tables. When read by a screen reader, cells in a table are read across by row, which may not reproduce information in the intended order. The W3C has created a tool (Tablin) for linearizing tables, so that the content is formatted in logical order for screen readers. It is helpful to include row and column headers ("<th>" tag), which supply general information on the cell contents. Large complex tables are problematic for wireless as well as low-bandwidth users since the entire table must be loaded before any text is displayed. When using tables, it is also helpful to include a summary of the table's content (as a "summary" or "longdesc" tag).

IMAGES

Images should always include an "alt" tag to supply a title or information for users who are not able to see the images (or choose to turn off automatic image downloading). If transparent gifs are used for spacing purposes (not recommended), the alt tag should just contain a space (i.e., alt= " "). It is not recommended (although widely practiced) to include text in images (for example, as buttons) since text enlarging will not also enlarge text in graphics. Overuse of images increases page load time for all users. Imagemaps should include alternative texts for hot spots. Client-side imagemaps are preferred to server-side since information on target urls is embedded in the sent file and therefore available for the browser to transmit to the reader.

SCRIPTS AND APPLETS

Generally server-based scripts (i.e., CGI) do not pose in themselves access problems (with the exception of server-side image maps mentioned above). However, client-side scripts (i.e., JavaScript) do pose difficulties since JavaScript is not supported on most alternative browsers. A page generated dynamically through JavaScript (with "document.write()"), for example, will not display at all in such a browser. Whenever possible, text, CGI, or other equivalents to, or explanations of, the JavaScript functionality should be supplied. The "noscript" tags can be used for this purpose. At a minimum files should be readable without JavaScript. Particular caution should be exercised in creating JavaScript links ("href='javascript:my_function()'"). If possible a CGI "fall-through" should be supplied, which duplicates the JavaScript functionality on the server. Java applets can be made more accessible through the use of Sun's Java Accessibility API. Alternative text descriptions can also be supplied.

MULTIMEDIA

The WAI advocates a redundant approach to multimedia use, that is, supplying the same content in several different formats. This means, whenever possible, supplying text transcripts or captions for audio and video files, or, if that is not possible, providing descriptions. Of course, for language learning, it might be desirable to have transcripts, partial or complete, available as an option to learners. The SMIL language ("Synchronized Multimedia Integration Language") provides a means for synchronizing text or captions with streaming media.

STRUCTURE

It's best to use a clearly identifiable and consistent structure in creating Web pages, including the proper use of header tags ("<H1>") to designate content structure. Use of header tags (rather than <font> tags to simulate headers) makes it easier for alternative browsers to parse correctly the page's structure. Long lists should be broken up and if possible grouped together and appropriately tagged. It is desirable to allow a means of jumping over lists of links and going directly to the main page content (for example, by using the "name" tags).

FORMATTING

In general, page formatting should be fluid not fixed, thus allowing page display to adapt to different screen sizes. Relative rather than absolute values should be used for sizing page elements. The use of Cascading Style Sheets (CSS) is encouraged; however page display should also be acceptable without CSS. Web authoring tools such as FrontPage achieve exact placement on the page through use of a complex set of nested tables. A preferred way to place elements on the page is to use CSS.

META-DATA

It is helpful for all users, but especially for those using alternate browsers to provide as much "behind-the-scenes" information as possible. In fact, "front-loading" basic information is very helpful throughout Web pages, in order to give users an indication of what content is being provided. Meta-data is included in the page header and can include information in a variety of categories; most helpful is the use of a standard set of categories such as the "Dublin Core." If you anticipate your Web pages will be accessed by users of the popular AvantGo Web client (for Palms, WindowsCE or WAP-enabled cell phones), you should include the following tag:

```
        <META name="HandheldFriendly" content="True">
```

This tag enables several HTML features that are normally turned off in the AvantGo and similar browsers. One piece of information important to include is the language of the page, since this can influence how screen readers process the page. Changes in language on the page should be tagged as well.

VALID HTML

Main-stream browsers (Netscape, Internet Explorer) have traditionally been very tolerant of loosely written HTML, but other browsers are not so forgiving. Nothing can cause more havoc in screen readers that non-standard, incorrect or incomplete HTML. It's a good idea to have an HTML validator check your code. This works best if you have declared in the first line of your file what version of HTML you are using ; the DTD ("document type definition") for HTML 4.0 is as follows:

```
   <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
          "http://www.w3.org/TR/1998/REC-html40-
19980424/loose.dtd">
```

Information about DTDs and different versions of HTML is available from the W3C Web site. Documents created referencing published formal grammars make the work of all user agents in interpreting and rendering pages easier and more reliable.

STYLE

Avoid links such as "Click here" or "Follow this link." They are not only awkward stylistically, they do not provide much information when read as one of a series of links on the page by a screen reader. Also, it's helpful to give warning before linking to something that is not a standard Web page link (i.e., a sound file or PDF file). If a link opens a new window, some indication of that fact should be provided. Colors alone should not be used to convey semantic information (for example, whether a text is hot-linked).

Tools and Approaches

There are a number of tools available which can help in creating accessible Web pages or in retrofitting existing pages. One of the most widely used is Bobby, which provides a quick analysis of a given page's accessibility. Bobby analyzes Web pages using the WAI Guidelines and provides a detailed report on problems. It is frequently used by Web developers because it also does browser-specific checks and reports, detailing problems with page display in a variety of browsers on different operating systems. Another useful program is Tidy, which helps clean up HTML code, converts to different language sets, and even translates into basic XML ("Extensible Markup Language").

It is very helpful to preview Web pages in a variety of browsers, including Lynx. Another browser that is very useful in evaluating pages for compatibility is Opera. Opera provides excellent standards support and a variety of features for access by users with disabilities, including full keyboard navigation, a completely customizable interface, and a powerful zoom function. It also has one-click options for viewing pages with or without images and for toggling between document and user settings for fonts and colors. It is also possible, of course, to check pages with a browser specfically designed for accessibility.

XML, when widely adapted and fully supported on browsers, will provide a more effective means to deliver the same content formatted in different ways for different browsers. Some support is already available in current Internet Explorer 5 and Netscape 6 browsers. A lot of new browser development for other devices includes XML support. Even if XML is not supported on the client end, servers can be set up to transform content from XML documents (using XSL "Extensible Style Sheet" transformations) into different formats appropriate to different browsers.

On the other end of the spectrum from XML, another alternative for accessibility is to create text-only versions of all pages. This is recommended by the WAI only as a last resort, if retrofitting pages is not an option. Experience has shown that keeping a separate set of text-only pages synced with the full client pages takes more time and effort than most Web authors/developers are likely to want to expand over the long run. Of course, if Web pages are generated dynamically (i.e., from a database), this may be easier to achieve.

Resource List

Info

- Policies Relating to Web Accessibility links to information on this issue in a number of countries
- Translations of Web Content Accessibility Guidelines 1.0 from W3c - in a variety of languages
- Guidelines and Standards accessibility standards for regions and cities in US
- Federal IT Accessibility Initiative Section 508 compliance
- Accessibility of the Internet in Postsecondary Education: Meeting the Challenge white paper by Cyndi Rowland of the University of Utah

- Translations of WAI QuickTips starting points for accessibility retrofitting
- Microsoft Accesibility
- People with special needs Apple accessibility information
- IBM Accesibility Info
- Java Accessibility Univeristy of Wisconsin
- Web Accessibility Initiative (WAI)
- HTML Techniques for Web Content Accessibility Guidelines
- Core Techniques for Web Content Accessibility Guidelines
- CSS Techniques for Web Content Accessibility Guidelines
- Web Content Accessibility Guidelines
- Mobile Access: Working towards seamless Web access from mobile devices
- HTML 4.0 Guidelines for Mobile Access
- Fact Sheet for "Web Content Accessibility Guidelines 1.0"
- Alternative Web Browsing
- Techniques for Web Content Accessibility Guidelines 1.0
- Authoring Tool Accessibility Guidelines 1.0
- Accessible Web authoring resources and education center
- Applying the ADA to the Internet: A Web Accessibility Standard from the American Bar Association
- Selfish Reasons for Accessible Web Authoring
- Designing for differences
- Web Accessibility and Users with Disabilities
- The Instant Access Treasure Chest Foreign Language Teacher's Guide to Learning Disabilities (Sonja Moore)

Authoring Tips

- Colour Blind Design Hints and Tips
- Color Contrast and Partial Sight
- Accessibility Features of CSS
- Accessibility and Cascading Style Sheets
- Sun's Java Accessibility API
- The Adaptive Technology Resource Centre from the University of Toronto
- WebABLE

Tutorials/how-to

- Getting Started: Making a Web Site Accessible
- Curriculum for Web Content Accessibility Guidelines 1.0 includes concrete examples
- WAI Quick Tips Reference Card
- Checklist of Checkpoints for Web Content Accessibility Guidelines 1.0
- Web-Savvy: Inclusive Web Design
- WebAIM (Web Accessibility in Mind) The Web Accessibility "How-To" Site
- Tutorials on Accessibility from WebAIM
- Web Color Study (University of Toronto)

Tools

- Evaluation, Repair, and Transformation Tools for Web Content Accessibility
- LYnx Viwer
- Viewable with any browser
- W3C's HTML Validation Service
- CAST's Bobby Service

- Tidy
- Home Page Reader (IBM)
- Opera
- Jaws
- WAVE 2.01 tool for analyzing Web pages for accessibility
- Tablin HTML table linearizer
- WAI Report Tool Web Accessibility Report Tool from W3c

All links validated on December 20, 2000.

About LLT | **Subscribe** | **Information for Contributors** | **Masthead** | **Archives**