Designing Universally Accessible Web Resources for People with Disabilities

Jon Gunderson, Ph.D.
Division of Rehabilitation – Education Services
College of Applied Life Studies
University of Illinois at Urbana/Champaign
1207 S. Oak Street, Champaign, IL 61820
E-mail: jongund@uiuc.edu
Outline

**Introduction**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Myths of the Web</td>
<td>1</td>
</tr>
<tr>
<td>Digital Divide</td>
<td>2</td>
</tr>
</tbody>
</table>

**Alternative Views of the Web**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard Support</td>
<td>5</td>
</tr>
<tr>
<td>Access to Text Descriptions</td>
<td>7</td>
</tr>
<tr>
<td>User Styling of Text</td>
<td>13</td>
</tr>
<tr>
<td>Speech Browsing</td>
<td>16</td>
</tr>
</tbody>
</table>
Glossary

ADA: United States American with Disabilities Act of 1990 that guarantees the rights of people with disabilities to have access to public spaces, services and employment.

ATAG: W3C Authoring Tool Accessibility Guidelines provide authoring tool developers with information on how to support the creation of accessible web content and be more accessible to people with disabilities.

Text Equivalent: A text description associated with non-text content like images, audio and video.

ALT Text: The short description associated with an image on the web, of the form alt="brief description of the image".

CSS (Cascading Style Sheet): Cascading Style Sheets is a W3C technology designed to style HTML and XML content for rendering in a browser. The specifications began with CSS1 in December 1996, evolved to CSS2 in May 1998 and currently CSS3 is under development.

Disability: A visual, hearing, muscular, learning or mental impairment that substantially limits one or more of the major life activities of an individual.

HTML Structure: Using the structural markup capabilities of HTML to indicate the relationships between information in a web resource.

Keyboard Shortcuts: The ability to use keyboard commands to control software, providing an alternative to pointing with a mouse to select functions.

LONGLDESC: A URI to a detailed description of an image and is an attribute of the IMG element.

Section 508: United States Section 508 rules and regulations are designed for use by federal agencies to provide access to services by citizens and accommodations to federal employees with disabilities. In December of 2000 the Electronic and Information Technology Accessibility Standards included web accessibility requirements.

Screen Magnifier: A software program that magnifies text and graphics, and controls the colors on a graphical computer system

Screen Reader: A software program that is used by people who are blind to have the elements on a computer screen read to them through synthesized speech or refreshable Braille display.
UAAG: W3C User Agent Accessibility Guidelines 1.0 provide information to developers of browsers and multi-media players on how to make their technologies more accessible.

Universal Design: The design of resources to adapt to the needs and capabilities of a wide range of users, including people with disabilities.

WAI: W3C Web Accessibility Initiative is a program of the W3C to promote the accessibility of the web to people with disabilities through education, design guidelines and review of web technologies for accessibility features.

WCAG: W3C Web Content Accessibility Guidelines 1.0 provide information to web content authors on how to create accessible web materials.
Abstract

The design of universally accessible web resources improves access to web resources for all users, including people with disabilities. The basis of universal design is to provide users with options and control over how they view information on the web. Universal design easily adapts to user capabilities or the capabilities of the technologies they are using. Users with visual impairments can style the text on the page to be easier to read. Able-bodied users using Personal Digital Assistants or speech based telephone browsers can access content just as easily as someone viewing the information in a graphical rendering since the structure of the information is separated from the style of rendering. Even graphical renderings are enhanced since web resources adapt to the dimensions and resolutions of the users display settings. Universally designed resources can be as graphically rich as other web resources, but use different technologies to create the graphical effects.
Introduction
Tim Berners Lee developed the first HTML Web browser/editor in 1990 to enable scientists at the CERN particle physics lab in Switzerland to share electronic documents on a wide range of computing systems. At the heart of the design of Hyper Text Markup Language (HTML) technologies is the concept of interoperability, the ability of providing and receiving electronic documents using public standards on wide variety of computing equipment. The use of public standards creates an environment where developers on a wide range of computing systems can develop tools for creating, serving and viewing electronic documents using the standards. In the beginning the focus was on the information. Users typically had a wide range of choices and control over the rendering of web documents. Authors were not very interested in controlling the rendering of HTML and indeed the HTML markup language has limited features for absolute control over rendering.

Myths of the Web
As the web was commercialized through the introduction of graphical browsers (NCSA Mosaic, Netscape Navigator and Microsoft Internet Explorer) in the mid to late 1990s there was a fundamental change in the relationship between the control authors had over rendering style of web resources and the users ability to control rendering. There are many reasons for this shift, but the result is that the vast majority web authors developing commercial content for the web primarily think of the web as a graphical medium. At the same time the most popular commercial browser developers provided users with fewer and fewer options for adjusting the rendering of web resources to a point where most web users today do not even know they have any control over the rendering of web content.
Users too have developed a belief of the web as a graphical medium. This has lead to the inaccessible design of web resources by web developers who increasingly create resources that only support graphical renderings and the use of the pointing devices (e.g. the mouse) to interact with dynamic content. An example of this belief is seen in many web authors narrowing their view of interoperability to having web pages visually appear the same in both Netscape Navigator 4.7 and Internet Explorer 4.0+ even though Netscape Navigator 4.7 is an outdated technology and does not conform to HTML 4.0 specifications (CITA Surveys, 2001a and 2001b). This approach leads developers to the use of images and complex table layouts for styling and positioning text and images, giving users little opportunity to access the content in non-graphical renderings of text in Braille or speech.

**Digital Divide**

The divide between people with visual impairments and able bodied web users was investigated by Pernice-Coyne and Nielsen (2001). They found that people who use screen magnification technologies could only complete web tasks about 25% of the time and people using speech output could only complete web tasks about 12.5% of the time. When compared to the able-bodied control group performance of completing tasks about 75% of the time, it is clear that current web design is creating tremendous barriers to people with disabilities. When the visually impaired and blind did complete tasks they took them about twice as much time and visited twice as many web resources as the control group. This indicates that web resources are not providing information on the structure or organization of the information on a web page that can be used by people with disabilities to efficiently identify and find information on the web resource.
It is estimated that in 1997 approximately 48 million Americans over the age of 15 years old have some type of disability and that about 17 million identified themselves as having a severe disability (US Census Bureau, 2001). As people age the percentage of people with disabilities increases from 1.6 percent for people between the ages of 15 and 24 years old, to 5 percent for people between the ages of 15 and 64 years old, and then triples to 17 percent for people over the age of 65. So a major part of the financial argument for designing universally accessible web resources is designing web resources to deal with an increasingly aging population, and the economic power and productive capacities they bring to our nation’s economy. Kay (2000) found this barrier in the use of computer technology by people with disabilities. People with disabilities own computers at half the rate as the general population and use the Internet about a fourth of the rate as the general population. There are many factors that influence computer ownership and Internet use. Probably one of the most critical factors is how the technology is designed to be inclusive of the needs of people with disabilities. Before concrete ramps and curb cuts were built-in to the physical structures of our society people with disabilities often had few places to go and were invisible to many people in the general public. In the same way electronic ramps and curb cuts need to be built-in to our electronic infrastructure before we will see the wide-spread presence of people with disabilities on the Internet.

**Alternative Views of the Web**

For the web to become more accessible to people with disabilities web authors need to understand that people will be viewing their resources in many different ways: including graphical, text and speech renderings. This section outlines the technologies people with
disabilities use to access the web. This includes the rendering options of popular web browsers and specialized technologies designed specifically for people with disabilities.

The W3C User Agent Accessibility Guidelines (Jacobs et. al., 2002) outline the types of features browsers and multimedia players need to provide in order for people with disabilities to be able to access web content. One of the primary requirements is the ability to support the keyboard. People with many types of disabilities for various reasons can only use the keyboard to control the browser. Functions that are not available through keyboard commands will not be available to people with disabilities. People with disabilities also need to be able to select what types of content they want to view. For example, people who cannot see images, benefit more from text descriptions of the images. They would configure the current browsers to render the text description of an image in place of the graphical image. Other types of control include the ability to control the styling of text font characteristics, and the foreground and background colors when the text is rendered. People with visual impairments often need to use sans-serif fonts, larger text size and specific color combinations to make the text readable.

Automatic behaviors supported by many graphical browsers that can be disorienting, like authors automatically generating new windows. New windows are disorienting because the user is not expecting a new window to open when they follow a link. Usually the new window is given focus. When the user tries to use the back function of the browser to reorient themselves to the previous web page the page doesn’t change, since the new window did not inherit the history of the previous window. This is a usability problem for all users, but it has an increased impact on people with disabilities. Since people with
disabilities often do not have information about the choices available to them or are as well oriented to the windows open.

People who are blind cannot use the computer screen at all and use synthetic speech and refreshable Braille displays to access web content. Speech navigation and browsing is much different than graphical browsing since the user only is able to view the content linearly. When using speech it is important to provide markup that allows users to skip to important structures like headers, navigation bars and links. Otherwise users need to read the entire document to understand the information available in the document.

**Keyboard Support**

Applications written for Microsoft Windows typically have very good keyboard support in contrast to applications written for Apple Macintosh and the various flavors of the Unix X-Windows systems. This generalization is also applicable to browsers. Browsers like Netscape Navigator ([http://www.netscape.com](http://www.netscape.com)), Internet Explorer ([http://www.microsoft.com/ie](http://www.microsoft.com/ie)) and Operasoft Opera ([http://www.opera.com](http://www.opera.com)) all have keyboard shortcuts for the following functions:

- Next link or form control
- Previous link or form control
- Select link
- Move focus to next frame
- Reload content (Refresh)
- Stop loading content
- Move to previous resource in history
Opera provides additional keyboard commands that allow the user to navigate the structural elements of web resources. This includes individual functions to navigate by headers (H1-H6), form controls and HTML element-by-element. This type of function allows users to more efficiently identify the main topics of a web page, since they don’t need to rely on the styling the author used for the header to identify the text as a major topic. On pages with a large number of links it can be rather tedious to navigate to a specific link using the simple next link function available in Internet Explorer and Netscape Navigator. Opera allows the user to navigate past large numbers of links (if headers are used properly by the author) to the header closest to the link they want to select and then use the next link function from this position. Opera has a second function for navigating to links. The user can use a keyboard commands to view the list of links in the document and can use various keyboard commands to view and select individual links, including using the letter keys to move to the links that start with that letter. This type of function helps people with minimal range of motion use their physical ability more efficiently and people with visual impairments to have more options for searching.
and selecting a link. The keyboard shortcuts for a browser or multimedia player can typically be found in the help system (Figure 1).

![Help files for Internet Explorer Keyboard Shortcuts](image)

**Figure 1:** Help files for Internet Explorer Keyboard Shortcuts

### Access to Text Descriptions

One of the most important configuration options is the ability to render text descriptions for images. HTML has two attributes of the IMG element that can be used to provide text descriptions, the ALT attribute for short descriptions and the LONGDESC attribute for providing a link to a longer description. Most graphical browsers render ALT text content in place of an image when the browser is configured to not render images, but the quality of the rendering varies considerably among current browser technology. One of the major issues with rendering text descriptions is the difference in space required to render the text descriptions. Often text descriptions require more graphical space than the original image, requiring a re-flow of content to accommodate the text description.
When images are used for spacing and positioning this can often create a distorted rendering of text, making it more difficult for the user to understand the content relationships.

Currently the major graphical browsers Opera 6.1, Internet Explorer 6.0 and Netscape 6.2 do not fully support access to the text descriptions for all images only a subset or under special conditions. The HTML IMG element is the most popular way authors include images in web pages, but other elements including AREA and INPUT can have ALT attribute content. The IMG element includes an ALT attribute and the LONGDESC attribute for associating text descriptions with images. Table 1 shows the capabilities of various browsers in rendering ALT text descriptions.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Operating System</th>
<th>Render ALT Text for IMG element</th>
<th>Render ALT Text for AREA element</th>
<th>User Styling of ALT text</th>
<th>Link to LONGDESC URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opera 6.01</td>
<td>Windows 98/2000/XP,</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Macintosh OS9 and Unix</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>Windows 98/2000/XP</td>
<td>Yes, except when scripts</td>
<td>No</td>
<td>Limited</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
<td>--------------------------------------</td>
<td>----------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>6.0</td>
<td>dynamically change the image source attribute</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Internet Explorer 5.1</td>
<td>Macintosh OS9</td>
<td>Yes, unless images are cached</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Netscape 6.2</td>
<td>Windows 98/2000/XP, Macintosh OS9 and unix</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes, through context menu that is only accessible with mouse commands</td>
</tr>
</tbody>
</table>

Table 1: Browser capabilities of rendering ALT text and providing a link to LONGDESC URI

Figures 2, 3 and 4 show the ALT text rendering of the same web page for Opera 6.1, Internet Explorer 6.0 and Netscape Navigator 6.2 respectively. Opera renders the ALT text and has extensive styling capabilities for the ALT text. Internet Explorer for Windows renders that ALT text and limits the ability of the user to style the ALT text. Netscape Navigator does not render ALT text when rendering of images are turned off. The text content of the ALT attribute is designed to provide a short text description of an image. The LONGDESC attribute provides a URI to a web resource that will provide a
more detailed description of the image. For example, if the image was a chart of what flavors of ice cream people prefer at a certain ice cream store, the LONGDESC could point to a web page with a text table that representation of the preferences of ice cream. Opera has a very good implementation of rendering ALT text, since it provides the user with extensive control over styling the ALT text. Other implementations, including Internet Explorer, provide various levels of access to ALT text to no access in the case of Netscape 6.2. Access to the ALT text has much better implementation than access to the LONGDESC attribute. It is ironic that the one browser listed, Netscape 6.2, that provides access to the LONGDESC URI does not provide access to the ALT text description. The table shows that no browser provides complete access to ALT text descriptions limiting the types of content some users will have access to using these browsers.
Figure 2: View of ALT text rendering in Opera

Figure 3: View of ALT text rendering in Internet Explorer
The `<INPUT>` element of type “image” and `<AREA>` element that defines the clickable areas on an image MAP also have ALT attributes. Currently none of the major graphical browsers support the in content rendering of ALT text associated with the AREA element, some provide access through display as a tooltip (e.g. a pop text description of an element that appears when the pointing device hovers over an element rendering) and many assistive technologies like screen can read the value of the ALT text. The lack of support makes the links of the image MAP elements inaccessible to many people with...
disabilities who are low vision and not using assistive technologies. Authors should always provide redundant text links for both server side and client side image MAPs.

**User Styling of Text**

People with visual impairments and learning disabilities that effect reading need to control the font characteristics, font size, and foreground and background colors of text. In early graphical browsers, like NCSA Mosaic, this was built-in feature and the user could completely configure the default style sheet used for rendering HTML. Current browsers vary widely in their ability to allow the user to control the rendering of text. Table 2 shows the capabilities of several popular browsers in allowing the user to control the rendering of text.
Figure 5: Opera 6.0 settings for setting the rendering style information used in author and user modes

The W3C Cascading Style Sheet (CSS) technology was designed to address many of the author and user styling issues of separating the structure of a web resource and the styling for a particular rendering. The advantage to developers in using style sheet technology is that a single style sheet can be used to control the rendering of any number of web pages, making it easier for web masters to change the look and feel of their website without having to individually edit pages or elements. One of the more powerful aspects of the CSS specifications is user style sheets. The W3C realized that users need control over rendering and the specification includes the concept of user style sheets overriding author supplied style sheets. Opera has actually implemented the concept of user and author styling and provides a very concrete interface for users to select author and user styling preferences (Figure 2). Opera also provides a one step command (clickable icon or single key press) for the user to switch between author styling and user styling of web content. This is a very useful feature not only for users, but also web developers. Web authors can easily switch between their designs and a high contrast styling that might be used by someone with a visual impairment. The high contrast setting helps them to check if their web design will work for someone with a severe visual impairment or using portable technologies like a PDA or speech browsers that do not have the same rendering characteristics as a graphical browser. Microsoft Internet explorer implements user style sheets, but does not allow the user to completely ignore style sheets supplied by the
Table 2 shows the features available to users to control the author supplied styles and to apply their own style sheets in various browsers.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Operating System</th>
<th>Ignore Author Font Sizes, Font Style and colors</th>
<th>Adjust default Font Sizes, Font Style and colors</th>
<th>Ignore Author Style Sheets</th>
<th>Add user Style Sheet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opera 6.01</td>
<td>Windows 98/2000/XP, Macintosh OS9 and Unix</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Explorer 6.0</td>
<td>Windows 98/2000/XP</td>
<td>Yes</td>
<td>Yes, limited font size control except through style sheets</td>
<td>Limited to fonts and colors</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet Explorer 5.5</td>
<td>Macintosh OS9</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes, but only in cascade with author style sheets (i.e. cannot use user style sheets when...</td>
</tr>
<tr>
<td>Author Style</td>
<td>User Style Sheets</td>
<td>Speech Browsing</td>
<td>Author Style Sheets are turned on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>----------------</td>
<td>---------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netscape 6.2</td>
<td>Windows 98/2000/XP, Macintosh OS9 and unix</td>
<td>Yes</td>
<td>Yes</td>
<td>Limited to fonts and colors</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2: Browser capabilities of overriding author styles with user style sheets

**Speech Browsing**

Speech browsing is a fundamentally different way of accessing web information. In a graphical rendering the author often uses spatial relationships to group information on the screen and the users passively scan the screen to identify the grouping of information. But a speech rendering is temporal and requires the user to issue commands to direct the browser to read and reread content. The user could issue a command to speak the entire content of a web page, but in general this would be an inefficient way for the user to locate information they are interested in. It is the equivalent of a sighted user reading the entire contents of a web page before they started looking for links or other groupings of information on the page. Most able-bodied users scan the screen for highlighted text and other visual cues to identify the groups of information in the web page. In a well designed web page the author has intentionally created cues to help users focus their attention on information the author thinks is important. The same is true in speech
browsing. If authors include structural markup, users using any technology can style that structure to highlight the information to the user by way of speech, text or graphical renderings.

Figure 6: Main menu reading options in IBM Home Page Reader

Speech browsers like IBM Home Page Reader (http://www.ibm.com/able) and Freedom Scientific’s JAWS (http://www.freedomscientific.com) screen reader have features for users to navigate HTML structural information. For example, they can navigate to elements marked as headers, links, form controls and through table data cells. These functions only work when authors use HTML header and other markup correctly in their web resources. Figure 6 shows the read menu in IBM Home Page Reader (HPR) which highlights the different ways that HPR can be used to navigate through content. By
providing users access to the structured markup it is easier for users to find the main
groups of content without reading the entire document. Speech can be styled to indicate
different types of elements. In IBM Home Page Reader the reading voice for links is
styled as a female voice and while non-link content is styled as a male voice for static
text (voice can be configured to other settings by the user). Form controls that have
explicit text labels associated with them can have their labels announced when the form
control receives focus.

Many web pages do not contain structural markup. This forces speech browsers to look
for implied structure if they want to offer the user more than just a linear reading of the
document. In the example of form controls, speech browsers may try to calculate the
relationships between the form control and text around the form control to determine the
label for the control. This is problematic since the guess can be wrong and instead of
helping the user, the user maybe confused over the purpose of the control. If images do
not have ALT text descriptions, current speech browsers may use the file name of the
image in hope that it may contain some useful information about the purpose and the
content of the image. These approaches may help accessibility when the calculations are
correct. When calculations are not correct they can seriously hurt accessibility by
increasing the confusion and misinformation to users which results in users taking more
time to explore the resource or improperly completing the task they are trying to
complete.