Accessibility

General guidelines
Disabilities
Barriers
Tools

Claudio Cathomen <c.cathomen@icsurselva.ch>
August 2005

1. Introduction

10% to 20% of the world's population have some form of a disability. Not all of those disabilities affect access to the Web, however some disabilities do. For example disabilities with vision, hearing, dexterity, short-term memory problems.

Other issues are the ages of the web users. The average age of population in many countries is increasing, and aging sometimes results in combinations of accessibility issues: vision & hearing changes, changes in dexterity & memory. Few organizations can afford to deliberately miss this market sector.

The goal of creating accessibility websites is, that every user, regardless of physical, sensory and cognitive disabilities, constraints and/or technological barriers can:

- Access the information
- Use the services
- Buy the products
- Talk to the people associated with each Web site.

Besides, a number of governments (including the EU, US) require Web accessibility for certain kinds of sites.

To produce compliant websites and applications it is important to know more about

- Disabilities
- Helping soft/ and hardware for disabled persons
- Standards
- Helping software / validators for developers

Anyway, it is strongly recommended to work together with disabled persons or organizations, so your site/ application is checked 1 to 1 by relevant people.

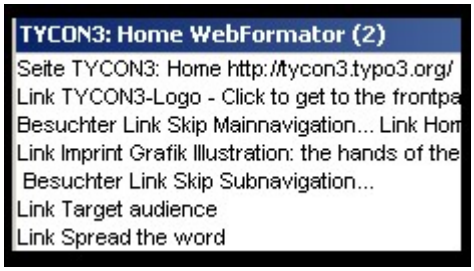
2. Impact of disabilities:

2.1 Blind persons

Blind persons use no screen, no mouse.
Blind people use "screen readers" and in combination text2speech applications.

As it turns out, there are software programs that convert text into synthesized speech and blind people are able to listen to Web content. These software programs are generally called "screen readers," even though they do quite a bit more than simply read the screen. Screen readers allow users to navigate through Web content in many ways. The user can simply let the screen reader read everything from top to bottom, one line at a time, or the user can use the tab key to navigate from link to link. The user can also navigate from one heading to the next (if the Web content has headings), from one frame to the next (if there are frames), or by other methods. Common screen readers include JAWS, Window Eyes, Home Page Reader, ZoomText (with the free availeble Webformator)).

Screen readers can also be used by those who are both deaf and blind, but rather than convert text into speech, screen readers for the deaf-blind convert text into Braille characters on refreshable Braille devices.



Screen readers are quite robust in their capabilities, but they do have limitations. They cannot completely substitute for the visual experience. Luckily, there are ways to compensate for screen reader weaknesses.

Images. Screen readers cannot describe images. The only way that a screen reader can convey the meaning of an image is by reading text in the document that serves as a substitute, or alternative, for that image. If there is no alternative text, or "alt text," then the screen reader cannot accurately convey the meaning of an image.

Visual layout. Also, screen readers cannot survey the entirety of a Web page as a visual user might do. A visual user can look at a Web page and quickly realize how the page is organized, then zero in on the most important content. A screen reader is not able to do this. It reads in a linear fashion, one word at a time. It cannot always intelligently skip over extraneous content, such as advertisements or navigation bars.

Data tables. Similarly, because screen readers must read linearly, data tables can be quite confusing. Imagine trying to listen to a large data table with 14 columns and 28 rows. By the time you get to the 20th row, you will have long forgotten what the heading was for column 8. It can be difficult, if not impossible to interpret such information.

JavaScript. The fact that screen reader users use their keyboard as their primary means of navigating the Web is a point that developers need to pay attention to. Without even realizing the consequences, some Web developers program their site in such a way that it works only if the person is using a mouse. Most of the time, keyboard inaccessibility is a result of JavaScript events that depend on either the click or the movement of the mouse. For example, any event triggered by the onmouseover event will only work when the user moves the mouse over that particular object. There is no way to access this type of content without providing some other method, either in addition to, or instead of the onmouseover event.

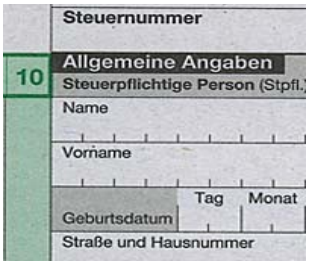
Key Concepts for Blindness

Challenges	Solutions
Users generally do not use a mouse	Don't write scripts that require mouse usage. Supply keyboard alternatives.
Images, photos, graphics are unusable	Provide text descriptions, in "alt" text and, if necessary, longer explanations (either on the same page or with a link to another page).
Users often listen to the Web pages using a screen reader	Allow for users to skip over navigational menus, long lists of items, ASCII art, and other things that might be difficult or tedious to listen to.
Users often jump from link to link using the TAB key	Make sure that links make sense out of context ("click here" is problematic).
Frames cannot be "seen" all at once. They must be visited separately, which can lead to disorientation.	Don't use frames unless you have to. If you use them, provide frame titles that communicate their purpose (e.g. "navigational frame", "main content").
It may be difficult for users to tell where they are when listening to table cell contents	Provide column and row headers (<th>). Make sure that tables--especially those with merged cells--make sense when read row by row from left to right.
Complex tables and graphs that are usually interpreted visually are unusable	Provide summaries and/or text descriptions.
Not all screen readers support image maps	Supply redundant text links for hot spots in image maps
Colors are unusable	Do not rely on color alone to convey meaning
Users expect links to take them somewhere	Don't write scripts in links that don't have true destinations associated with them (e.g. href="javascript: function(this)")

2.2 Partially sighted persons

People with low vision often use screen magnification software
People with low vision often benefit from higher contrast.
People with low vision sometimes override the document's default background and font colors.

The visual acuity of people with low vision varies widely, but, in general, low vision is defined as a condition in which a person's vision cannot be fully corrected by glasses, thus interfering with daily activities such as reading and driving. Low vision is more common among the elderly, but it can occur in individuals of any age as a result of such conditions as macular degeneration, glaucoma, diabetic retinopathy, or cataracts. Each of these conditions causes different types of effects in a person's vision.



Form (Paper); Macular degeneration

The most common technology that people with low vision use is the screen magnifier. This is a software program that zooms in on a small area of the screen, allowing people with low vision to see it more clearly. Common screen magnifiers include ZoomText and MAGic. Some kinds of content are difficult to interpret when enlarged. For example, graphics that contain text can become blocky and pixilated, making the text difficult to understand.

High Contrast. Sites with low contrast can be difficult to read for people with low vision. Some poorly designed sites on the Web have bad color combinations such as blue links on black backgrounds, red text on green backgrounds, or other combinations that are not easy on the eyes for anyone, but especially not for people with low vision. There's no hard rule as to how much contrast is enough, but use your best judgment. However, it's usually not too difficult to tell when color combinations do not contrast adequately.

Overriding Font and Background Colors. Some people with low vision will change the settings in their operating system and/or browser to not only enlarge the text, but to increase the contrast of the text in relation to the background. Some people like to have a black background with white or yellow text. Others prefer to have a white or yellow background with black text. These are the most common settings, but there other people prefer other high contrast settings.

Horizontal Scrolling. This point is not so much a matter of accessibility as it is usability. You have probably come across Web sites that require you to use your horizontal scrollbar to see the content on the right of the screen, even though you had your browser window maximized. This can be a bit annoying to people with perfect vision, but it is even more so for people who use screen magnifiers and are forced to scroll even further to the left and right inside of the small enlarged space they are viewing.

Challenges	Solutions
Text in graphics does not enlarge without special software, and looks pixilated when enlarged	Limit or eliminate text within graphics
Users may set their own font and background colors	Allow them to do so by using as much real text as possible, rather than text within graphics.
Screen magnifiers reduce the usable window size	To reduce that amount of horizontal scrolling, use relative rather than absolute units (e.g. use percentages for table widths instead of pixels)

2.3 Motoric disabilities

Some people may not be able to use a conventional input device, such as the mouse or the keyboard. Therefore it is important for software functions to be accessible using both or either device; Keyboard shortcuts and mouse gestures are ways to achieve this. More specialized solutions like on-screen keyboards and alternate input devices like joysticks and trackballs are also available.

Arthritis

One-handed keyboard (for the right hand)

Key Concepts: Motor Impairments

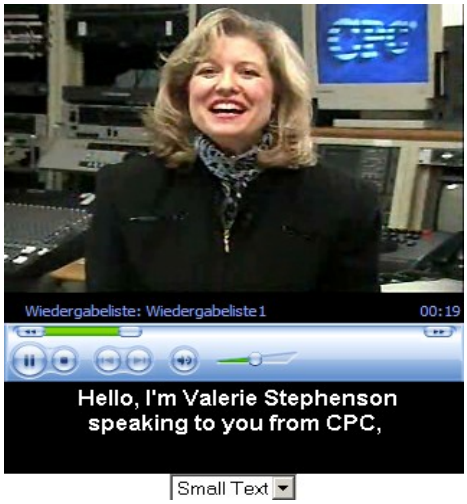
Challenges	Solutions
Users may not be able to use the mouse	Make sure that all functions are available from the keyboard (try tabbing from link to link).
Users may not be able to control the mouse or the keyboard well.	Make sure that your pages are error-tolerant (e.g. ask "are you sure you want to delete this file?"), do not create small links or moving links.
Users may be using voice-activated software.	Voice-activated software can replicate mouse movement, but not as efficiently as it can replicate keyboard functionality, so make sure that all functions are available from the keyboard.

2.4 (Partially) Deaf persons

Deafness is not an all-or-nothing condition. Although there are individuals who are completely deaf, there are also individuals with varying degrees of functional hearing loss. Degrees of hearing loss are often categorized as mild, moderate, severe, profound. Those who refer to themselves as deaf usually have either severe or profound hearing loss. Those with lesser degrees of hearing loss are commonly referred to as hard-of-hearing.

Most developers don't think about individuals who are deaf when they think of Web accessibility. But video, audio, and multimedia content is becoming more and more common on the Web. Video content is available on most major news Web sites, even some local news Web sites. Unfortunately for those who are deaf, captioned audio is still almost nonexistent on the Web. The tools to caption Web video exist, and the concept of captioning has been around for decades. It's just a matter of doing it.

Captioned Web video is even required by law in some places (i.e. federal government and other Web sites that fall under Section 508 of the United States).



Captioned Web video

2.5 Cognitive disabilities

The concept of cognitive disabilities is extremely broad, and not always well-defined. In loose terms, a person with a cognitive disability has greater difficulty with one or more types of mental tasks than the "average" person. There are too many types of cognitive disabilities to list here, but we will cover some of the major categories. Most cognitive disabilities have some sort of basis in the biology or physiology of the individual. The connection between a person's biology and mental processes is most obvious in the case of traumatic brain injury and genetic diseases, but even the more subtle cognitive disabilities often have a basis in the structure or chemistry of the brain.

A person with profound cognitive disabilities will need assistance with nearly every aspect of daily living. Someone with a minor learning disorder may be able to function adequately despite the disorder, perhaps even to the extent that the disorder is never discovered or diagnosed. Admittedly, the wide variance among the mental capabilities of those with cognitive disabilities complicates matters somewhat. In fact, one may reasonably argue that a great deal of Web content cannot be made accessible to individuals with profound cognitive disabilities, no matter how hard the developer tries. Some content will always be too complex for certain audiences. This is unavoidable.

Nevertheless, there are still some things that designers can do to increase the accessibility of Web content to people with less severe cognitive disabilities.

Key Concepts for Cognitive Disabilities

Challenges	Solutions
Users may become confused at complex layouts or inconsistent navigational schemes.	• Simplify the layout as much as possible. • Keep the navigational schemes as consistent as possible
Users may have difficulty focusing on or comprehending lengthy sections of text	Where appropriate, group textual information under logical headings. Organize information in manageable "chunks."
One method of input may not be sufficient	Where appropriate, supplement text with illustrations or other media, and vice versa.

3. W3C's Web Accessibility Initiative (WAI)

The W3C Web Accessibility Initiative (WAI) develops strategies, guidelines, and resources to help make the Web accessible to people with disabilities.

http://www.w3.org/WAI provides

- Accessibility Tools List
- Introduction to Web Accessibility
- Quick Tips to Make Accessible Sites
- Web Content Accessibility Guidelines (WCAG)
- Checklist of Checkpoints for Web Content Accessibility Guidelines 1.0 (http://www.w3.org/TR/WCAG10/full-checklist.html)

The W3C/WAI checklist is an easy-to-follow list of the guidelines and helps developers to create accessibility conform sites and applications.
Each checkpoint has a priority level assigned by the Working Group based on the checkpoint's impact on accessibility.

[Priority 1]
A Web content developer must satisfy this checkpoint. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use Web documents.
Example: Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content).

[Priority 2]
A Web content developer should satisfy this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.
Example: Use style sheets to control layout and presentation.

[Priority 3]
A Web content developer may address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to Web documents.
Example: Provide summaries for tables.

4. Legal regulations in different countries

The Web's emergence as a pivotal form of Information and Communications Technology (ICT) raises interesting questions about application of existing laws and policies to this new medium, and the importance of all members of society, including people with disabilities, being able to access this information medium.

There is a growing body of national laws and policies which address accessibility of ICT, including the Internet and the Web. There is also great variety of approaches among these laws and policies: some take the approach of establishing a human or civil right to ICT; others the approach that any ICT purchased by government must be accessible; others that any ICT sold in a given market must be accessible; and there are still other approaches.

http://www.w3.org/WAI/Policy provides a list of policy links for some country's.

5. Basic technical rules and principles

Page Design

- Be consistent with the page layout throughout the web site this allows for easier and faster navigation
- Use code to provide accessibility or provide text only version of each page of the web site
- If page contains an image map, provide a text version with links and descriptions as some tools designed for people with disabilities to browse web sites cannot read image maps
- Don't use frames or provide a method to view without frames
- Provide an option to skip repetitive links.
- Make the site script independent because certain specialized accessibility tools cannot use client side scripting
- Don't use deprecated elements (i.e.: bold, italic, center), use of style sheets is preferred
- Make sure page style sheet independent. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document
- Provide metadata to add semantic information to pages and sites
- Create a logical tab order through links, form controls, and objects to allow for faster and easier navigation

Tables

- Only the last cell should be allowed to wrap
- Use "longdesc" tag for explaining the content of tables
- Place a break before each ending table data cell
- Put row and column headers in data tables.
- Associate all data cells with header cells so that screen readers will read the content of cell with the header of that cell
- Do not use tables for layout unless the table makes sense when linearized, because screen readers and other tools read tables linearly. Otherwise, if the table does not make sense, provide an alternative equivalent

Links

- Use "title" tag to explain where each link takes the user
- Use "title" tag to explain each link for e-mails
- Try to incorporate graphical links with text descriptions for some links (i.e.: next, previous, etc.)

Images

- Use "alt" tag for all images
- Use "longdesc" tag for further explanations, if needed
- Provide a text version of images for lengthy descriptions (i.e.: image map, org. chart gif, etc.)
- Don't use flashing, flickering or blinking images
- Use High Contrast Images so that these images can be easily seen by people with visual disabilities including color blindness.
- For rules, a good option for the "alt" tag: ALT="- - -"
- For bullets, a good option for the "alt" tag: ALT="*"

Color

- Use contrasting colors so that these colors can be easily seen by people with visual disabilities including color blindness
- Browser safe colors are best
- Ensure that all information conveyed with color is also available without color, for example from context or markup
- Use Hex code for colors and not names

Text

- Don't use phrases such as "click here"
- Don't refer to items in color, (i.e.: "refer to the pink bulleted items")
- Expand abbreviations and acronyms for the first occurrence on each page
- Don't use moving text
- Contrast text colors with background colors
- Don't use H1, H2, H3, etc. for font effects, instead Use header elements to convey document structure and use them according to specification

Forms

- Provide an e-mail address or link as an alternative to completing on-line forms
- Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned

Other

- Identify language and Identify any changes in language
- Divide large blocks of information into more manageable groups where natural and appropriate
- Provide a means to skip over multi-line ASCII art
- Avoid use of ASCII Art as screen readers will read "| fsu |" as "vertical line fsu vertical line"
- Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video

6. Electronic (online) check tools

Online validation tools

	Automatic Checkpoints			Manual Checkpoints		
	Status	Errors	Instances	Status	Warnings	Instances
Priority 1	✓	0	0	⚠	10	14
Priority 2	⊗	1	1	⚠	17	20
Priority 3	✓	0	0	⚠	10	10

W3C
XHTML validator
http://validator.w3.org/

CSS validator
http://jigsaw.w3.org/css-validator/validator-uri.html

Bobby
http://webxact.watchfire.com

Cynthia Says
http://www.cynthiasays.com/

http://www.w3.org/WAI/ER/existingtools.html provides a good list of certification and validation tools.

7. Working Methods

If you plan to build a accessible website, you should

- know about disabilities
- provide validated code
- work on one step, then check
- consult disabled persons/ disabilities organizations

Ressources:

http://www.fsu.edu/~webguide/accessibility/
http://www.w3.org/WAI/Policy/
http://www.webaim.org/
http://webxact.watchfire.com