

Session on Accessibility for ITS webteam

Introductions

- Who we are
- Our role
- Your knowledge and understanding of accessibility

5-10 mins

Overview

Accessibility: What does it mean?

- making information available to all users regardless of disability
- a broad range of disabilities must be allowed for
- different design considerations are required

Why is it important?

- Disability Discrimination Act
- DDA Action Plan
- Deakin Policy
(approved by Academic Board in January 2002 and available on DUGS—search for ‘accessibility’+‘policy’)
- W3C and section 508
- Be aware that we are not only concerned with people who are identified as ‘disabled’ through the DRC—applies to anyone with a short-term disability such as a broken arm or with a mild disability such as short-sightedness.
- Universal design

Your responsibilities as a Deakin staff member

- You are responsible for ensuring that materials provided for students and staff are available in an 'accessible' format where it can be reasonably provided
- Applies to all materials provided in electronic format: online (including non-teaching materials), in CD-ROM format, or on floppy disk.

Adaptive technologies

Different types of disabilities

- *People who are blind*
Usually use screen reading software which is then sent to a text-to-speech synthesizer or refreshable Braille display
- *People with low vision including blurred or fogged vision, loss of central vision, tunnel vision etc.*
Commonly enlarge or otherwise enhance current area of focus. Need to be able to adjust fonts, colours, and cursors to make them more visible.
- People with colour blindness
Operate in monochrome mode
- People who are hard of hearing or deaf
- *People with physical disabilities*
Wide range of abilities and limitations
Access generally dependent on being able to manipulate the interface.
- *People with language or cognitive disabilities*
Need software designed to be very user friendly

Types of assistive technologies

- *Screen readers* e.g. Jaws
Applications that use the operating systems standards will work best

- *Screen magnifiers*
Software for low-vision users that allows the user to customise the size of images and text
May also permit changing default colours
Track cursor or the active region of the screen—therefore applications and software that use a custom cursor can be a problem
- *Adaptive keyboards*
Applications and software that depend on reading keystrokes directly from the keyboard rather than through the operating system are not likely to be accessible
- *Voice recognition software* e.g. Dragon Professionally Speaking
Beneficial to users who have difficulty typing
Applications and software that allow full access through keyboard commands will work well.
- *Single switches* e.g. head wands
For users who can only control the computer with one or two specific movements. Software scans through options on the screen allowing the user to trigger the switch when the option they wish to choose is highlighted.

5-10 mins

Equivalent vs alternative access

- *Equivalent access*
Provides the same activity but it is mediated in a different modality
e.g. providing a course textbook in Braille format
- *Alternative access*
Provides a different activity but one that is designed to meet the same objectives
e.g. a mobility impaired student conducting science experiments in a virtual laboratory.

- Equivalent access should be provided as first option. Alternative access only if equivalent access is not possible.

Direct access vs compatible access

- *Direct access*
User can operate all on-screen controls and access content without using an assistive technology
- *Compatible access*
Designed to be compatible with assistive technology

General principles

- *Allow for customisation based on user preference*
Font, font style, font colour, font size
Cursor size, style and blink rate
Size of text and images, including video
Screen layout, colours and backgrounds
Timing of events
Keyboard settings
- *Provide equivalent access to auditory and visual content based on user preference*
Adding text description for all static images
Providing a text transcription of auditory content
Using 'longdesc' attribute for images that require more lengthy descriptions
Captioning of auditory content
Provide audio description tracks for multimedia, describing visual aspects of the content.
- *Provide compatibility with assistive technologies and complete keyboard access*
Don't assume the user can use a mouse
- *Provide context and orientation information*
Links must be meaningful—avoid 'Click here'
Inform users about how to navigate, when to find menu info and how long the page is

Provide a way to skip standard page headers and navigation links, so that users who are already familiar with the page layout can skip directly to the primary content

Maintain a consistent layout of pages so the users do not have to keep familiarising themselves with a different layout

Provide alerts if a new browser window will be opened automatically

- *Follow standards and guidelines*
W3C, WCAG, User Agent Guidelines, Authoring Tool guidelines, HTML 4.0, CSS standards
(<http://www.w3c.org>)
- Consider XML
Used to create languages that separate structure and display

Text

Problems

- Hard-coding of fonts so users can't change style, size, color
- Text presented with background images or poor contrast colors
- Text presented in an image format
- Multi-column formats that screen readers cannot process in the correct order

Solutions

- Use style sheets in conjunction with correct structural markup
- Avoid graphical representations of text
- Use XML, HTML, plain text

Audio

Problems

- Lack of captioning and/ or transcription
- Poor sound quality
- Inability to control volume

Solutions

- Provide text transcript or captions—MAGpie is a free tool for adding captions, subtitles and audio descriptions to digital media (<http://www.ncam.wgbh.org/webaccess/magpie>)
- Provide volume controls
- Provide visual equivalents to audio alerts

Images

Problems

- No alternate text provided
- Poor image resolution

Solutions

- Provide text representation of images
- Use SVG features that improve accessibility

Multimedia

Problems

- Digital video without captions or audio descriptions

Solutions

- Provide audio description describing essential visual elements

Threaded message boards

Problems

- Complex framesets without title and/ or name attributes
- Lack of ALT text on buttons that function with Javascript to expand or contract threaded discussions
- Form fields that do not accommodate keyboard navigation

Solutions

- Ensure all actions can be completed from the keyboard
- Provide help files, including an orientation to the interface
- Use text for navigation links
- Use style sheets to create buttons
- Provide by-pass links to allow users to skip repetitive navigation links and go directly to the content of the page

Organisers, schedulers and calendars

Problems

- Scripts that use event handlers that are device-dependent (i.e. require a mouse)
- Inability to navigate among table cells and access header and other table cell information due to incorrect mark-up.
- Content of table cells does not linearise correctly (Use The Wave to analyse linearise reading order http://www.temple.edu/inst_disabilities/piat/wave/)

Solutions

- Ensure tools are usable when scripts are **not** supported or provide an alternative page
- For data tables, identify row and column headers, use abbreviations for header labels and identify structural groups of rows or columns.

- Test table content to ensure it linearises correctly or provide a linear alternative

Interface controls

Problems

- Controls labelled with images rather than text
- Controls require use of a mouse
- Display options not easily located or not accessible through keyboard navigation

Solutions

- Ensure all actions can be completed from the keyboard
- Test interfaces using assistive technologies

Navigating the interface

Problems

- Complex framesets without title or name attributes
- Table of contents with expand/collapse features that are indicated with images with no text labels
- Menu bars built in scripting languages not usable with assistive technologies

Solutions

- Provide names, titles, or text labels for each element of the interface
- Ensure all parts of the interface can be reached by keyboard and document appropriate keystrokes

Forms

Problems

- Illogical tab order among controls

- Complex layout of controls that makes it difficult to determine which label matches which control, or to logically connect a series of related controls
- Form fields in search utilities that do not accommodate keyboard navigation
- Categories of form fields that are colour-coded or marked with an image, such as which fields are required, or where an error exists

Solutions

- Ensure the tab order makes sense
- Use clear labels for each form item
- Ensure all actions can be completed from the keyboard
- Do not rely on colour alone to differentiate information
- Provide a means for users to easily locate and correct form entries that have errors
- Place a default value in edit boxes or on the top line of drop-down lists

Interactive exercises

Problems

- Tasks can only be completed with a mouse even though they do not require drawing
- Activities that require monitoring information at one side of the screen while simultaneously entering information on the other side
- Information that is based on numbers but is shown visually
- Lesson directions provided in audio with no text transcript or captions
- Text displayed over a patterned background

Solutions

- Ensure all actions can be completed from the keyboard
- Provide features that allow users to access multiple sources of information separately when they are delivered simultaneously
- Allow user to customise the timing of events

Some common problems

- *Drop-down lists*
Can be difficult to navigate for users using the keyboard to access options or a speech reader
Provide a 'Go' button separate from the list of choices
- *Forms*
Put the labels adjacent to their controls
Use HTML markup to associate the controls explicitly with their labels
- *Large files*
Users often have problems downloading large files.
Remember to 'zip' where necessary.
- *Help documentation*
Users with disabilities use them more than other users.
Make sure it's provided, is comprehensive and accessible!
- *Word / PDF documents*
Structural elements are just as important as for HTML.
Make sure standard Word heading styles etc. are used.

Exercise

Images

Photographs



ALT= "??????"

Images used as text



ALT= "??????"

Graphics buttons used for navigation



ALT= "??????"

Images used for visual purposes e.g. bullets



ALT= "?????"

Alternatives for video, audio, animations

Use of colour

The preferred method is shown in red.

Method A

Method B

Required fields are shown in green

Name

Email

City

Fix these examples so that a user with a monochrome screen could interpret them.

Markup and style sheets

Fix the following code...

```
<p class=subhead>
```

```
<table>
  <tr>
    <td width="4%"></td>
    <td width="96%">list item 1</td>
  </tr>
  <tr>
    <td><im src="bullet.gif"></td>
    <td>list item 2</td>
  </tr>
</table>
```

Javascript

- Use of scripts such as Javascript may create barriers for some users of adaptive technology. For example, if a script generates content on the "mouseover" event handler, as in the case of "rollovers," anyone who relies on keyboard access will not be able create the event that will cause the browser to generate the text.
- In addition, screen readers will often read the content of the script itself in a meaningless jumble of numbers and letters. Although this jumble is text, it cannot be interpreted or used.
- Javascript components that do not generate content dynamically are generally not a problem for users of adaptive technology. Examples include confirmation of entry of form fields, browser version identification, etc.

Functional Text

- Include "functional text" when web pages rely on scripted programming instructions.
- Text will be presented that conveys an accurate message as to what is being displayed by the script.
- E.g. If a web page uses a script to create a graphic map of menu choices when the user moves the pointer over an icon, the web site designer may be required to incorporate "redundant text links" that match the menu choices because functional text for each menu choice cannot be rendered to the assistive technology.

Some techniques recommended by the Web Accessibility Initiative

- Ensure links that trigger scripts work when scripts are turned off or not supported (e.g. do not use "javascript:" as the link target).
- If it is not possible to make the page usable without scripts, provide a text equivalent with the NOSCRIPT element, or use a server-side script instead of a client-side script.
- In HTML, use NOFRAMES at the end of each frameset. For some applications, server-side scripts may be more accessible than client-side scripts.
- Use the <NOSCRIPT> element to describe what is being displayed by the script if the client technology does not support the scripting language.

Example:

```
<SCRIPT type="text/tcl">
...some Tcl script to show a billboard of sports scores...
</SCRIPT>
<NOSCRIPT>
<P>Results from yesterday's games:</P>
```

```
<DL>
<DT>Bulls 91, Sonics 80.
<DD><A href="bullsonic.html">Bulls vs. Sonics game highlights</A>
...more scores...
</DL>
</NOSCRIPT>
```

Flash/ Dreamweaver accessibility features

- Problems with Macromedia Flash presentations in terms of providing keyboard accessibility
- Some mainstream browsers, such as Netscape and early versions of IE, do not allow the user to use the TAB key to bring focus to the Flash object. For IE, the ACCESSKEY attribute can be added to the OBJECT tag that contains the Flash Presentation so a key combination (ALT 6) can bring the object into focus. With more recent versions of IE, the TAB key can be pressed repeatedly until the movie receives focus.
- Once the Flash object is in focus however, some browsers, such as IE, will not allow the user to bring focus back to the page in which it is embedded - hence the user becomes "trapped" in the Flash object.
- Hard of hearing or deaf users will be unable to hear the audio component of a Flash element. Include a transcript of the narrative.
- Similarly, a blind user using assistive technology that does not access the Flash movie will need access to a description of the movie and alternative format for the audio output available through the D-link (or LONGDESC).

WebCT and accessibility

What's good about it and what's not

5 mins

Evaluation

Different browsers (IE, Netscape, Opera, Mozilla)

JAWS

Bobby and other automated tools

CSS style validator

Common sense

User feedback