

DESIGNING FOR ACCESSIBILITY

Lecture 8

Agenda

- Disabilities
- Accessibility
- Accessibility APIs
- Accessibility evaluation

DISABILITIES

Basic notions

- Impairment: any **loss or abnormality** of psychological, physiological or **anatomical structure or function**.
- Disability: any **restriction** or lack (resulting from an impairment) **of ability** to perform an activity in the manner or within the range considered normal for a human being.
- Handicap: a **disadvantage** for a given individual that limits or prevents the fulfillment of a role that is normal

Impairments

- **Vision Impairments**
- **Dexterity and Mobility Impairments**
- **Hearing Impairments**
- **Learning Impairments**
- **Language and Communication Impairments**
- **Age-related Impairments**

Disabilities

- **“Disabilities** is an umbrella term, covering **impairments, activity limitations, and participation restrictions**. An impairment is a problem in body function or structure; an activity limitation is a difficulty encountered by an individual in executing a task or action; while a participation restriction is a problem experienced by an individual in involvement in life situations. **Thus disability is a complex phenomenon, reflecting an interaction between features of a person’s body and features of the society in which he or she lives”**
(*World Health Organization*)
- **Currently around 10 per cent of the total world's population, or roughly 650 million people, live with a disability**
- **females have higher rates of disability than males.**

Visual disabilities

- Vision difficulties and impairments include
 - low vision
 - color blindness
 - blindness.
- Visual impairment is vision loss (of a person) to such a degree to qualify as an additional support need through a significant limitation or visual capability resulting from either disease, trauma, or congenital or degenerative conditions that cannot be corrected by conventional means, such as refractive correction, medication or surgery.

ACCESSIBILITY

Accessibility

- Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible
- **Accessibility is often used to focus on people with disabilities and their right of access to entities, often through use of assistive technology.**
- Accessibility, as it pertains to information technology (IT), is about removing barriers that inhibit the access of certain groups, including people with disabilities, mature users, and non-native language learners
- Understanding accessibility requires an awareness of the special needs of multiple user groups, including people with disabilities and mature users with age-related disabilities

Understanding (visual) disabilities

- People with visual disabilities need text equivalents for images, because their assistive screen reader technology cannot obtain information from images
- The mouse is not useful because it requires hand and eye coordination
- Will navigate the Web using keyboard (the Tab key is pressed to move the focus to an item that needs to be selected and the Enter key is pressed instead of clicking the mouse button)
- Low vision users need the assistance of a hardware or software magnifier to enlarge the text beyond simple font enlargement
- Color blind and low vision users benefit from high contrast colors

Understanding (visual) disabilities

- The assistive technologies that can be used by people with visual disabilities are:
 - screen magnifiers
 - screen readers
 - speech recognition software
 - speech synthesizers
 - refreshable Braille displays, Braille embossers,
 - talking word processors
 - large-print word processors

Assistive Technologies

- Assistive technology is a generic term that includes assistive, adaptive and rehabilitative devices for people with disabilities and includes the process used in selecting, locating and using them.
- **Screen readers** are software applications that attempts to identify and interpret what is being displayed on the computer screen.
- The interpretation is then represented to the user with text-to speech, sound icons or a Braille output.
- The visually impaired people are using screen readers in combination with other assistive technologies, such as screen magnifiers.

Assistive Technologies

- **Voice recognition** software allows a person to simulate typing on a keyboard or selecting with a mouse by speaking into the computer.
- **Speech synthesizers** (often referred to as text-to-speech systems) receive information going to the screen in the form of letters, numbers and punctuation marks and then “speak” it loud. Using speech synthesizers allows blind users to review their input as they type.
- **Talking word processors** are software programs that use speech synthesizers to provide auditory feedback of what is typed.
- **Screen magnification** software allows a low-vision computer user to more easily read portions of the screen. They work like a magnifying glass. They enlarge a portion of the screen as the user moves the focus.
- **Refreshable Braille displays** provide tactile output of information on the computer screen. The user reads the Braille letters with his or her fingers and then, after a line is read, refreshes the display to read the next line.

Assistive Technologies

- **Braille embossers** transfer computer generated text into embossed Braille output. Braille translation programs convert text scanned in or generated via standard word processing programs into Braille, which can be printed on the embosser.
- **Large-print word processors** allow the user to view everything in large text without added screen enlargement.
- **Alternate input devices** and switches (including alternative keyboards, electronic pointing devices, sip-and-puff systems, wands and sticks, joysticks and trackballs) allow individuals to control their computers through means other than standard keyboard and pointing device.
- **Touch screens** allow direct selection or activation of the computer by touching the screen.
- **On-screen keyboards** provide an image of a standard or modified keyboard on the computer screen. The user selects a key with a mouse, touch screen, trackball, joystick, switch or electronic pointing device.

Assistive Technologies

- **Keyboard filters** include typing aids, such as word prediction utilities and add-on spelling checkers. These products reduce the required number of keystrokes. Keyboard filters enable users to quickly access the letters they need and to avoid inadvertently selecting keys they do not want.
- **Comprehension software** allows a dyslexic or a learning disabled computer user to see and hear text as it is manipulated on the computer screen.
- **Word prediction programs** allow the user to select a desired word from an on-screen list located in the prediction window. This help individuals increase written productivity and accuracy and increase vocabulary skills through word prompting.
- **Reading comprehension programs** focus on establishing or improve reading skills through ready-made activities, stories, exercises or games. These programs can help users practice letter sound recognition and can increase the understanding of words by adding graphics, sounds and possibly animation.

Assistive Technologies

- **Reading tools and learning disability programs** include software designed to make text-based materials more accessible for people who struggle with reading. Options can include scanning, reformatting, navigating or speaking text out loud. These programs help individuals who have difficulties seeing or manipulating conventional print materials; people who are developing new literacy skills or who are learning English as a foreign language; and people who comprehend better when they hear and see text highlighted simultaneously.
- **Screen review utilities** make on-screen information available as synthesized speech and pairs the speech with a visual representation of a word, for example, highlighting a word as it is spoken. Screen review utilities convert the text that appears on screen into a computer voice. This helps some people with language difficulties and impairments by giving them information visually and aurally at the same time

Software Accessibility

- The technical specification ISO/TS 16071, “*Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces*” was released in June 2003, by the International Organization for Standardization
- provides guidance on the design of software that is accessible and connects and interacts with supporting tools such as screen readers, Braille displays and screen magnification software
- **designed to improve the interface between the disabilities of users and software they use**, as well as to make optimal use of resources by incorporating those accessibility features for which there is a clear user requirement into the design early

Software Accessibility ISO Guidelines

- General guidelines
 - input/output alternatives
 - enable user to perform the task effectively with any single input device
 - provide user-preference profiles
 - enable user settings for timed responses
 - provide object descriptions
 - accessibility features should be easy to turn on and off
 - safeguard against inadvertent activation or reactivation of accessibility features
 - inform user of accessibility feature on/off status
 - enable persistent activation
 - avoid seizure-inducing blink rates
 - provide undo functionality
 - enable user control of time-sensitive presentation of information
 - clarify natural language usage

Software Accessibility ISO Guidelines

- Assistive technologies
 - use system-standard input/output
 - provide object labels
 - make event notification available to assistive technologies
 - make object attributes available to assistive technologies
 - present user notification in a relevant manner

Software Accessibility ISO Guidelines

- Key-board input configuration
 - enable sequential entry of multiple keystrokes
 - provide customization of delay before key acceptance
 - provide customization of same-key double-strike acceptance
 - provide customization of key repeat rate
 - provide customization of post-key-press delay of repeat onset
 - provide keyboard control of pointer functions
 - provide notification about toggle-key status
 - provide accelerator keys
 - reserve accessibility key-mappings
 - enable remapping of keyboard functions

Software Accessibility ISO Guidelines

- Software control of pointing devices
 - enable the adjustment of the location of button functions
 - enable multiple clicks with single key press and release
 - enable button hold with a single button press and release
 - enable delay of pointer-button-press acceptance
 - enable delay of pointer movement acceptance after mouse-down event
 - enable customization of multiple-click interval and target area
 - enable pointer speed and ratio adjustment
 - provide alternatives to chorded key presses

Software Accessibility ISO Guidelines

- Display fonts
 - enable font customization and legibility
 - adjust the scale and layout of objects as font-size changes
- Displays
 - enable users to customize viewing attributes
 - use text characters as text, not as drawing elements
 - provide access to information displayed in “virtual” screen regions
 - enable appropriate presentation of tables
- Color
 - Provide alternatives to the use of color as the sole source of information
 - provide color palettes designed for people who have visual impairments
 - allow users to create color pallets
 - use user-determined color settings
 - allow users to customize color coding
 - provide alternatives to coding by hue

Software Accessibility ISO Guidelines

- Audio output
 - enable audio customization
 - default frequency range for non-speech audio
 - provide specified frequency components for audio warnings and alerts
 - allow users to choose visual indication of audio output
- Errors and user notification
 - allow task-relevant warning or error information to persist
- On-line documentation and help
 - provide alternative methods for accessing on-line documentation and help
 - provide generic-help content
 - provide on-line documentation and help on accessibility features

Software Accessibility ISO Guidelines

- Customization of user preferences
 - enable customization of user preferences
 - enable customization of common interface elements
 - enable cursor and pointer customization
 - provide capability to use preferences across locations
- Window appearance and behavior
 - enable non-pointer navigation directly to windows
 - enable “always on top” windows
 - provide user control of multiple “always on top” windows
 - enable user choice of effect on input focus on window stacking order
- Keyboard input focus
 - provide focus cursor
 - provide keyboard navigation
 - provide navigation to task-appropriate groups of controls
 - retain input focus location

Web accessibility

- Web accessibility means that people with disabilities can use the Web.
- Web accessibility means that people with disabilities can perceive, understand, navigate and interact with the Web
- Web accessibility encompasses all disabilities that affect access to the Web, including visual, auditory, physical, speech, cognitive and neurological disabilities.
- While access to people with disabilities is the focus of web accessibility, it also benefits people without disabilities.

Web Content Accessibility Guidelines

- In 1999 the *Web Accessibility Initiative* (WAI) - a project of *World Wide Web Consortium* (W3C) - published the Web Content Accessibility Guidelines WCAG 1.0
- Provide several layers of guidance:
 - **Principles** – the four principles provide the foundation for the Web accessibility
 - **Guidelines** – 12 guidelines provide the basic goals that authors should work toward in order to make content more accessible to users with different disabilities; they are not testable, but provide the framework and overall objectives to help authors understand the success criteria and better implement the techniques
 - **Success Criteria** – for each guideline, testable success criteria are provided to allow WCAG 2.0 to be used where requirements and conformance testing are necessary such as in design specification, purchasing, regulation and contractual agreements; in order to meet the needs of different groups and different situations, three **levels of conformance** are defined: A (lowest), AA, and AAA (highest)
 - **Sufficient and Advisory Techniques** – for each of the guidelines and success criteria, the working group has also documented a wide variety of techniques; the techniques are informative and fall into two categories: those that are sufficient for meeting the success criteria and those that are advisory; the advisory techniques go beyond what is required by the individual success criteria and allow authors to better address the guidelines; some advisory techniques address accessibility barriers that are not covered by the testable success criteria
-

WCAG's Principles and Guidelines

- **Principle 1: *Perceivable*** – Information and user interface components must be presented to users in way they can perceive.
- *Guidelines:*
 - *Text Alternatives:* provide text alternatives for any non-text content so it can be changed into other forms people need, such as large print, Braille, speech, symbols or simpler language
 - *Time-based media:* provide alternatives for time-based media
 - *Adaptable:* create content that can be presented in different ways (for example simple layout) without losing information or structure
 - *Distinguishable:* make it easier for users to see and hear content including separating foreground from background

WCAG's Principles and Guidelines

- **Principle 2: *Operable*** – User interface components and navigation must be operable.
- *Guidelines:*
 - *Keyboard Accessible*: make all functionality available from a keyboard
 - *Enough time*: provide users enough time to read and use content
 - *Seizures*: do not design content in a way that is known to cause seizures
 - *Navigable*: provide way to help users navigate, find content and determine where they are

WCAG's Principles and Guidelines

- **Principle 3: *Understandable*** – Information and the operation of the user interface must be understandable.
- *Guidelines:*
 - *Readable:* make text content readable and understandable
 - *Predictable:* make Web pages appear and operate in predictable ways
 - *Input Assistance:* help users avoid and correct mistakes

WCAG's Principles and Guidelines

- **Principle 4: *Robust*** – Content must be robust enough that it can be interpreted reliably by a wide range of user agents, including assistive technologies.
- *Guidelines:*
 - *Compatible:* Maximize compatibility with current and future agents, including assistive technologies

MICROSOFT Accessibility

- Windows offers several programs and settings that can make the computer easier and more comfortable to use.
- The user can adjust the accessibility options, the display options, the mouse options, the keyboard options, the sounds options and also can use some programs that are assistive technologies.
- **Magnifier** is a help to people with low vision, but anyone can appreciate its ability to enlarge hard-to-see text and pictures. *Full-screen mode* magnifies the entire desktop, and *lens mode* zooms in on particular areas. Inside the magnifier window, buttons can be clicked and the input of the text can be done as usual
- **On-screen keyboard** offers the possibility to “keyboard without a keyboard” with a choice of several different input methods: clicking mode, hovering modes, and scanning mode. With a *touch screen*, the input of the text could be done by tapping directly on the screen. It also has word prediction, which speeds things up by offering a list of possible words as you type.
- **Narrator** can read on-screen text aloud and describe some events (like error messages), helping using the computer without a display.

Microsoft Active Accessibility

- *Microsoft Active Accessibility* 2.0 is an API devised by Microsoft so that accessibility aids can track what is going on inside the user interface of any software package that supports it.
- COM-based technology that improves the way accessibility aids work with applications running on Microsoft Windows. It provides dynamic-link libraries that are incorporated into the operating system as well as COM interface and application programming elements that provide reliable methods for exposing information about user interface elements.
- By using Active Accessibility and following accessible design practices, developers can make applications running on Windows more accessible to many people with disabilities.
- Active Accessibility is designed primarily for C, C++, and Microsoft Visual Basic developers.
- In general, developers need a moderate level of understanding about COM objects and interfaces as well as about Unicode.

ACCESSIBILITY EVALUATION

Accessibility Evaluation

- The two basic approaches to accessibility evaluation are:
 - Use of a software tool
 - Use of a human evaluator
- **Software tools** can quickly identify objective problems such as images without alt text, form elements without <label> tags, tables without headers, and so on.
- Some tools can even identify a few of the more subjective problems, such as suspicious alt text, suspicious link text, and text that might be more appropriate as headings.
- Several software tools can spider through web sites and produce reports for the entire site, including statistical analyses of the most frequent errors, a list of pages on which errors occur, and other useful information.
- Gathering this type of information without such software would be almost impossible.
- Software tools are incapable of determining whether content is logical, understandable, or intuitive.

Accessibility Evaluation

- Accessibility evaluation must take account of what it is like to experience the web with different senses and cognitive abilities and of the various unusual configuration options and specialist software that enable web access to people with particular disabilities.
- **When** should it be performed?
- “Test early, test often”
- Tacking on testing at the end of the development process has two risks:
 - Projects tend to run over-time and over-budget. Testing is often rushed, omitted, or ignored thanks to such pressures.
 - It is more work to fix problems discovered late in a process than to do things right from the start.
- to ensure quality and save time and money, accessibility evaluations should start right at the beginning of product design and be included in subsequent development iterations through to final delivery.

Accessibility Evaluation

- Before accessibility evaluation begins, you need to determine what the key requirements are for that project, given its environment, intended audience, and resources.
- Some requirements will be set by third parties like governments and clients
- External requirements
 - **Governments.** This typically takes the form of general legislation against discriminating against people with disabilities, rather than mandating a particular standard or enumerating precise conformance requirements. An important exception is when legislation enforces a particular standard for public sector ([Section 508](#) is a piece of US federal legislation, which mandates that websites produced for federal agencies must conform to at least a specific set of defined requirements)
 - **Customer policies.** For example, [Shell currently try to ensure their websites conform to the "Double-A" conformance level of WCAG 1.0](#), so if you were developing a website for Shell you would need to meet (at least) the same standard.
 - **Marketing utility.** Compliance with a particular standard, such as Section 508, might help sell a project to clients concerned about accessibility.
 - **Internal accessibility policies at your organization.** For example, projects produced by the BBC need to comply with the [BBC's Accessibility Guidelines v1.3](#).

Accessibility Evaluation

- There are basically two groups who conduct testing: experts and users.
- Expert testing is important because experts understand how the underlying web technologies interact, can act as a clearing house for knowledge about different user groups, and have the inclination to learn dedicated testing tools.
- User testing is crucial because users are the real experts in their own abilities and their own assistive technology.
- User testing can also reveal usability gaps between more and less technical users, and between people who are familiar with the web site in question (such as the expert testers themselves) and people who aren't (new users).

Accessibility Evaluation

- **Expert testing**
- There are four components to expert testing:
 - **Tool-guided evaluation:** where a tool looks for accessibility problems and presents them to the evaluator (this would include accessibility checkers and code linters).
 - **Screening:** where the expert simulates an end-user experience of the web site - often you don't need to look very far to find accessibility problems (you might do no more than load the page in your browser and notice the text is very hard to read).
 - **Tool-based inspection:** where the evaluator uses a tool to probe how the various bits of a web site are working together.
 - **Code review:** where the evaluator looks directly at the code and assets of a web site to scour for problems.

Accessibility Evaluation

- Semi-automated accessibility checkers:
 - <http://achecker.ca/checker/index.php>
 - <http://www.cynthiasays.com/>
 - <http://www.tawdis.net/ingles.html?lang=en>
- Related to different standards

Accessibility Evaluation

- **Screening and using end-user assistive technology**
 - Screening involves emulating the experiences of people with disabilities while testing.
 - using assistive technology to interact with a site or attempting to restrict one's abilities in some manner. For example:
 - Using a mouthstick to press keys while testing keyboard accessibility.
 - Viewing a page with the [Vischeck](#) simulator, which attempts to present the page, images included, as people with different forms of colorblindness see it.
 - Turning off a monitor while using a screen reader in conjunction with a browser.
 - Use of assistive technology needs to be accompanied by experience of how everyday users employ the technology and conclusions drawn from such use should ideally be confirmed with expert users

Accessibility Evaluation

- **User testing**
- No amount of developer inspection and screening can substitute for the raw clash between a user and a web site
-
- do not underestimate the benefits of doing even small-scale user testing.
- Recruiting testers:
 - [WebAIM Accessibility Discussion List](#).
 - [Web Accessibility Initiative Interest Group Mailing List](#): a forum for discussion of issues relating to Web accessibility.
 - [British Computer Association of the Blind mailing list](#): for discussing Information Communication Technologies (ICT) for visually impaired people.
 - [Magnifiers Yahoo! Group](#).
 - jfw@freelists.org: A mailing list for users of the JAWS screen reader.
 - [GW-Info](#): A mailing list for users of the GW Micro Window-Eyes screen reader.
 - [Dolphin software users Yahoo! Group](#).
 - [NVDA users mailing list](#).
 - [Thunder users mailing list](#).
 - discuss@macvisionaries.com: A list about use of OS X by the blind.
 - macvoiceover@freelists.org: Apple VoiceOver users.
 - [Blinux-list](#): A list about the use of Linux by people who are blind and visually impaired.
 - [GNOME Orca users](#).
 - [Ai Squared Forums](#): Including users of the popular ZoomText magnifier.
 - [Deaf-Macs Yahoo! Group](#): For deaf, hard-of-hearing, and Usher or deafblind Mac Users.
 - [deaf-uk-technology Yahoo! Group](#): Deaf-related technology discussion.

Accessibility Evaluation

- Choosing tasks:
 - It is very efficient to observe users simply exploring a website.
 - Try setting the users some specific tasks to accomplish.
 - Ask them what they think and listen to what they say.
 - Pay attention to what they do, because that may differ from what they say: stated preference is a poor guide to performance.
- Example: if you are testing a video sharing site for accessibility, do not begin by asking them if they can use particular controls (“That’s the volume slider. Can you adjust the volume?”). Instead, give them scenarios and ask them to achieve key user tasks. For example:
 - Browse videos and choose one to play.
 - Search for a video.
 - Upload a video.
 - Pause the video, play the video, mute the video, unmute the video, rewind the video and play it again.
 - Rate a video.
 - Share a video with a friend.

Accessibility Evaluation

- **Communicating the results of accessibility testing:**
 - document precisely what was evaluated
- If you tested conformance with a particular standard, be specific about exactly where conformance has succeeded and failed.
- Whenever raising a problem, make sure to put it in real, human terms and explain how the problem might adversely affect users.
- Describe how to reproduce the problem and test for its resolution.
- Suggest practical techniques for achieving conformance or improving accessibility.

Accessibility Evaluation

- Example:
- **Problem:** *The dropdown menu cannot be opened without using a mouse to hover over top menu items, and the keyboard focus disappears off-screen as you tab through the menu.*
- **How to reproduce:** *Open the page in your browser and attempt to reach a subitem of the menu using the keyboard alone.*
- **Explanation:** *Web navigation should be device-independent, so that users using devices other than mice—such as blind users or users with motor disabilities—can access content and functionality. Currently, such users can not access the items in submenus and sighted users using the keyboard may be confused when the focus indicator disappears.*
- **Conformance implications:** *Keyboard operability is a requirement for WCAG 1.0 and WCAG 2.0 Level “A” compliance (see WCAG 1.0 Guideline 9 and WCAG 2.0 Guideline 2.1).*
- **Suggested remedies:** *When JavaScript is not available, use a simple list of links to subpages for each sublist of navigation. On sub pages, present the main navigation followed by the sublist. When JavaScript is available, remove the sublist from the DOM and add sublists for each menu item on the click event, which can be triggered by keyboards, mice, speech recognition, and touch screens alike.*

Homework 😊

- Try navigating a complex site of your choice without using the mouse. What difficulties do you encounter? How could the developers of the site help you?
- Turn off CSS and do your normal browsing for a day. What problems do you encounter?
- Turn off JavaScript and do your normal browsing for a day. What problems do you encounter?