Making Software Accessible to the Disabled

Tim Ginn

CISC 497, Section B

R. Linley

December, 2007

Making Software Accessible to the Disabled

Designing software is a complex task often requiring taking into account the interests of a variety of stakeholders. Unfortunately, there exists a group of people who are often completely forgotten in the design of systems. People with disabilities can have different requirements for software depending on their circumstances. To properly accommodate people with disabilities not only must an appreciation of their unique challenges be gained; but, also a mastery of certain development techniques, tools and methodologies must be attained. A good way of examining this process is to break down the topic into a series of questions:

1. What types of disabilities might people have that can affect their use of software?
2. What problems do people with disabilities have with current software?
3. What adaptive technologies might disabled people use with a computer?
4. What techniques and tools can a software developer use to accommodate people with disabilities?
5. What side effects are there to making software accessible?

This literature review will focus on answering these questions and will bring to light some of the social, ethical and legal issues surrounding the design of accessible software.

What types of disabilities might people have that can affect their use of software?

Mark Pilgrim's Dive Into Accessibility Day 1-5 introduces the types of disabilities that people may have through the introduction of several characters each having a well-defined back story that

defines them as people rather than focusing on their disability. These characters include Jackie (who lost her sight in a car accident a few years ago and cannot read braille), Michael who has protanopia (a form of colour blindness), Bill who is unable to move his right arm, has shakes in his left arm and after a stroke sometimes loses track of what he's looking at, Lillian who is an immigrant from China with limited vision and difficulty reading English and Marcus who was blind from birth and can read braille. Eight percent of American males are colour blind and an estimated 54 million people in America have a disability (Thatcher, 2007, Other Forces for Accessibility section, para. 5).

What problems do people with disabilities have with current software?

Problems that people with disabilities encounter in their use of computer software range from minor announces to critical flaws that prevent any use of the software at all. Minor annoyances include things like some images that are not essential to understanding the meaning of something not containing text alternatives for people without sight, font sizes that are a little bit too small for a person with poor vision to read clearly (but not small enough that reading with squinting is impossible) and so on. One example of a critical flaw is the use of colours that are indistinguishable from one another for someone with a certain type of colour blindness combined with the requirement to differentiate based solely on colour (for example, click the “red” button or the “green” button). Another example is the use of animations (for example flashing advertisements) that can be extremely distracting for someone with a cognitive disorder preventing the comprehension of other material on the page. A final type of critical

flaw is using only a single media to present multimedia information (for example, only an image with no descriptive text that could be read aloud or having content available as only audio with no closed captioning) (Pilgrim, 2002, Day 1-5 sections; Thatcher, 2007, Other Forces for Accessibility section, para. 1-7).

What adaptive technologies might disabled people use with a computer?

Someone who is blind might make use of a screen reading program such as Fire Vox or JAWS which use a synthesized voice to read the content of a page aloud (Chen, 2006; Freedom Scientific, 2007). Someone who is blind and can read braille might make use of a braille output device which are available in many adaptive technology facilities at universities and allow someone to read lines of text one by one (Queen's University, 2006; University of Washington, 2007). A blind person will typically learn to use many keyboard shortcut keys for the software they use regularly. A person with low vision may use software to magnify parts of the screen or may run their computer with a large monitor at a low resolution. Someone with mobility difficulties or repetitive strain injury could use alternatives to a standard mouse and keyboard such as a headgear that allows the computer to move a mouse cursor based on head movements, a track ball, touch pad, foot pedals, a joystick, voice input systems, predictive text, or keyboards with large keys or different layouts (University of Washington, 2007). A blind musician might use a combination of a screen reading program, special music software such as GOODFEEL, a braille embosser to print music sheets and a braille display and note-taker resembling a laptop to compose, produce and revise music (Dancing Dots, 2005). Unfortunately, some people with disabilities who could

benefit from adaptive technologies may not use any of them due in part to a large financial cost of many accessibility tools (often thousands of dollars in specialized software alone), unavailability in developing nations, using a public computer that does not have some of these technologies and some people with disabilities simply never have them properly diagnosed preventing useful technologies from being recommended in the first place.

What techniques and tools can a software developer use to accommodate people with disabilities?

A wide variety of techniques and tools exist to help developers accommodate people with disabilities. The first and most obvious technique is to conduct extensive user testing with people with disabilities. This is unfortunately not always realistic due to time constraints, budgetary constraints and the difficulty in finding people with each of the disabilities that might need to be accommodated for. For web-based applications, some developer tool kits such as Google's AxsAJAX provide well-tested methods of making dynamically changing pages work well with adaptive technologies to effectively update the (Chen & Raman, 2007). For desktop software, using typical best practices such as including keyboard shortcuts for menu items can go a long way to making the program significantly easier to use for people with some disabilities.

A tool called Fangs allows sighted developers to view a page as a blind person would hear it read by the JAWS screen reader software. Fangs is not appropriate for a blind person to use because it does not perform text to speech synthesis, instead it just displays the text that would be read aloud; but, doing

this can be very effective for helping a sighted developer test how a screen reader would cope with a page (Krantz, 2006). The use of alternative text when images are used to convey information is helpful for people who have difficulty seeing the images, cannot see the images, or who have chosen to turn off image loading perhaps because they have a slow connection to the internet. A good technique for testing websites for issues with alternative text and a layout of information that makes sense is to use a text only browser such as Lynx to browse the website normally (Lynx, 2004).

A good developer of any software should have an understanding of Section 508 published by the United States Government (U.S. General Services Administration, n.d.). While Section 508 requirements technically only apply to technology to be purchased by the United States Federal government, the recommendations contained within it are very well researched and it serves as an effective reference for designing accessible systems (both software and hardware) in a more general sense, too (Thatcher, 2007, What is Section 508? section, para. 4-5). There are several automated accessibility testing tools available such as Watchfire WebXACT which can inspect a given website for compliance with the recommendations of Section 508 and the World Wide Web Consortium Website Accessibility Initiative's Web Content Accessibility Guidelines (Watchfire Corportation, 2004). Despite automated testing, some checkpoints on the accessibility guidelines can only be verified manually such as ensuring that the alternative text descriptions provided accurately represent the content of the image they are replacing (World Wide Web Consortium, 2007).

Chris Pederick's Web Developer Toolbar provides the ability to easily disable image loading, display access keys that can be used as shortcuts, disable cookies and Javascript, view pages in a linear

order like a screen reader might read them, outline page structures such as headings, perform automated accessibility validation tests and more in one convenient and time-saving package (Pederick, 2007).

To address colour perception problems, it is often useful to have an understanding of the contrast between colours. Tools exist to inspect a given colour scheme to determine whether or not the colours meet the published guidelines of the World Wide Web Consortium (Snook, 2007). Another thing that software developers can do is read and follow articles in their field which address specific problems and strategies to address them for example proper design of hyper-text markup language forms, ways to obfuscate e-mail addresses, the proper technique for explaining acronyms and other abbreviations, and how to make a map usable (Happy Cog, 2007).

Internationalization of software is something that can be tricky even before taking into account disabilities; but, mistakes can make things far worse for people with disabilities For example, screen readers might attempt to read text using the wrong language settings which can at best result in an incorrect accent and poor pronunciation and at worst a completely incomprehensible and confusing dialog from which no useful information at all can be gathered . There are some things that can be useful in preventing international users from ending up in the wrong section of a website or software such as avoiding using the flag of a country that speaks a language to select that language because flags are often used to select regions and using them for languages can be misleading especially if the explanatory text saying to select a language isn't itself in a language the user knows (Pilgrim, 2002, Day 7 section).

Where software uses animations, sounds or videos these animations shouldn't play automatically if they are in a position where they would interfere with a screen reader or distract someone with a certain

kind of cognitive condition. If animations, sounds or videos are used,there should be a clear and easy way to stop them from playing. A website which hosts many pages that violate these principles of properly handling animations/videos/music is MySpace where users have profile pages, many of which being playing music automatically when loaded forcing someone who would like to stop the music to search through the page to find a stop button that's not consistently placed or styled amongst different profile pages.

What side effects are there to making software accessible?

There are a number of positive side effects from making software accessible. Often accessibility features result in more features that would be used by power-users (such as additional keyboard shortcuts). Another benefit, particularly relevant to web-based software is the benefit that virtually all current search engines behave like a blind user and will read and make use of alternative text for images and the order of content on pages in determining rankings (Happy Cog, 2007, High Accessibility is Effective Search Engine Optimization section). By accommodating for low vision users through the use of adjustable resolutions a convenient side-effect is that users of portable devices such as personal digital assistants, electronic ink book readers such as Amazon's Kindle and cell phones will typically have a better experience because the resolution of their displays are low relative to a typical desktop monitor or they are limited to monochrome in the case of Kindle or reduced colour sets in the case of some cell phones/PDAs (Happy Cog, 2007, Pocket-Sized Design: Taking Your Website to the Small Screen section). Avoiding the pitfalls of automatic animations/videos/sounds can provide benefits for any user

who browses the Internet with their speakers turned on who may not want to listen to a particular thing or may be listening to a music CD or watching a video in the background while using any given piece of software. In most cases, by implementing additional accessibility measures there is no effect on the software experience for people with perfect vision and hearing.

References

Chen, C. (2006). Fire Vox: A Screen Reading Extension for Firefox. Retrieved December 1, 2007, from Fire Vox website: http://firevox.clcworld.net/

Chen, L & Raman, T.V. (November, 2007). Introducing AxsJAX – Access-Enabling AJAX. Retrieved December 1, 2007, from Google Code Blog: http://google-code-updates.blogspot.com/2007/11/introducing-axsjax-access-enabling-ajax.html

Dancing Dots (2005). GOODFEEL Configuration for Blind Users. Retrieved December 1, 2007, from Dancing Dots website: http://www.dancingdots.com/main/blindsample.htm

Freedom Scientific (2007). JAWS® for Windows® Overview. Retrieved December 1, 2007, from Freedom Scientific website: http://www.freedomscientific.com/fs_products/software_jaws.asp

Happy Cog (November, 2007). A List Apart: Topics: User Science: Accessibility. Retrieved December 1, 2007, from A List Apart website: http://www.alistapart.com/topics/userscience/accessibility/

Krantz, P. (October, 2006). Fangs – the screen reader emulator. Retrieved December 1, 2007, from Standards schmandards website: http://www.standards-schmandards.com/projects/fangs/

Lynx. (February, 2004). Lynx Information. Retrieved December 1, 2007, from Lynx website: http://lynx.browser.org/

Pederick, C. (June, 2007). Web Developer :: Firefox Add-ons. Retrieved December 1, 2007, from Firefox Add-ons web site: http://addons.mozilla.org/extensions/moreinfo.php?id=60

Pilgrim, M. (2002). Dive Into Accessibility. Retrieved December 1, 2007, from Dive Into Accessibility website: http://diveintoaccessibility.org/

Queen's University (September, 2006). Facilities & Equipment in Stauffer Library- Resources in the Stauffer Library Adaptive Technology Centre. Retrieved December 1, 2007, from Queen's University Library website: http://library.queensu.ca/websrs/facilities-Adap-Tech-Lab.html

Snook, J. (Feburary, 2005). Colour Contrast Check. Retrieved December 1, 2007, from snook.ca website: http://www.snook.ca/technical/colour_contrast/colour.html

Thatcher, J. (November, 2007). Web Accessibility – Section 508. Retrieved December 1, 2007, from JimThatcher.com website: http://jimthatcher.com/webcourse1.htm

U.S. General Services Administration (n.d.). Section 508: The Road to Accessibility. Retrieved December 1, 2007, from Section 508 website: http://www.section508.gov/

University of Washington (September, 2007). Resources in the Access Technology Lab. Retrieved December 1, 2007, from Computing and Communications Access Technology Lab website: http://www.washington.edu/computing/atl/DOCS/atl2.html

Watchfire Corporation (2004). Watchfire WebXACT. Retrieved December 1, 2007, from Watchfire WebXACT website: http://bobby.watchfire.com/

World Wide Web Consortium (2007). Website Accessibility Initiative (WAI). Retrieved December 1, 2007, from World Wide Web Consortium website: http://www.w3.org/WAI/