

Process Report for Accessibility Resource Adaptation

Jon Witts

Abstract

In this report I will discuss the process I undertook whilst converting the University of Huddersfield's "Student Regulations" handbook from Microsoft Word format into a more accessible resource, using eXtensible Hyper Text Mark-up Language (XHTML). The adapted resource is available on the Internet and can be viewed by any user with an Internet connection and any web browser software. My main focus was to produce an adaptation of this document that would be more accessible for users that rely on screen reading software to navigate the Internet and to use computers.

Discussion

The resource I chose to adapt was the University's "Students' Regulations". This document is available from the university website as a word document. It is one hundred pages long and contains a wealth of important information for all students of the University. The target group of users that I chose to try and increase the accessibility of this resource for were students who had a visual disability and those users who may use screen reading software to help them to read documentation.

When creating a document in Microsoft Word, it is possible to have the software create a Table of Contents automatically, providing that the document utilises the correct formatting within Word. A well formatted document will do more than just allowing a table of contents to be created. It will allow screen reading software to make sense of the document for the user. For example, If you have a sub-heading of a main section of the document that you have formatted as a level two heading, the software will announce the heading as a level two heading. However, if you have just changed the font size and made the text bold, the software will most likely just read the text, not letting the user know that the information was meant as a level two heading. When using Microsoft Word to create a document, it is possible to use these styles to correctly format your document. However, the majority of users do not use these style options in Word, and in effect create inaccessible documents.

The original document was a series of points or items, most with sub points, making up each section of the list of regulations. The document was, however, formatted as a series of paragraphs and bullet points, with no common method of presentation between the separate sections. To make this resource more accessible, I needed to ensure that there

was a common format of presentation, and that the format I chose made sense semantically.

As well as the presentational methods within the original document that made it inaccessible to users, the software used, (Microsoft Word) also creates issues of inaccessibility itself. These issues of inaccessibility extend beyond the realm of visually impaired users. The software creates files with a very large file size compared to the text only or HTML equivalent, which can cause issues for any user that is using a slow Internet connection to try and download the document. Perhaps the biggest inaccessibility issue with producing documentation in Microsoft Word is that you are requiring all users to purchase, install and run a piece of proprietary software, just to read your document. Many computer users do not have Microsoft Word installed and many more have no wish to use proprietary software on their computer systems. If you have no choice other than releasing your document in Microsoft Word format, you should at the very least provide a link to a "Word Viewer" piece of software, that the user could download to enable them to read the document. (Whiting, J. 2006) However this is still not an entirely acceptable solution. By requiring the user to download a piece of software to read your document, you are again falling foul of the problem of enforcing users to download large files. Also most "Word Viewers" are only compatible with Microsoft Windows Operating Systems. According to OneStat, Microsoft holds around 97% of the operating system market, (ONESTAT.COM 2006) however most computers that are shipped with Microsoft Windows do not come with Microsoft Word by default.

I chose to use XHTML to adapt this resource and make it more accessible to users with visual impairment of whatever degree. The use of XHTML would also make the resource more accessible for all users too. There would be no requirement to use a proprietary piece of software to view the resource. By posting the XHTML version of the document on-line, you know that any user who can access the Internet will be able to access your documentation, all they would need is a web browser and an Internet connection.

The latest standards and requirements for correctly coded XHTML revolve around the idea of using the correct tag for the type of information that the document is trying to convey. For example, if the section of text is a second level heading, it should be tagged using the <h2> tag. Likewise, if an amount of information is being conveyed as a series of list items, then they should be tagged as an list, either ordered or unordered. This method of coding will ensure that the document not only displays correctly in all standards compliant web browsers, but that it will also make sense to screen reading and other semantic software applications.

A correctly written XHTML document will always be more accessible to users with visual impairments than a Word document, due to the issues stated above.

To begin to adapt the University's Word document to an accessible XHTML document, I first copied all of the text from Word into a blank html document in DreamWeaver. This removed all of the formatting that had been applied in Microsoft Word and left me with only the plain text of the document. Being the original document was around 100 pages long, I decided to break it down in to smaller pages in my XHTML adaptation. The original document was structured in twelve sections and four appendices. The document also started with an introduction, followed by a list of changes since the last time the document was updated and then the contents of the document. I decided to use the existing sections as the point at which I would split my XHTML version.

I decided that it would make the resource clearer for users if the contents and introduction were at the beginning of the document, and that the contents linked directly to the page. I took the section that was the list of changes, and made them into a page of their own, placing a link to them on the introduction page. I formatted this list of changes as an ordered list, that had sub list when necessary. Each of the changes that were displayed in this list, linked back to the section in the document that the change was indicating. This then left me with the introduction and contents as the first page for the document. I formatted the contents as an ordered list, with each item in the list linking to the section of the document that it referred to.

I then developed a common navigation for each page that I placed at the top of each page. A common navigation theme is important for accessibility too. The navigation of web pages allows all users to find their way around the document. I chose to place a simple two link menu at the top of each page. The first link led back to the contents page, and the second link in the menu navigated the user to the next chapter in the document. In line with the idea of the semantic ideals behind XHTML, my navigation was coded as a list. The reason for this is because navigation is, in essence, a series or list of links, so presenting the navigation in this format makes the most sense semantically. (Brewer, D. 2007)

As I wrote each page, I used a plug-in to Mozilla Firefox to verify that the code was correctly written. This plug-in is called "HTML Validator" (Gueury, M.), it allows you to see with an icon and text display in the bottom right of the browser window, how many errors and warnings the web page you are looking at has. It is based on the World Wide Web Consortium's (<http://www.w3.org/>) algorithms for checking web page coding. I ensured that every single page passed with no errors and no warnings before I considered it to be a completed page. Once I had uploaded the pages to my personal website, I also checked them through the World Wide Web Consortium's own HTML Validator at

<http://validator.w3.org/> . This service will tell you if there is an error in your coding and also give you pointers towards good web coding practice.

The document that I have developed is purely textual. There are no pictures, graphics, sound or video. This is an intentional decision in terms of its accessibility. The original document did not have any graphics, so I feel that it will not make it any more accessible by adding graphics to the document. The other thing that I did not do with the resource was to apply any style sheets to it. With the semantic nature of XHTML, you often get very plain looking documents, such as this adapted resource. The World Wide Web Consortium has developed a technology to allow developers to create XHTML documents that remain semantically clear whilst also allowing developers to control the visual appearance and layout of the web page. This technology is called Cascading Style Sheets (CSS) and sits alongside XHTML in creating graphic and design rich, accessible web sites. The reason I chose not to add any CSS to this resource was because of who my target audience were. For visually impaired users the graphological and presentational devices in a web page are of less importance than the content and structure of the document, and such users are unlikely to find such extra information useful when navigating through a document. By using pure XHTML, it enables the user to control the style and feel of the document by applying their own user style sheet. This would mean that a user who had dyslexia and found that a red background and a sans-serif font helped them read text better, could define a personal style sheet in their web browser that changed all background colours to red and all text to a sans-serif font. (Jacobs, I. and Brewer, J. 1999) As the resource I have developed has no style sheet associated to it, if you view it in a web browser that does not have a personal style sheet running, you will see the document with the web browser's default style. For example, many web browsers default to Times New Roman for their font; however if you change the default settings for your web browser, the resource I developed will reflect those changes.

Recommendations

To develop this resource further it will be necessary to design a series of style sheets targeted to specific users, i.e. dyslexic users could have had a choice of background colours for the resource and a choice of fonts; users with low vision could have had the option to increase the font size through the web page. Both of these would have been possible using a different set of CSS rules for each version and applying them to the XHTML version of the resource.

To really test its accessibility it would have been great to trail the resource with a user who relies entirely on screen reading software to navigate around computers, to see how easy they found using the information in the adapted resource.

Conclusion

I started with a very lengthy document that had been written by a large number of people and adapted over a long period of time. This had led to a number of issues with its format and presentation that had made it very inaccessible to a number of users, particularly those with visual impairments. I manipulated and restructured the content of this resource to develop a more accessible version of this document using XHTML code. The reason that I chose this language was mainly because it is focused on the correct semantic use of mark-up for content and in such provides a very accessible resource for anyone using a screen reading piece of software. I used two sources to verify my code to ensure that it was written correctly, these were the plug-in built into Mozilla Firefox and the World Wide Web Consortium's on-line verification service.

Whilst the resource is successful in being adapted into semantically correct XHTML, it could have developed further by incorporating different style sheets for users with different impairments. However in leaving the document as XHTML with no CSS, it leaves the resource open to be adapted further by the end user who is reading the document, to suit their own personal preference. The resource will also be usable by most other agents, i.e. mobile phones and PDAs, as there is no style sheet or table layout that would make it unusable on the smaller screens that these devices have. The text and menus within the resource will gracefully resize to practically any screen size.

Bibliography

1. ACCESSIT (2002) *How accessible are Microsoft Word documents?* [Online] Available: <http://www.washington.edu/accessit/articles?266> [Accessed 15th March 2008]
2. Brewer, D. (2007) *The importance of using lists for navigation* [Online] Available: <http://dustinbrewer.com/the-importance-of-using-lists-for-navigation/> [Accessed 16th March 2008]
3. Gueury, M. (Unknown) *HTML Validator for Firefox and Mozilla* [Online] Available: <http://users.skynet.be/mgueury/mozilla/> [Accessed 12th March 2008]
4. Gugerty, J. (2002) *Accessibility Features of Microsoft Word*. [Online] Available: <http://www.cew.wisc.edu/accessibility/tutorials/MSWordFeatures.htm> [Accessed 15th March 2008]
5. Jacobs, I. and Brewer, J. (ed.) (1999) *Accessibility Features of CSS* [Online] Available: <http://www.w3.org/TR/CSS-access> [Accessed 13th March 2008]
6. MASSACHUSETTS INSTITUTE OF TECHNOLOGY (2005) *MIT Web Accessibility guidelines* [Online] Available: <http://web.mit.edu/atic/www/accessibility/developweb.html> [Accessed 16th March 2008]
7. Nielsen, J. (2005) *Top Ten Web Mistakes of 2005* [Online] Available: <http://www.useit.com/alertbox/designmistakes.html> [Accessed 15th March 2008]
8. ONESTAT.COM (2006) *Microsoft's Windows dominates the OS market on the web*. [Online] Available: http://www.onestat.com/html/aboutus_pressbox46-operating-systems-market-share.html [Accessed 15th March 2008]
9. Pinsonnault, C. (Unknown) *Web Site Navigation Best Practices* [Online] Available: <http://ezinearticles.com/?Web-Site-Navigation-Best-Practices---Navigation-Design-Guidelines&id=453706> [Accessed 16th March 2008]
10. Stallman, R. (2008) *We Can Put an End to Word Attachments*. [Online] Available: <http://www.gnu.org/philosophy/no-word-attachments.html> [Accessed 17th March 2008]
11. WEB ACCESSIBILITY CENTRE (Unknown) *WAC Tutorial: Best Practices* [Online] Available: <http://www.wac.ohio-state.edu/tutorials/bestpractices/default.htm> [Accessed 16th March 2008]
12. Whiting, J. (2006) *NCDAA – Microsoft Word Factsheet*. [Online] Available: <http://ncdae.org/tools/factsheets/word.cfm> [Accessed 17th March 2008]