

Towards an Accessible Learning Management System

Tristan Kalnins-Cole

Web Engineering Group

School of Electrical and Information Engineering

University of Sydney, NSW 2006, Australia

tristankc@optus.net

Dorian Peters

CoCo Research Centre

School of Development and Learning

University of Sydney, NSW 2006, Australia

d.peters@edfac.usyd.edu.au

Keywords

Accessibility, accessibility compliance evaluation, learning management systems, e-learning platforms, .LRN

Abstract

One of the most consistent problems with modern learning management systems is their failure to comply with international standards for accessibility. Accessibility guidelines ensure that students can use educational websites, regardless of physical disability. When compliance is not met, students with disabilities may be excluded from online learning. This is a problem, not only because of the clear moral obligation educational institutions have to provide equal access to learning for all students, but also because accessibility compliance has been mandated by law in many countries including the U.S., Canada, the U.K. and Australia. Yet, when educational institutions turn to the vendors of, even the most popular, learning management systems, they are faced with vague statements of commitment but no actual compliance with international standards. One of the key advantages of open source systems is the user community's freedom to improve the software and make changes as necessary. In this paper we describe an accessibility evaluation, the first step to compliance, for the open source elearning platform .LRN. We provide the results of this evaluation, and consequent recommendations for development changes that would make .LRN compliant with the W3C international standards for accessibility.

Introduction

Universities are increasingly turning to the online and blended delivery of courses. Simultaneously, the number of people with disabilities is increasing as the population ages (Vanderheiden, G.C. 1990). The percentage of mature age and adult students, or 'life-long learners' is also increasing (DETYA 2000). Already, studies show that one in five people report having some form of disability (Loiacono, E.T. 2003). The need for accessible services is undeniable and growing. Fortunately, Internet accessibility legislation has already been well established in numerous countries (W3C 2005) such as the UK (United Kingdom 1995), Australia (Australia, 1992), Canada (Treasury Board of Canada 2004) and the USA (United States of America 2002). In the United States, Section 508 of the Americans with Disabilities Act, which requires federal agencies to make their electronic and information technology accessible to people with disabilities, was mandated in 1998. Yet, seven years later, educational software vendors are still struggling to comply with the very minimum international guidelines.

The international standards for accessibility are provided by the World Wide Web Consortium's (W3C) Web Accessibility Initiative. The W3C provides a set of guidelines that ensure access by assistive technologies such as screen readers and alternative navigation devices. Most government legislation on accessibility is based on these guidelines. The most popular learning management systems including, WebCT and Blackboard, provide corporate statements of commitment to accessibility, but still no verified compliance. WebCT states that they are working towards compliance: "WebCT is working with leaders in the accessibility field to ensure that our software complies with both industry standards and federal guidelines for accessibility...We are committed to building our software towards the World Wide Web Consortium's Web Accessibility Initiative Standards (WAIS, W3C)." (WebCT, Inc. February 2005) Blackboard has gone a bit further and demonstrates compliance with the U.S. section 508 guidelines, but not yet with the W3C international guidelines: "We are working with leaders in the accessibility field to contemplate industry standards and federal guidelines for accessibility" (Blackboard, Inc. 2005). We have yet to see a statement of compliance to international standards from either vendor.

One of the unique advantages of working with open source software is the freedom the community has to make changes and improvements to the software, as they become necessary. Proprietary software brings with it dependence on a third party. The community must wait until the vendor chooses to make needed changes and then release them. If vendor priorities are focused elsewhere, changes like accessibility compliance, may take months or years to be released. Therefore, we sought to take the first step in making .LRN, an open source learning management system, compliant with the W3C international guidelines. The first step to compliance is evaluation -- when a system is already in place, it is necessary to determine where accessibility obstacles occur and to document them. This evaluation can then form the basis for a development plan to fix these obstacles and achieve compliance. The evaluation process we have used can also be applied to other open source elearning systems.

In the first section, we describe the systematic accessibility evaluation process as it was carried out on a default installation of the .LRN elearning platform. In the second section, we present the results of this evaluation and summarise the guideline infringements. In the last section, based on these results, we provide a set of recommendations that, if implemented, will lead to a fully compliant elearning management system.

1 Evaluation Process

There are essentially two methods to accessibility testing: manual and automated. It is entirely possible to test a web site manually without the assistance of automated tools. However for large web sites with hundreds of pages this is infeasible. Moreover it is impossible to completely test a web site for accessibility using only currently available automated tools.

The W3C Web Content Accessibility Guidelines (WCAG) are broken down into three priority levels. These are known as priority 1 checkpoints (the minimum set of guidelines that a site has to meet to be compliant), priority 2 checkpoints and priority 3 checkpoints.

1.1 Automated Testing

The purpose of automated testing tools is to determine whether the web site being tested *may* comply with accessibility guidelines. These tools can detect some accessibility infringements such as images without alternate text descriptions. They can also detect *possible* infringements and instruct the tester to take a closer look. However, if the automated testing tool finds nothing wrong with a web site, that does not necessarily mean the site is accessible. Automated testing is essentially an initial phase in the testing process. Once completed the manual testing can begin.

There are a number of testing tools available today. The most well known tool is called Bobby (Watchfire Corporation, 2005) which tests for all WCAG priorities. The creators of Bobby provide a free evaluation version, which works the same as the full version except that it runs on a single web page and allows the tester to test at a rate of one page per minute. An alternative to Bobby is LIFT (UsableNet, Inc. 2004). Like Bobby, LIFT provides an evaluation version but only checks a subset of all the rules provided in the full version. The third and final automated tester investigated was Cynthia Says (ICDRI 2t al. 2005). It too provides a free online evaluation version with run times identical to Bobby (one page per minute). However, Cynthia allows the tester to choose which WCAG compliance levels to test for and can even emulate a range of different browsers. A number of different tools are compared by Arch (2000) and Kirchner (2002 and 2003). Because budget was more of a constraint than time for this project, Bobby and Cynthia Says were selected for further evaluation, as they are the only fully functional testing tools available at no cost.

The two chosen testing tools were evaluated using a sample page (in this case the WCAG website was used). Both tools found that the site did not violate any accessibility rules that can be checked automatically. However, both tools reported that there were issues that needed to be checked manually (such as screen flicker and style sheet usage). Although both tools gave similar results, their information is displayed in very different ways.

After evaluating these automated testing tools and reading other available evaluations (Thatcher, J. 2005b, Thatcher, J. 2005c) it became clear that limiting testing to a single tool would be poor practice. Since the reports displayed by Cynthia and Bobby show similar information in a very dissimilar fashion, running both tools during the testing process would provide more accurate results. Hence, both Cynthia and Bobby were selected as the automated testing tools to use in the evaluation.

1.2 Manual Testing

The second phase in a complete accessibility evaluation is manual testing. There is essentially only one way to manually test for accessibility. That is, being thorough and using good judgment. Different people have different variations of essentially the same technique. Firstly, the guidelines to be checked should be clearly set out. The WCAG are formatted in easy to read checkpoints with examples. The W3C also provide checklists to facilitate this process (Cynthia Says uses the W3C checklist layout when producing its report).

Well-known accessibility expert Jim Thatcher (2005d) describes manual testing as “human judgment”. He cites examples where human judgment is necessary such as deciding whether alternate text for images is appropriately descriptive. “The judgment comes in when evaluating the quality, and possibly existence, of the text equivalents. Does a text string serve as a text equivalent for the non-text element?” Ford (2004) raises the point that when testing for accessibility, it is ideal for the tester to use the site in the ways more closely related to how users with disabilities will be using the site: “You cannot test a feature for keyboard accessibility if your mouse is only inches away from you and still plugged in.”

The W3C have released a useful document describing a testing methodology (W3C 2002). Their method consists of three major steps. Firstly a preliminary study is carried out. This entails the use of automated tools to detect any obvious failures. They also include a brief manual test where the tester performs tasks such as using the website with different browsers, turning off images or using assistive technology to access the site. Secondly a comprehensive study is carried out. The idea of the preliminary study is for the tester to get a feel for the site’s current level of compliance. The comprehensive study identifies the exact current level of compliance and/or why it fails to meet any or other levels. The comprehensive study includes studying the HTML code of the page and having multiple users evaluate the site. Lastly a review of procedures for ongoing monitoring is carried out to ensure that future changes to the site will not invalidate its current compliance level.

1.3 Testing Methodology

After considering the many different existing methods of manual and automated testing it was concluded that a modified version of the W3C process would be most appropriate for this evaluation. The evaluation was to be more detailed than a preliminary analysis, but time and resources prohibited a full conformance evaluation (which includes user testing). Accordingly a hybrid methodology built from certain parts of the W3C process was devised. This hybrid process includes all preliminary testing and some aspects of the conformance evaluation. The modified and completed methodology that was used for testing .LRN is provided below.

1. Select a representative sampling of different kinds of pages from .LRN.
 - a. Identify and disclose the target conformance level of WCAG 1.0.
 - b. Identify and disclose a page selection for manual and user testing which includes at least one of each different type of page on the site, and all pages on which people are more likely to enter your site. NOTE: there are special considerations for web sites with database-driven dynamically generated content.
2. Automated evaluation
 - a. Use two general accessibility evaluation tools (Bobby and Cynthia) and note any problems indicated by the tools.
3. Manual evaluation
 - a. Examine page selection using relevant WCAG 1.0 checkpoints.

- b. Use Mozilla 1.0 and Internet Explorer 6.0 to browse .LRN while doing the following:
 - i. Turn off images, and check whether appropriate alternative text is available.
 - ii. Turn off the sound, and make sure audio content is still available through text equivalents.
 - iii. Use browser controls to vary font-size: verify that the font size changes on the screen accordingly; and that the page is still usable at larger font sizes.
 - iv. Test with different screen resolution, and/or by resizing the application window to less than maximum, to verify that horizontal scrolling is not required.
 - v. Change the display colour to grey scale and observe whether the colour contrast is adequate.
 - vi. Unplug the mouse and tab through the links and form controls on a page, making sure that you can access all links and form controls, and that the links clearly indicate what they lead to.
 - c. Read over the pages: is the text clear and simple?
 - d. Use a text browser (Lynx) and examine .LRN while answering these questions
 - i. Is equivalent information available through the text browser as is available through the GUI browser?
 - ii. Is the information presented in a meaningful order if read serially?
4. Summarize results
- a. Summarize the types of problems encountered, as well as best practices that should be continued or expanded.
 - b. Indicate the method by which problems were identified, and clearly state that this was not a full conformance evaluation
 - c. Recommend follow-up steps, including full conformance evaluation, which includes validation of markup and other tests, and ways to address any problems identified.

2 Results

2.1 Results of Automated Testing

One obstacle that presented itself at the start of the automated testing phase was the inability to use automated validators on pages that require secure log in. To gain access to .LRN a user must login, but the evaluation versions of Bobby and CynthiaSays do not have login functionality, and thus, were unable to test any pages other than the login page.

The solution was to use a graphical web browser (specifically Mozilla 1.0), navigate through all selected pages, and save static copies of each page to the hard drive. Additionally, since the evaluation versions of Bobby and CynthiaSays are not downloadable they could not be run on the locally saved pages. Thus the static pages were uploaded to a web server and could then be successfully tested. The results of the automated tests are displayed below.

2.1.1 Priority 1 Automated Testing Results

Initially, the sites were tested for the W3C's list of first priority checkpoints. This is the minimum number of guidelines with which a site has to comply in order to be deemed accessible.

For priority 1 checkpoints, the results were extremely satisfactory. Of all checkpoints only one was unmet by the system. This was the requirement for all images to have alternative text descriptions. All pages were found to have violations of this checkpoint.

Unmet priority 1 checkpoints:**AltText (Checkpoint 1.1)**

Provide alternative text for all images. All images should contain a short alternative text description that represents the function of the graphic.

2.1.2 Priority 2 automated testing results

All 30 priority 2 checkpoints were tested. Testing showed that the system violates only five of these 30 checkpoints. Figure 1 summarizes the errors found and is followed by a list of the unmet checkpoints.

| Page (Administration) | Tests | | | | |
|-----------------------------|--------|---------|----------------|-----------|---------|
| | Handle | RelSize | SameLinkPhrase | ConPhrase | FormLbl |
| 1_Login | | X | | | X |
| 1.1_MySpace | | X | X | | |
| 1.1.1_ManageMembership | | X | X | | |
| 1.1.1.1_JoinCommunityAction | | X | X | | |
| 1.2_MyCalendar(Day) | | X | X | X | X |
| 1.2.1_AddItem | X | X | X | X | X |
| 1.2.2_ViewCalendarEntry | X | X | X | X | X |
| 1.2.2.1_EditEntry | X | X | X | X | X |
| 1.2.2.2_DeleteEntry | | X | X | X | X |
| 1.2.3_CalendarList | | X | X | X | X |
| 1.2.4_CalendarWeek | | X | X | X | X |
| 1.2.5_CalendarMonth | | X | X | X | X |
| 1.3_MyFiles | | X | X | X | |
| 1.3.1_CreateURL | | X | X | | X |
| 1.3.2_UploadFile | X | X | X | | X |
| 1.3.3_CreateFolder | | X | X | | X |
| 1.3.4_UserFiles | | X | X | | X |
| 1.3.4.1_ViewSharedFolder | | X | X | | X |
| 1.3.4.2_DeleteFolder | | X | X | | |
| 1.4_ControlPanel | | X | X | | |
| 1.4.1_Help | | X | X | | |
| 2_ForgotPass | | X | | | X |
| 3_ChangeLocale | | X | | | X |
| 4_Register | | | | | |

| Page (Community) | Tests | | | | |
|-----------------------------|--------|---------|----------------|-----------|---------|
| | Handle | RelSize | SameLinkPhrase | ConPhrase | FormLbl |
| 1_CommunityHome | | X | X | | |
| 1.1_CommunityForum | | X | | | X |
| 1.1.1_ViewMessage | | X | X | | X |
| 1.1.1.1_RequestNotification | | X | X | | X |
| 1.1.1.2_PostMessage | | X | | | X |
| 1.1.1.2.1_PreviewPost | | X | | | |
| 1.1.1.3_ForwardMessage | | X | | | X |
| 1.1.2_CreateNewThread | | X | | | X |
| 1.1.3_SearchResults | | X | | | X |
| 1.2_FAQMain | | X | | | |
| 1.3_FileStorage | | X | | | |
| 1.3.1_ViewFolderContents | | X | X | | |
| 1.3.1.1_ViewFileDetails | | X | X | X | |
| 1.4_People | | X | | | |
| 1.4.1_SortPeople | | X | | | |

Figure 1: An X indicates that an error (of the column type) was detected on that page. (Note: We only show the tests that uncovered at least one violation. For a list of all 30 priority 2 tests performed, refer to the WCAG)

Unmet priority 2 checkpoints:

Handle (Checkpoint 9.3)

Make sure event handlers do not require use of a mouse.

RelSize (Checkpoint 3.4)

Use relative sizing and positioning (% values) rather than absolute (pixels).

SameLinkPhrase (Checkpoint 13.1)

Do not use the same link phrase more than once when the links point to different URLs.

ConPhrase (Checkpoint 13.1)

Create link phrases that make sense when read out of context.

FormLbl (Checkpoint 12.4)

Explicitly associate form controls and their labels with the LABEL element.

2.1.3 Priority 3 Automated Testing Results

All 19 priority 3 checkpoints were tested. Testing revealed that the system violates only four of these 19 checkpoints. Figure 2 and Figure 3 summarize the errors found and are followed by a list of the unmet checkpoints. (Note: Only tests that uncovered at least one violation were included in the tables. For a list of all 19 priority 2 tests performed, refer to the WCAG)

| Page (Administration) | Tests | | | |
|-----------------------------|--------|------|----------|------------|
| | TblSum | Lang | DefChars | Whitespace |
| 1_Login | X | X | X | X |
| 1.1_MySpace | X | X | | |
| 1.1.1_ManageMembership | X | X | | |
| 1.1.1.1_JoinCommunityAction | X | X | | |
| 1.2_MyCalendar(Day) | X | X | X | |
| 1.2.1_AddItem | X | X | X | |
| 1.2.2_ViewCalendarEntry | X | X | X | |
| 1.2.2.1_EditEntry | X | X | X | |
| 1.2.2.2_DeleteEntry | X | X | X | |
| 1.2.3_CalendarList | X | X | X | |
| 1.2.4_CalendarWeek | X | X | X | X |
| 1.2.5_CalendarMonth | X | X | X | X |
| 1.3_MyFiles | X | X | | |
| 1.3.1_CreateURL | X | X | X | |
| 1.3.2_UploadFile | X | X | X | |
| 1.3.3_CreateFolder | X | X | X | |
| 1.3.4_UserFiles | X | X | X | |
| 1.3.4.1_ViewSharedFolder | X | X | X | |
| 1.3.4.2_DeleteFolder | X | X | | |
| 1.4_ControlPanel | X | X | | |
| 1.4.1_Help | X | X | | |
| 2_ForgotPass | X | X | X | |
| 3_ChangeLocale | X | X | | |
| 4_Register | | | | |

Figure 2: An X indicates that an error (of the column type) was detected on that page

| Page (Community) | Tests | | | |
|-----------------------------|--------|------|----------|------------|
| | TblSum | Lang | DefChars | Whitespace |
| 1_CommunityHome | X | X | | |
| 1.1_CommunityForum | X | X | X | |
| 1.1.1_ViewMessage | X | X | X | X |
| 1.1.1.1_RequestNotification | X | X | X | |
| 1.1.1.2_PostMessage | X | X | X | |
| 1.1.1.2.1_PreviewPost | X | X | | |
| 1.1.1.3_ForwardMessage | X | X | X | |
| 1.1.2_CreateNewThread | X | X | X | |
| 1.1.3_SearchResults | X | X | | |
| 1.2_FAQMain | X | X | | |
| 1.3_FileStorage | X | X | | |
| 1.3.1_ViewFolderContents | X | X | | |
| 1.3.1.1_ViewFileDetails | X | X | | X |
| 1.4_People | X | X | | |
| 1.4.1_SortPeople | X | X | | |

Figure 3: An X indicates that an error (of the column type) was detected on that page

**Unmet priority 3 checkpoints:
TblSum (Checkpoint 5.5)**

Provide a summary for tables.

Lang (Checkpoint 4.3)

Identify the language of the text.

DefChars (Checkpoint 10.4)

Include default, place holding, characters in edit boxes and text areas.

Whitespace (Checkpoint 10.5)

Separate adjacent links with more than whitespace.

2.2 Manual Testing

2.2.1 Text-Only Navigation

Lynx (2005) was used to navigate through the .LRN installation. The purpose of using a text-only browser was two-fold. Firstly, some users do not have access to a graphical browser so it is important that they can use the site without one. Secondly, a text-only browser is a decent substitute for a screen reader, as both devices must access a web page serially. This text-only browsing revealed a number of problems with the accessibility of .LRN that could not be revealed by automatic testing. These are addressed in the development guidelines.

2.2.2 Graphical Browser Navigation

Various graphical browsers (Internet Explorer 6.0, Mozilla 1.0 and Opera 7.50) were used to view the .LRN installation. At various stages of testing, certain features were disabled in the browser to see if the accessibility of the site was affected. For example, using Mozilla 1.0 and disabling JavaScript enabled the testing of client-side scripting. Again, violations discovered here are addressed in the development guidelines.

3 Development Guidelines

A complete list of development guidelines was created based on the results of automatic and manual testing. Each problem uncovered via testing was assigned a unique problem ID and documented along with a compliance solution in the following format.

Problem ID:

A unique identifier of the problem of the form X.Y where X is the lowest priority level breached by the problem and Y is an index number unique inside its priority level.

Problem Name:

A short, descriptive, name to help identify the problem.

Page(s) Effected:

A list, or explanation, of which pages, or areas, of the site are affected by this problem.

Standards Violated:

The W3C Checkpoint that this problem violates.

Description:

A description of the problem. Some descriptions are accompanied by images but the descriptions are intended to be read, and comprehended, without the need for the images.

Solution:

A brief suggestion of how the problem can be solved.

In this way, every infringement on every page was identified, in some cases a screen shot was taken to specifically illustrate the problem, and a solution specific to each checkpoint violation was provided.

Here is an example of one instance (with problem ID) of a priority 1 checkpoint in violation.

Problem ID: 1.2

Problem Name: Header Logo Alt Text

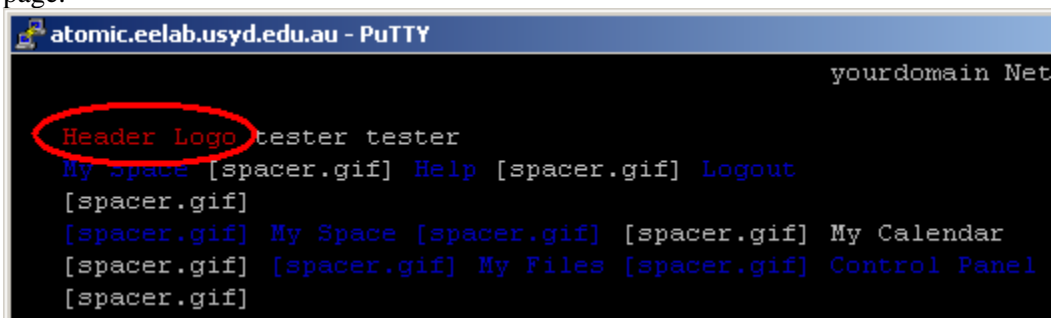
Page(s) Effected: All pages

Standards Violated:

1.1: Provide a text equivalent for every non-text element [Priority 1]

Description:

The alt text for the header logo reads as "Header Logo". This description is unnecessary since this image can be completely ignored without losing any comprehension of the web page.



Solution:

Change the alt text to "" so screen readers know this image can be ignored.

Another 29 instances of checkpoint violations were uncovered and documented with relevant solutions provided. It is beyond the scope of this paper to include the complete list. For complete descriptions of all problems and their development solutions, see the evaluation document available online at: <http://www.eelab.usyd.edu.au/~tkal6763/THESIS/accessibility.html>

4 Supporting the open source development process

There are two aspects to managing accessibility compliance for an open source LMS: one is retrofitting the core system and all current applications, the other is ensuring that all newly created applications are built to standards from the start. Those creating new packages have a key advantage, as it is generally easier to build accessibility compliance in from scratch. Either way, developers will need both an awareness of the problem, and incentive to build to the standard, as it does take some extra effort.

For many institutions, their legal and moral obligations will motivate them to make their applications accessible. For others, accessibility is an unfamiliar issue and legal obligations may appear less relevant. Therefore, it is worth investigating ways to provide further incentive from within the community itself. Communities may come up with a variety of creative ways to get accessibility included in the development process. One way, is through certification.

As an example, the .LRN development community is centred around a consortium. The consortium's mission is to support the software release process, provide software quality assurance by registering components as ".LRN-certified", and facilitate coordination and communication among the user and technical communities. Certification is a useful quality assurance method, and could be used for ensuring accessibility compliance as well. For development communities that employ some kind of certification process, certification is granted to a new package provided it meets certain requirements. One of these requirements should be level A accessibility compliance (the minimum set of guidelines as defined by the W3C). Thus, those seeking certification for new products would be made aware of accessibility issues, and would be provided with a clear motivation to build in compliance from the start.

Conclusion

There is a clear need for fully accessible learning management systems to be made available, so that educational institutions can meet the needs of all students, including those with disabilities. Since commercial vendors have yet to release compliant versions of the most popular LMS, one solution is to build compliance into open source systems. Evaluation is the first step to making these systems compliant. The process we have used to evaluate one open source system, .LRN, can be applied to any LMS. Results of a thorough evaluation, using both automated and manual testing of a default installation of .LRN, show that it is very close to being compliant with international accessibility standards. Only a single issue prevented it from satisfying the minimum set of checkpoints for Level A compliance, and the solution to this issue is easy to implement. In contrast, a total of nine checkpoint violations were uncovered for priorities 2 and 3 combined, making AAA compliance a longer-term goal. As a way of encouraging open source developers to build new applications according to standards, development communities should consider ways to provide incentive for compliance, such as certification that is dependent upon it. It is hoped that the development recommendations provided here will be used to ensure an accessible .LRN in the near future, and that more open source development communities will make accessibility compliance a goal for their systems.

References

- Australia (1992) Australian Disability Discrimination Act 1992
<http://scaleplus.law.gov.au/html/pasteact/0/311/top.htm> (Accessed February, 2005)
- Blackboard, Inc. (Accessed February, 2005) Accessibility <<http://www.blackboard.com/products/access/index.htm>>
- DETYA (Department of Education, Training and Youth Affairs) Australia (2000) Higher Education Students Time Series Tables, Selected Higher Education Statistics 2000.
- Faulkner, S. and Arch, A. (2000) "Accessibility Testing Software Compared"
The Ninth Australian World Wide Web Conference, 5-9 July 2003 Gold Coast
<<http://ausweb.scu.edu.au/aw03/papers/arch/paper.html>>
- ICDRI (The International Center for Disability Resources on the Internet), ISOCDISAB (Internet Society Disability and Special Needs Chapter), HiSoftware (Accessed February, 2005) Welcome to Cynthia Says Portal
<<http://www.cynthiasays.com>>
- Kirchner, M. (2003) "Benchmark for testing the evaluation tools for Web pages accessibility" Web Site Evolution, 2003. Proceedings. Fifth IEEE International Workshop 22 September 2003 pp. 66 – 73

Kirchner, M (2002) "Evaluation, repair, and transformation of Web pages for Web content accessibility: Review of some available tools" Web Site Evolution, 2002. Proceedings. Fourth IEEE International Workshop 2 October 2002; pp 65 – 72

Loiacono, E.T. (2003) "Improving Web accessibility" IEEE Computer Volume 36, Issue 1, January 2003 pp 117 – 119

Lynx (Accessed February 2005) Lynx Information <<http://lynx.browser.org>>

Miniwatts International, Inc (2005) Internet Usage Statistics: The Big Picture <<http://www.internetworldstats.com/stats.htm>> (Accessed February, 2005)

Ford, S. (2004) Testing for Accessibility <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc/html/accessibility_testing.asp> (Accessed February 2005)

Thatcher, J. (Accessed February 2005a) Web Accessibility - What not to do <<http://www.jimthatcher.com/whatnot.htm>>

Thatcher, J. (Accessed February 2005b) Evaluation of Bobby <<http://www.jimthatcher.com/bobbyeval.htm>>

Thatcher, J. (Accessed February 2005c) LIFT from UsableNet <<http://www.jimthatcher.com/lifteval.htm>>

Thatcher, J. (Accessed February 2005d) Testing for Section 508 Accessibility <<http://www.jimthatcher.com/testing.htm>>

Treasury Board of Canada (2004) Canadian Common Look and Feel Guidelines http://www.cio-dpi.gc.ca/clf-nsi/index_e.asp (Accessed February, 2005)

United States of America (2002) Section 508 <http://www.section508.gov/> (Accessed February, 2005)

UsableNet, Inc. (2004) UsableNet – Website Testing Systems <<http://www.usablenet.com>> (Accessed February, 2005)

United Kingdom (1995), Disability Discrimination Act 1995 <<http://www.disability.gov.uk/dda/>> (Accessed February, 2005)

Vanderheiden, G.C. (1990). "Thirty-something million: should they be exceptions?" Human Factors, 32(4), pp. 383-396.

Watchfire Corporation (Accessed February, 2005) Welcome to Bobby Worldwide <<http://bobby.watchfire.com/bobby/html/en/index.jsp>>

WebCT, Inc. (Accessed February, 2005) WebCT Standards > Accessibility <http://www.webct.com/standards/viewpage?name=standards_accessibility>

W3C (World Wide Web Consortium) (2005) Policies Relating to Web Accessibility <<http://www.w3.org/WAI/Policy>> (Accessed February 2005)

W3C (World Wide Web Consortium) (2002) Evaluating Web Sites for Accessibility <<http://www.w3.org/WAI/eval>> (Accessed February 2005)