

Accessibility in a Virtual Classroom: a Case Study for the  
Visually Impaired Using WebCT

by

Shohreh Hadian

B.Sc., City University, Washington, USA, 1987

A Thesis Submitted in Partial Fulfillment of the  
Requirements for the Degree of

**MASTER OF SCIENCE**

in the

Department of Computer Science.

© SHOHREH HADIAN, 2004

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by  
photocopy or other means, without the permission of the author.

Supervisors: Dr. Margaret-Ann Storey

## Abstract

Web based education has become a norm in delivery of courses in universities, schools and corporations. In order to ensure universal access to this form of education, it is important for developers, authors and users to become aware of accessibility and of the potential barriers that may be encountered in these virtual environments. To this end, this thesis investigates accessibility barriers faced by users with visual impairment in virtual classrooms.

The research starts by introducing a definition of accessibility in the context of good software design practices. Here, accessibility has been looked at from interoperability, adaptability, learnability, operability, changeability and fault tolerance perspectives.

Most of the web based learning tools assume the role of an authoring tool or an user agent. The research was approached from two perspectives: the author's responsibilities to make their web material more accessible to assistive devices for the blind; and from a user's perspective, the barriers faced by the visually impaired were noted during a case study and recommendations have been noted for administrators, authors and users of teaching and learning tools.

From an authoring perspective, two commercial tools, WebCT and Blackboard, have been evaluated for their ability to help and guide the author in creating code that adheres to the Web Accessibility Initiative (WAI) guidelines implemented by the World Wide Web Consortium (W3C).

From the learner's point of view, a user study was designed and implemented to identify the barriers faced by the visually impaired. The study involved three participants covering a broad spectrum of disability from totally blind to partially

blind. A fully sighted user was also selected for control purposes. To make the course material accessible to the blind, LIFT in combination with Dreamweaver, was selected for the user study. Also, assistive devices were identified for use by people with visual impairment and three devices were used during the testing phase: JAWS, Zoomtext and Braille. Recommendations have been put forward for developers and authors to better understand how people use assistive devices, and how design features can enhance or impede accessibility.

Finally, this research has raised fundamental new issues. In the quest for universal access and design for all-inclusive virtual classrooms, user diversity and reliability of client-server architectures are also important design issues. To this end, a prototype was designed to demonstrate the potential use of Intensional Sequential Evaluator (ISE) and its role in customizing accessibility for all user groups by allowing the design of multi-versioned web sites. The issue of reliability is also considered and some recommendations are provided.



# Table of Contents

<b>Abstract</b>	<b>ii</b>
<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Rationale and Objectives . . . . .	2
1.2 Definition of Accessibility . . . . .	4
1.3 A Brief Review of the Literature . . . . .	7
1.4 Visual Impairment . . . . .	10
1.5 Outline of the Thesis . . . . .	11
<b>2 Authoring, Verification and Repair</b>	<b>14</b>
2.1 Evaluation Framework . . . . .	15
2.2 Software Criteria . . . . .	17
2.2.1 Operability . . . . .	17
2.2.2 Learnability . . . . .	18
2.3 Authoring and Learning Tools . . . . .	19
2.3.1 WebCT . . . . .	20
2.3.2 Blackboard . . . . .	21
2.3.3 Other Tools . . . . .	21
2.4 Evaluation of Two Specific Tools: WebCT and Blackboard . . . . .	23
2.4.1 Authoring Issues . . . . .	23
2.4.2 Partial User Agent . . . . .	29
2.5 Verification and Repair Tools . . . . .	41
2.6 Synopsis . . . . .	43
<b>3 Assistive Technologies</b>	<b>45</b>
3.1 Types of Assistive Devices for the Blind . . . . .	46

3.1.1	Screen Readers . . . . .	46
3.1.2	Screen Magnification . . . . .	47
3.1.3	Refreshable Braille Displays . . . . .	48
3.1.4	Braille Embossers . . . . .	50
3.1.5	Haptic Devices . . . . .	50
3.1.6	Voice Recognition . . . . .	50
3.2	Synopsis . . . . .	51
<b>4</b>	<b>A Case Study for the Visually Impaired Using WebCT</b>	<b>53</b>
4.1	The Testing Environment . . . . .	54
4.2	The Hardware and Software . . . . .	55
4.3	Material Selection . . . . .	57
4.4	Course Content Verification and Repair . . . . .	57
4.4.1	Tool-Assisted Issues . . . . .	58
4.4.2	Manual Issues . . . . .	61
4.4.3	LIFT bugs . . . . .	64
4.5	User Study . . . . .	64
4.5.1	Participant selection . . . . .	65
4.5.2	Design of questionnaire . . . . .	66
4.5.3	Tasks . . . . .	67
4.5.4	Results . . . . .	69
4.5.5	Observations . . . . .	74
4.6	Study limitations . . . . .	76
4.7	Recommendations . . . . .	76
4.7.1	Navigation Bar . . . . .	77
4.7.2	Loading of Pages . . . . .	77
4.7.3	Title Page . . . . .	77
4.7.4	Labels . . . . .	77
4.7.5	Images . . . . .	78
4.7.6	Pop-up Windows . . . . .	78
4.7.7	Scrolling . . . . .	79
4.7.8	Page Layout . . . . .	79
4.7.9	Tables . . . . .	80
4.8	Synopsis . . . . .	80
<b>5</b>	<b>Other Accessibility Issues</b>	<b>82</b>
5.1	Maintainability and Changeability . . . . .	84
5.1.1	Example: A repository for assistive devices . . . . .	89
5.2	Reliability and Fault-Tolerance . . . . .	91
5.3	Synopsis . . . . .	92

<b>6</b>	<b>Conclusions and Recommendations</b>	<b>93</b>
6.1	Research Contributions . . . . .	95
6.2	Future Work . . . . .	96
	<b>Bibliography</b>	<b>97</b>
<b>A</b>	<b>Correlation between WCAG and Section 508</b>	<b>103</b>
<b>B</b>	<b>Experimenter's Handbook for WebCT</b>	<b>109</b>
B.1	Pre-Study Questionnaire . . . . .	110
B.2	Orientation . . . . .	120
B.3	Assistive Tool Setup . . . . .	122
B.4	Warm-up Session . . . . .	123
B.5	Access Tasks . . . . .	124
B.6	Post-Questionnaire/Interview . . . . .	130
<b>C</b>	<b>Participant Consent Form</b>	<b>132</b>

## List of Tables

1.1	Desirable features in software design . . . . .	6
2.1	Evaluation for accessibility of web-based authoring tools . . . . .	33
2.2	Evaluation for accessibility of web-based authoring tools (cont.) . . . . .	34
2.3	Evaluation for accessibility of web-based authoring tools (cont.) . . . . .	35
2.4	Evaluation for accessibility of web-based learning tools - partial user agent . . . . .	36
2.5	Evaluation for accessibility of web-based learning tools - partial user agent(cont.) . . . . .	37
2.6	Evaluation for accessibility of web-based learning tools - partial user agent (cont.) . . . . .	38
2.7	Evaluation for accessibility of web-based learning tools - partial user agent (cont.) . . . . .	39
2.8	Evaluation for accessibility of web-based learning tools - partial user agent (cont.) . . . . .	40
4.1	Time taken and ranking attributed to the participants for task completion	71
A.1	The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act . . . . .	104
A.2	The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.) . . . . .	105
A.3	The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.) . . . . .	106
A.4	The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.) . . . . .	107
A.5	The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.) . . . . .	108
A.6	Section 508 paragraphs related to user agent recommendations . . . . .	108

## List of Figures

1.1	Accessibility barriers in web-based education . . . . .	3
1.2	Diagram showing the sub-characteristic used in the definition of accessibility . . . . .	7
1.3	User study performed from an author's and user's perspective . . . . .	10
2.1	Accessibility sub-characteristics used in the qualitative evaluation of learning tools . . . . .	15
3.1	Braille display . . . . .	49
4.1	Accessibility sub-characteristics considered in the user study for the visually impaired. . . . .	54
4.2	The methodology adopted for the user study . . . . .	56
4.3	The filtered and layered testing environment . . . . .	57
4.4	The LIFT error message templates showing the wizard applet, the evaluation applet and the reference box . . . . .	59
4.5	The LIFT error message templates showing manual issues . . . . .	63
5.1	Accessibility barriers in web-based education . . . . .	83
5.2	The ISE program structure . . . . .	86
5.3	The IHTML multiversioning tool . . . . .	87
5.4	The IHTML setup . . . . .	88
5.5	The search engine . . . . .	90
5.6	The internal structure of the search engine . . . . .	90

*“Be the change you want to see in the world.”*

## Acknowledgements

I am deeply grateful to my supervisor, Dr. M. A. Storey, for her invaluable guidance and advice and motivation during the course of my graduate studies. She has provided me with a platform on which to further my knowledge and understanding of human-computer interaction and accessibility issues in a virtual classroom.

I would also like to acknowledge Dr. W. Wadge for his suggestions and advice as a member of the supervisory committee.

# Chapter 1

## Introduction

For the last ten years, the web has become a viable medium for teaching, researching and learning. The primary users include universities, schools and corporations. In order to ensure universal access to this form of education, it is important for developers, authors and users to become aware of accessibility issues and of the potential barriers that may be encountered in these virtual environments.

Web-based education can be adapted to different learning styles, rates of learning, and communication modes. Issues related to distance, transportation and physical access are reduced. A preliminary study conducted at the University of Toronto in 1998 [1] revealed that the available Web-based teaching tools fail to properly consider accessibility. A subsequent study in 2000 noted considerable improvements in interface design as developers are becoming aware of accessibility barriers [2]. Further studies [3] indicate, however, that web accessibility is about three times better for sighted users than for users with visual impairments. This represents an obstacle to an all-inclusive education that needs to be corrected before teaching and learning tools are further developed. Fortunately, teaching and learning tools are still evolving and if proactive steps are taken, more inclusive design guidelines can be established

and implemented.

## 1.1 Research Rationale and Objectives

Web-based education presents several benefits as students with sensory, physical or learning disabilities are able to adjust the rate of content delivery. The material can be presented in various formats, and it is possible to review supporting material without interrupting the flow of learning.

However, the virtual classroom has created new accessibility concerns for people with disabilities while eliminating the physical and temporal barriers of access.

Computer systems need to work as a supporting 'tool'. Bederson states that the user needs to stay in the task domain and not in the interface domain. Interruption, literal or conceptual, affects the flow of learning and this gets in the way of users concentrating on their task. In other words, when users have to constantly resolve issues in the periphery of the task, or struggle to fix interface issues, they are unable to concentrate on the task at hand [4]. An optimal learning experience can be ensured when tools are designed with usability and accessibility in mind.

In 2000, Wang and Storey [5] conducted a study on the usability of web-based learning tools. The study revealed that tools seem to enhance learning when they are perceived as being invisible, and authors play an important role in the process. Their study also recommends that the development of web-based learning tools with usability and accessibility in mind facilitates the learning process.

This thesis addresses the accessibility issues in web-based learning environments, and a case study has been implemented specifically aimed at people with visual impairments. This user group presents challenging problems when designing for inclusiveness. The accessibility barriers addressed in this thesis are summarized in Figure

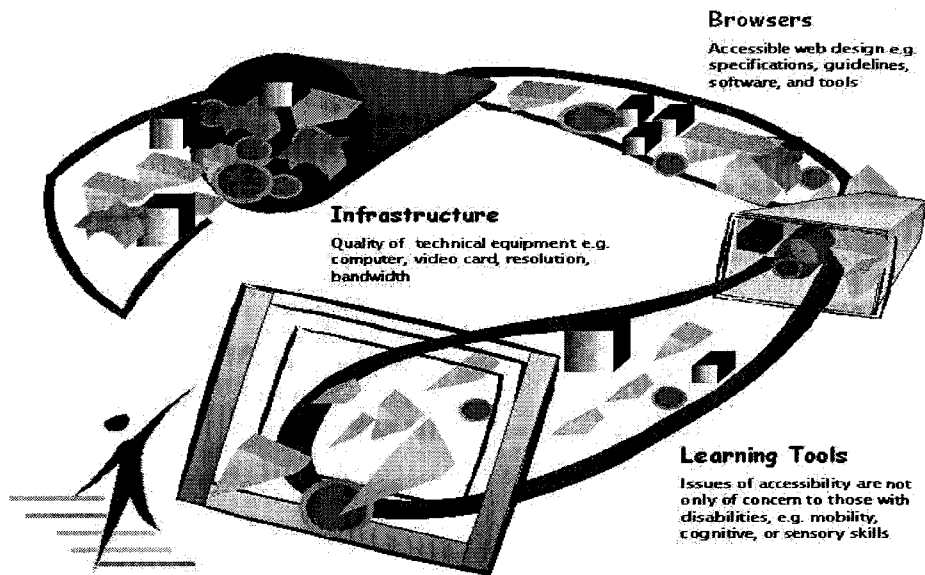


Figure 1.1: Accessibility barriers in web-based education

1.1 and these are: (i) web-based learning tools and assistive technologies for the visually impaired; (ii) browser design issues; and (iii) infrastructure.

This study builds upon the existing foundation of information on Web accessibility by providing further discussion and research on this topic. The study investigates barriers faced by the visually impaired in virtual learning environments, and provides guidelines for developers, authors and users to ensure accessibility of online materials. The findings and recommendations from the present study may help future development of teaching and learning tools.

## 1.2 Definition of Accessibility

The primary concern in the development of accessible web-based information lies in the definition of accessibility. On the one hand, web accessibility has been perceived solely as a disability issue. On the other hand, it is an essential component in the development of universal design [6]. Examples of universal design in buildings are the inclusion of ramps and automatic door openers as well as Braille labels on elevator control buttons. Just as wheelchair ramps to buildings have become the norm in architectural design, accessible web design must become a basic consideration in the design of Web-based learning resources. Using universal design principles to create a web resource ensures that all users can access information at a Web site regardless of their abilities, their disabilities, or the limitations of their equipment and software [7, 8].

According to Owston [9], "access to learning" means making education more accessible for someone with a disability, and this in turn makes it more usable for everyone else. Accessibility also means that anyone using any kind of Web browsing technology must be able to visit any site and get a full and complete understanding of the information as well as have the full and complete ability to interact with the site. For example, when accessing poorly designed multimedia based Web content, some users cannot see graphics because of visual impairments; others cannot hear audio because of hearing impairments; and a large number of users have difficulty navigating sites that are poorly organized with unclear directions because they have learning disabilities.

In order to understand the characteristics related to accessibility, a rationale based on the ESSI-SCOPE software design guidelines was adopted [10, 11]. ESSI-SCOPE is an EU (European Union) funded project that aimed to raise awareness of quality

issues in software products. This research has identified a list of desirable features in software design. In this thesis, we have selected the subset of these characteristics that impact accessibility and these are highlighted in Table 1.1 and these are graphically depicted in Figure 1.2. Note that the subset includes only those characteristics that directly affect accessibility.

*Interoperability* is the attribute that bears on the tool's ability to interact with specified operating systems; *adaptability* is the attribute that bears on the opportunity for the tool's adaptation to different specified environments without applying actions or means other than those provided for this purpose for the software considered; *operability* is the ability of software that bears on the user's effort for operation and operation control; *learnability* is the attribute that bears on the user's effort for learning its application; *changeability* is the attribute that bears on the effort needed for modification and environmental change; and *fault-tolerance* is the attribute of the software that bear on its ability to maintain a specified level of performance in case of software faults.

Table 1.1: Desirable features in software design

We focus on the highlighted issues (shown in <b>bold</b> ) [10]		
Characteristics	Sub-characteristics	Definitions
Functionality	Suitability	Attributes of software that bear on the presence and appropriateness of a set of functions for specified tasks
	Accurateness	Attributes of software that bear on the provision of right or agreed results or effects
	<b>Interoperability</b>	Attributes of software that bear on its ability to interact with specified systems
	Compliance	Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions
	Security	Attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs or data
Reliability	Maturity	attributes of software that bear on the frequency of failure by faults in the software
	<b>Fault tolerance</b>	Attributes of software that bear on its ability to maintain a specified level of performance in case of software faults or of infringement of its specified interface
	Recoverability	Attributes of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it
Usability	Understandability	Attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.
	<b>Learnability</b>	Attributes of software that bear on the users'effort for learning its application.
	<b>Operability</b>	Attributes of software that bear on the users'effort for operation and operation control
Efficiency	Time behaviour	Attributes of software that bear on response and processing times and on throughput rates in performances its function
	Resource behaviour	Attributes of software that bear on the amount of resource used and the duration of such use in performing its function
Maintainability	Analyzability	Attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified
	<b>Changeability</b>	Attributes of software that bear on the effort needed for modification, fault removal or for environmental change
	Stability	Attributes of software that bear on the risk of unexpected effect of modifications
	Testability	Attributes of software that bear on the effort needed for validating the modified software
Portability	<b>Adaptability</b>	Attributes of software that bear on the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered
	Installability	Attributes of software that bear on the effort needed to install the software in a specified environment
	Conformance	Attributes of software that make the software adhere to standards or conventions relating to portability
	Replaceability	Attributes of software that bear on opportunity and effort using it in the place of specified other software in the environment of that software

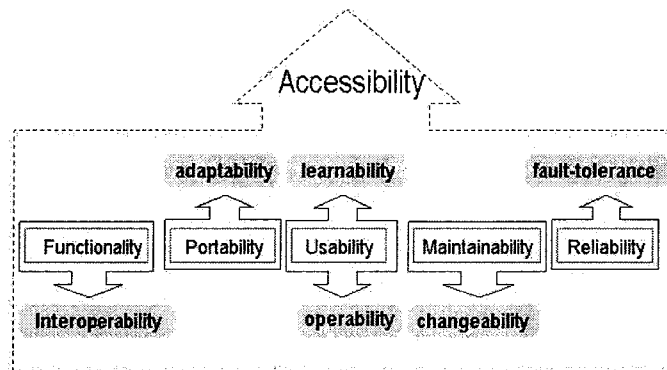


Figure 1.2: Diagram showing the sub-characteristic used in the definition of accessibility

### 1.3 A Brief Review of the Literature

The National Educational Association for Disabled Students (NEADS) reported on the education and employment experiences of 424 college and university students and graduates from across Canada. They found that students with disabilities faced access barriers to education and employment at every stage. The study stressed the critical need to take proactive steps to ensure that new education practices, such as distance education, promotes accessibility [12].

In April 1997, the World Wide Web Consortium (W3C) [13] created the Web Accessibility Initiative (WAI) to promote accessibility of web-based information [14]. The W3C develops the common protocols used on the Web to promote universal access. The WAI is responsible for ensuring that access provisions are integrated into new standards of Web technology. In June 2001, Section 508 of the 1973 Rehabilitation Act was implemented in the United States to ensure that federal employees with disabilities have equitable access to electronic and information technology [15].

The Government of Canada has also implemented the Connecting Canadians ini-

tiative [16] known as the Common Look and Feel (CLF) Standards. Approved in 2000, the purpose of CLF was to ensure that all Canadian Federal Government web sites adhere to guidelines with respect to design and these include accessibility, e-mail, notices, navigation and format, and official languages. Understanding that many Canadians access the Internet using assistive devices, the government wants to ensure equitable access to everyone for its web-based information, regardless of the form and type of technology used.

Currently, there are a few research initiatives trying to address the accessibility issues on web-based information. The DO-IT (Disabilities, Opportunities, Internetworking and Technology), based at the University of Washington, is an organization that assists people with disabilities in successfully pursuing academics and careers, and offers workshops and educational outreach to promote the use of technology to maximize the independence, productivity and participation of people with disabilities [17].

EASI (Equal Access to Software and Information), based in Rochester New York, is a virtual organization that provides information and guidance in the area of access-to-information technologies by individuals with disabilities. Outreach programs include both on-site and on-line workshops, and use of web to research and disseminate information to colleges, universities, K-12 schools, libraries and into the workplace [18].

The SNOW (Special Needs Opportunity Windows) project is based at the University of Toronto. The goal of the SNOW project is to provide professional development and teaching resources to educators of students with special needs. This effort has focussed on the development and moderation of on-line courses and discussion forums, dissemination of information and resource materials related to the education of students with special needs, and electronic delivery of curriculum resources in accessible

formats [19]

In 1998, Bruce Landon investigated the pedagogical aspects of web based authoring tools [1]. The SNOW project extended this study with another in 2000, and concluded that these tools had somewhat improved the accessibility of web-based information, but further work was needed [2]. However, this research did not address the issue of accessibility of tools with respect to users with disabilities.

The NODE Learning Technologies Network, based in London, Ontario, is a not-for-profit electronic network, facilitating information and resource-sharing, collaboration and research in the field of learning technologies for post-secondary education and training. The NODE's Web site is a focal point for information and discussion forums on issues related to teaching, learning and technological development [20].

The focus of the research on accessibility so far has been from a disability perspective. The onus has been placed on the user to find solutions to access web information. However, usability studies have concluded that learning can be enhanced when the tools themselves are perceived to be invisible to the user [21, 22, 23]. In order to provide optimal experiences to the user, this thesis proposes to approach the issue from the author's perspective as well, as illustrated in Figure 1.3. To that end, this research explores and identifies challenges of integrating the various components (browsers, authoring tools, learning tools and adaptive technologies) necessary to ensure optimal experience for the end users.

Most of the open literature on the subject is currently available online and not in the conventional library journals, as the subject is very new in this research area.

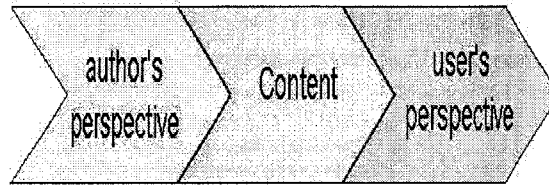


Figure 1.3: User study performed from an author's and user's perspective

## 1.4 Visual Impairment

At the present time, in North America, 24% of the population has experienced significant vision loss. Whereas 1% are blind from birth, the other 23% have experienced vision impairment later in life. The major causes of vision impairment include diabetes and macular degeneration [24]. According to a study at the University of Washington, the number of people with low vision and potential blindness cases in the near future is rising rapidly as the population ages [25]. Considering that 100 million people in North America are visually disabled without corrective lenses, and over 11 million people have severe visual conditions not correctable by glasses, these figures present persuasive statistics about working with visually impaired people.

Assistive devices provide special computing software and hardware to make things easier and more accessible for people with disabilities. The major disability types include visual and physical impairments; hearing deficiencies, learning disabilities and speech impediments. In order to understand the issues related to accessible Web design, it is important to have a basic understanding of specific access systems that accommodate, replace or augment sensory and motor functions of users with special needs.

For blind people, assistive or adaptive devices are available in the form of voice recognition systems, alternative pointing devices, screen readers and screen magnifiers. For example, a computer and a screen reader can give a blind person access to online books and journals. A computer, scanner, and a screen reader provide a way for a blind person to read a printed book [26]. A description of some of the assistive output technologies is presented in Chapter 3.

## 1.5 Outline of the Thesis

In order to identify issues related to the design and delivery of an accessible web-based course, we designed and implemented a case study for the visually impaired using a popular web-based authoring/learning tool. The study enabled the identification of accessibility barriers for the visually impaired both from the author and the user point of view, and subsequently recommendations were suggested to guide designers and institutions. The thesis is organized in six chapters.

Chapter 1 has already presented an introduction to the subject, the research rationale and goals, a definition for accessibility is proposed, and the current state of the art on the research and developments in this rapidly evolving field.

Chapter 2 This part of the research extends the work of wang [21] which looked at the usability of web-based learning tools. The research presents a qualitative evaluation, from the author's perspective, of the same two popular web-based authoring tools (Blackboard and WebCT) with the focus on accessibility for disabled users. The software criteria and characteristics for accessibility from the authoring perspective include operability and learnability. "Section 508" of the rehabilitation act was used as a framework for evaluation.

The last section in this chapter presents verification and repair tools that can be

used to enhance a-posteriori the accessibility of HTML pages. In this thesis, LIFT [27] was used to repair the accessibility of the online course content selected for the user study presented in Chapter 4.

Chapter 3 provides an overview of the assistive or adaptive devices that are used by people with visual impairments to enable them to access web material. A description of available devices is presented in order to inform the reader on the various solutions adopted by users with disabilities.

Chapter 4 presents a case study for the visually impaired based on a course which is currently part of the curriculum of the Computer Science Degree at the University of Victoria (SENG 310 - Human Computer Interaction). This course has been the subject of a previous case study on the usability of web based learning. The course content was verified and repaired to improve the accessibility for the visually impaired. Three participants who are visually impaired were selected. The participants spanned the range of visual impairment, from totally blind to partially blind. Additionally, a fully sighted user was included for control purposes. The course content was delivered using the web-based learning tool (WebCT) in a controlled environment. The focus of this case study was to identify and document the barriers that the visually impaired users face in a virtual classroom setting and provide course content development guidelines. The case study provides a view of this technology from the administrator's, author's and user's perspective.

Chapter 5 investigates the two further issues that affect accessibility i.e. changeability and fault-tolerance. It is necessary to support portability and interoperability so that data can flow through many networks in a way transparent to the user [28]. In this respect, changeability relates to the diversity in the type of users, and the delivery of user-targeted information. To this end, an application based on multi-versioning (customizing the delivered material) for different types of users was designed and im-

plemented. Fault-tolerance relates to the attributes of software to maintain a specified level of performance in case of software or hardware faults. To this end, we refer to a distributed multimedia system that was designed for fault tolerance services using quality of service adaptation during faults in servers.

Finally, Chapter 6 provides some concluding remarks, a summary of our recommendations to enhance accessibility of web based instruction, and finally research ideas for further work are proposed.

## Chapter 2

# Authoring, Verification and Repair

Over the last five years, there has been a surge of concern regarding web accessibility on the part of web developers. This can primarily be attributed to the Section 508 (United States) and CFL (Canada) requirements now mandating the accessibility of government web sites. Nevertheless, the challenge to make sure that all authors follow accessibility guidelines still remains.

Verification and repair tools aid the author in identifying the changes needed in an existing HTML code in order for the pages to conform to accessibility standards. The interface generally presents a series of prompts and dialogue boxes, requiring input on the part of the author to meet compliance based on the set of accessibility standards or level of compliance chosen. While there has been a surge of new developments in the area of verification and repair, little attention has been paid to the role of authoring tools, such as HTML editors.

This chapter builds upon the current literature on web authoring/learning tools. The study provides a qualitative evaluation of two popular tools (*WebCT* and *Blackboard*) based on operability and learnability software design sub-characteristics from

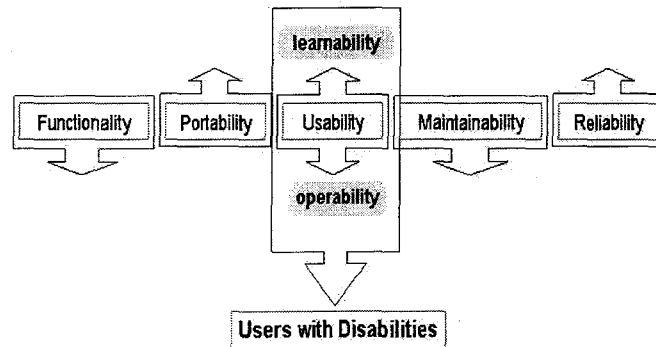


Figure 2.1: Accessibility sub-characteristics used in the qualitative evaluation of learning tools

the author's perspective and aimed at users with visual impairment, as illustrated in Figure 2.1. At the end of the chapter, currently available verification and repair tools are described.

## 2.1 Evaluation Framework

In 1998, the American with Disabilities Act (ADA) of 1990, was amended to include the Section 508 of the Rehabilitation Act [15]. In June 2001, to make Section 508 enforceable, the Access Board published specific standards that spelled out what makes information (technology products, computer software, electronic office equipment and web-based services) accessible to people with disabilities, including those with vision, hearing and mobility impairment.

WCAG sets guidelines for accessibility of web pages in terms of priority levels. Priority 1 (P1) level guidelines are recommendations essential for ensuring web content accessibility for some of the users. Priority levels 2 and 3 (P2 and P3) are extended

recommendations to ensure universal inclusiveness. Section 508 relates to the U.S. federal government legislation designed to enforce the priority 1 level issues identified by the WCAG.

Each guideline has a priority level assigned based on the impact on accessibility as follows:

P1 A Web content developer must satisfy this requirement. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying this criterion is a basic requirement for some groups to be able to use Web documents. Adherence to all P1 guidelines corresponds to WCAG 1.0 Level A conformity.

P2 A Web content developer should satisfy this requirement. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this guideline will remove significant barriers to accessing Web documents. Adherence to all P1 and P2 guidelines corresponds to WCAG 1.0 Level AA conformity.

P3 A Web content developer may address this guideline. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this criterion will improve access to Web documents. Adherence to all P1, P2 and P3 guidelines corresponds to WCAG 1.0 Level AAA conformity.

Tables A.1-A.5 in Appendix A present the Web Content Accessibility Guidelines (WCAG) and these guidelines are cross-referenced with Section 508 §1194.22 of the Disabilities Act, which relates to a set of standards on web-based intranet and internet information and applications. As the developers are required to comply with Section 508, these priority level 1 guidelines are used here as criteria.

Finally, Table A.6 presents the requirements put forward by Section 508 which are not part of the P1 issues in WCAG guidelines. These issues (Section 508 paragraphs l-p) pertain to utilizing scripting languages to display content and creating interface elements that can be read by assistive technologies.

## 2.2 Software Criteria

The definition of accessibility presented in Chapter 1 listed the criteria that should be considered in software design. From the authoring perspective, the operability and learnability characteristics have the most influence in ensuring accessibility and these are explained next.

### 2.2.1 Operability

The ease to which a feature, such as an ALT text field in an image insertion utility is accessible during the authoring process is referred to as operability. Thus several levels of operability can be defined in order to qualify the accessibility of the tool. For example, an interface may be available which allows the insertion of an image on the web page. To make that image accessible to a student who is blind, an alternative text description is required. A utility for including the alternative text can be very clearly identified and explained (full support), or it can be hidden away in secondary levels of the interface (partial support), or it may not be present at all (no support) [2].

For example, continuing with the ALT text example, an image tool may:

1. Provide no way to add ALT text.

2. Provide a way to enter ALT text but either as a secondary step, or hidden away in a field on a secondary window or it may provide a way to enter the ALT text on the interface.
3. Provide a field to enter ALT text just below the field that identifies the image or it may prompt the designer when ALT text has not been entered or requires the designer to include ALT text before continuing.

### 2.2.2 Learnability

Learnability deals with issues such as how does the authoring tool help the author to know about the barriers and what does it do to help them solve and overcome these barriers through help files. The learnability properties provide choice, selection, or input utilities for creating course content by the course designer/instructor. The extent to which these utilities are designed and available within the tool's system are addressed by the learnability sub-characteristic. It refers to the presence of help files, explanations and examples that describe techniques to increase the accessibility of Web pages.

For example, if no information for this accessible authoring practise is mentioned in the help system this qualifies as no support; if the feature is mentioned in help and has context sensitive prompting and/or includes explanation in the help system (partial support); and finally if the feature is mentioned in help, context sensitive prompting, includes explanation, and has explanatory examples, this qualifies the characteristic as fully supported.

## 2.3 Authoring and Learning Tools

In order to perform a qualitative analysis of web-based teaching and learning tools, one must take into account their role. Most of the web based learning tools assume the role of an *authoring tool* and/or to a lesser extent of an *user agent*. As a user agent, the learning tools retrieve and render web content such as quizzes and assignments, etc. Authoring tools allow an instructor to design and create instructional material for publishing on the web. Although most of these authoring tools may require HTML language knowledge, some programming or other technical expertise is needed to effectively use these tools.

Key characteristics of authoring/learning tools typically include synchronous and asynchronous facilities for communication such as capability for secure student login via standard Java browser, centralized database-centered syllabus with links to internal or external web pages, on-line, time-monitored quizzes with randomized dynamically-generated testing, chat and discussion groups, and integrated email.

To make web content accessible means making the authoring and learning tool interoperable (functionality), adaptable (portability), operable and learnable (usability). The tools needs to have the ability to interact with specified operating systems (interoperability); and the tool needs to adapt to different environments without interference or actions other than those provided for this purpose (portability). For accessibility, the authoring tools also need to be operable to enhances the user's effort for operation and operation control; and learnable which bears on the user's effort for learning the application. The operability and learnability sub-characteristics determine how well the authoring tool deals with user interface issues, and the way the tool helps the author to navigate, locate, and learn about the availability of utilities that make the web material accessible to the user.

Wang and Storey [21] have concluded a study on the usability of web-based learning tools from the users, instructors and administrators point of view. Their study focused on the learnability and operability issues from a usability perspective. This qualitative and quantitative study concentrated on the setting up of the tools, checking the various components, and navigability issues as it relates to the user. Furthermore, their findings were supported by Pearrow [29], in his book "Web Site Usability Handbook", which concluded that accessibility is a critical component of usability. Here, we have extended the usability study by focusing on the operability and learnability of authoring tools with emphasis on accessibility.

### **2.3.1 WebCT**

WebCT was originally developed by the Department of Computer Science at the University of British Columbia [30]. WebCT has a following based on its early days as a "Shareware" UNIX-based product. The three groups of end-users (students, instructors and course administrators) access a centralized server through a web browser. The course content consists of HTML pages or PDF files. The class management features include an assignment submission tool, an online quiz tool and a checking grades tool.

In its latest version 4.0 (July 2003), WebCT has continued the implementation of measures initiated in the Version 3.6 (July 2001) to address compliance with each of the paragraphs of the Section 508 guidelines. WebCT now provides help on strategies for accessible course design, and have made more specific documentation available in the WebCT Course Tool Help system.

### 2.3.2 Blackboard

Blackboard's product (previously known as "Courseinfo") is a commercial course web-based tool based on a client-server architecture [31]. The Blackboard product suite does not provide significant internal content development/authoring tools, but has import features from many stand-alone content development tools.

From the architectural perspective, the three groups of end-users (students, instructors and administrators) access a centralized server using a web browser. The features intended for students include an announcement board, course information, course documents, an assignment area and communication tools.

### 2.3.3 Other Tools

For completeness this section describes other less popular tools that are currently available in the market. The information presented has been obtained from the literature.

Lectora Publisher [32] was designed to author e-learning courses without any prior programming knowledge. Educators are able to create any number of learning objects, from interactive tests to product team updates, quickly and easily. Lectora can be used as an authoring tool to complement Blackboard.

Virtual-U was initiated at Simon Fraser University [33]. Virtual-U has a bulletin board messaging, and it is suited for online learning using virtual communication and collaboration. However, the use of frames, the incorrect use of tables, and inconsistency between views makes it difficult for adaptive technologies to navigate.

LearningSpace [34] is a product of IBM client-server systems and it is an add-on application written in Lotus Notes. It offers a wide variety of pedagogical tools in a relatively accessible environment.

Prometheus [35] was developed at the George Washington University to meet the online infrastructure needs of educational institutions, and it is one of the first "open-source" software learning management systems. In 2002, it was purchased by Blackboard. Partners are encouraged to share code and advice in an open environment-facilitating customization opportunities.

Darasoft Class Act [36] is a web-based course management system designed for educational institutions. It can be used to deliver web based learning support via online forums, discussion groups and web links to content. It is possible to create courses, forums, topics and threads, students can post messages/replies and upload file attachments with their messages. Web based assessments/assignments can take the form of timed true or false, multiple choice, text input and answer upload questions, with online results posting and automated marking.

Manhattan Virtual Classroom [37] was developed at the Western New England College and it has been released under the Open Source General Public License for free. Anyone can download the complete, fully functional system, along with the source code, and it includes chat line, areas for teachers to post syllabus and other handouts, grade modules, and email system. Manhattan is open and non-proprietary.

IntraLearn [38] integrates remote registration, lessons, interactivity, streaming multimedia, communications, testing, tracking and reporting to provide a secure and self-contained online education solution.

Other tools include Authorware 5.2 [39], Illuminatus Opus [40], Kallidus 6 Authoring System [41], to name a few.

## 2.4 Evaluation of Two Specific Tools: WebCT and Blackboard

For the purposes of the current study, two tools are evaluated for accessibility from two perspectives: authoring and partial user agent. WebCT has a large user base from its early days as a shareware Unix-based product and it has been adopted by the University of Victoria. However, Blackboard has been installed at a large number of academic institutions across North America.

These tools have been evaluated for accessibility compliance with Section 508 of the Disability Act. This accessibility study is made from an authoring and user agent perspectives with consideration to the learnability and operability characteristics. The qualitative results for WebCT and Blackboard are provided in Tables 2.1 - 2.6. The evaluation has been based mostly on the information available in the open literature and also from the experience gained and observations during the implementation of WebCT for the case study outlined in Chapter 4. The results of this evaluation based on the WCAG guidelines and Section 508 are explained next:

### 2.4.1 Authoring Issues

1. **WCAG (1.1)/Section 508(a) - A text equivalent for every non-text element shall be provided**: Text equivalent means adding words to represent images, applets, sounds, multimedia presentations, etc. Text is considered accessible to almost all users since it can be read by screen readers and braille readers and it may be magnified for partially blind users. The text description needs to describe the function or purpose of the image. The text equivalent may be entered using the "alt", and "longdesc" commands. Both WebCT and Blackboard

interface images have alt tags supplied and this feature allows screen readers to interact with the software. However, Blackboard has an added feature as it prompts for a text equivalent when a course designer adds non-textual content to a course. WebCT puts the onus on the developer to comply with this requirement.

2. **WCAG (1.2)/Section 508(e) - Redundant text links shall be provided for each active region of a server side image map**: An "image map" is a picture on a web page that provides different links to other web pages. By clicking on a location of a server-side image map only specifies the coordinates within the image when the mouse is activated. The ultimate selection of the link must be deciphered by the computer serving the web page. This guideline is necessary because when a web page uses a server-side image map to present the user with a selection of options, browsers cannot indicate to the user the link that will be followed when a region of the map is activated. Therefore, the redundant text link is necessary for people with visual impairment to provide access to the page for anyone not able to see or accurately click on the map. In this particular case, it was found that neither WebCT nor Blackboard use server-side image maps.
3. **WCAG (1.4)/Section 508(b) - Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation**: This guideline requires that when an audio portion of a multimedia production is captioned, the captioning must be synchronized with the audio. Synchronized captioning is necessary so that someone reading the captions could also listen to the speaker and other associated relevant body language with the speech. Synthesized speech is critical for blind people and for many people with cognitive and learning disabilities, and deafness. Braille is essential for individuals who are

both deaf and blind, as well as many individuals whose only sensory disability is blindness. In compliance with this guideline, neither tool interface contains multimedia components that requires synchronization. However, if the user requires support for linked multimedia files, both tools provide capabilities to synchronization between alternative equivalents.

4. **WCAG (2.1)/Section 508(c)** - Ensure that all information conveyed with color is also available without color, for example from context or markup.: people who are color blind or with visual impairment may find the web page unusable when colors are used as the sole method for identifying screen elements or controls[42]. From an authoring perspective, for example, when asking for input from users, do not write "Please select an item in the green box." Instead, ensure that information is available through other style effects (e.g., a font effect) and through context (e.g., comprehensive text links). Blackboard provides documentation to course developers on best practices to meet the color requirements. Both tools allow the author to enter markup or upload external documents regardless of whether content is structured and separate from presentation.
5. **WCAG (5.1)/Section 508(g)** - For data tables, identify row and column headers. Tables should be used only for data tables and not to organize pages. Some browsers allow users to navigate among table cells and access header and other table cell information. However, unless marked-up properly, these tables will not provide the intended information. This guideline is of greatest benefit to people who access a table using screen readers and braille displays. If there is a need to present data in a tabular format, the author should use the HTML TABLE element and its supporting elements and attributes (like TR, TD, TH and CAPTION). The alternatives (such as using the PRE tag for pre-formatted

text or using style sheets) will actually make understanding of tabular data more difficult for people using assistive devices. Both WebCT and Blackboard indicate that they have added row and column header to their tables such as search results in order to take advantage of new screen reader capabilities. However, in terms of uploaded material and learnability perspective, they still present shortcomings.

6. **WCAG (5.2)/Section 508(h)** - For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells: When tables have structural divisions beyond those implicit in the rows and columns, the authors should use appropriate markup to identify those divisions. Large tables of data can be difficult to interpret for people using assistive devices. Use of the "scope" attribute is one of the most effective ways of making HTML compliant with these requirements. As mentioned above, both WebCT and Blackboard have added row and column headers to data tables in order to take advantage of newer screen reader capabilities. The tools will also add additional markup if there is uncertainty in the table data structure. However, they do not flag the uploaded tables and this can be potentially problematic since the onus is on the author of the tables to catch these potential issues.
7. **WCAG (6.1)/Section 508(d)** - Organize documents so they may be read without style sheets: Style sheets enable users to define specific viewing preferences to accommodate their specific disability. For example, users with low vision may create their own style sheet so that, regardless of what web pages they visit, all text is displayed in an extra large font with white characters on a black background. For good access, it is critical that designers ensure that their web pages do not interfere with user-defined style sheets. By default, the users should be allowed to use "external" style sheets. Both authoring tools

indicate that their application is readable without associated style sheets. With respect to style sheets, both tools satisfy the operability characteristic but they fall short on the learnability feature.

8. **WCAG (7.1)/Section 508(j)** - Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz: People with photosensitive epilepsy can have seizures triggered by flickering or flashing [43]. Flashing and flickering elements on a web page are usually present in animated gifs, Java applets, and third-party plug-ins. Neither WebCT nor Blackboard have components that cause the screen to flicker, but uploaded files are not checked by the tools and may not comply with this guideline.
9. **WCAG (9.1)/Section 508(f)** - Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape: There are two basic types of image maps: "client-side image maps" and "server-side image maps". Client-side image maps can be assigned their own link that specifies what web page to retrieve when a portion of the picture is selected. The client-side image maps permit the author to assign text to each image map "hot spots", unlike server-side image maps which are discussed above for WCAG1.2-Section 508(e). This means that a screen reader can easily identify and activate regions of the map. Both authoring tools use only client-side image maps., but there is no indication on the tool developers strategy regarding the learnability characteristic.
10. **WCAG (11.4)/Section 508(k)** - A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page

changes: As it is difficult to keep alternative pages up to date with the full content of the original page, alternative pages should be provided only after the author has tried all of the other pertinent techniques outlined in the Web Content Accessibility Guidelines to make the original page accessible (unless the alternative page is automatically generated from the same source as the original page). Both WebCT and Blackboard currently do not require nor do they support text-only pages and also they will not check pages that do not comply with the guidelines.

11. **WCAG (12.1)/Section 508(i) - Frames shall be titled with text that facilitates frame identification and navigation**: On the one hand, frames are very useful as they provide a means of visually dividing the viewing area into areas that can be separately rewritten. On the other hand, frames can also present difficulties for users with disabilities as frames are not easily identifiable when using assistive devices. For example, it is common to use "navigational bars" in a fixed position on the screen and have the content of the web site retrievable by activating one of those navigational buttons. The new content is displayed in another area of the screen. Since the navigational bar does not change, it provides a stable "frame-of-reference" for users and makes navigation much easier. Unfortunately, blind people may become lost if the differences between the two frames are not clearly established. The most obvious way to accomplish this requirement is to include text within the body of each frame that clearly identifies the frame. WebCT's frame sets are titled to facilitate identification and navigation. It became clear from the user study presented in Chapter 4 that although the frames were named, blind users still had difficulty with the navigation between the frames. WebCT's interface was accessible and a list of available frames was present, and the screen reader did not have difficulty

identifying the "title" of the frames, however the naming conventions was an issue for the users. Blackboard states that frame sets have supplemented their original name attribute with more descriptive title attributes. They have also added text describing the frame set and tried to enhance accessibility with links to help documentation. In WebCT, for navigation links located in the body of the main content page, code has been added to allow screen readers to detect and skip the navigation links. Blackboard isolates navigation elements in a separate frame, allowing users with screen readers to directly access the main content of a page without listening to the navigation links each time they access a page.

## 2.4.2 Partial User Agent

As mentioned earlier, the teaching and learning tools can assume the functionality of either an authoring tool and/or a partial user agent. In Section 508, Paragraph 1194.22 there are five additional guidelines that tool developers must adhere to to support accessibility and these are outlined in §1194.22 (l,m,n,o,p). These sub-paragraphs deal mostly with rendering and receiving information such as scripts, applets and forms.

1. **WCAG (6.3, 6.4, 8.1 and 9.3)/Section 508 §1194.22 (l) - When pages use scripting language to display content, the information provided by the script shall be identified with functional text that can be read by assistive technology:**  
WebCT does not use dynamic scripting for content presentation, however it does use Java script to conditionally write some content and navigation to a page. Blackboard uses only minimal dynamic scripting for content presentation.
2. **WCAG (6.3, 6.4, 8.1)/Section 508 §1194.22 (m) - When a page requires**

that an applet, plug-in or other application be present on the client system to interpret the page content, the page must provide a link to a plug-in or applet that complies with §1194.21 (a) through (l): WebCT supports plug-ins as embedded content in WebCT pages, however it is necessary for the plug-ins to be installed on user's computers in order for them to view the material. WebCT has worked to make the Chat and Whiteboard tools accessible by providing an accessible interface to navigate. Blackboard supports many plug-ins as embedded content in Blackboard pages, or as automatically launched files. Also, it is necessary for the plug-ins to be installed on user's computer in order for them to view the material.

3. **WCAG (10.2, 12.4, 9.3)/Section 508 §1194.22 (n) - When electronic forms are designed to be completed online, the form shall allow users with assistive technologies to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues:** a person using assistive technology should be able to identify the purpose of any form control element and to be able to manipulate it. WebCT has a form label next to the form input elements that are referenced including input boxes and radio buttons. This allows screen reader users to appropriately associate the labels with the form element. The forms can be activated by keyboard or by mouse input. Blackboard uses forms to help authors upload course content. The forms are made accessible by placing form labels adjacent to the form input elements they reference. In general, text labels immediately precede text input boxes, follow radio button options, and are directly above text area inputs. This allows screen readers to appropriately associate the labels with the form element.
4. **WCAG (13.5, 13.6)/Section 508 §1194.22 (o) - A method shall be provided**

that permits users to skip repetitive navigation links: The "skip navigation" provision of the Section 508 is related to a couple of WCAG P3 checkpoints, but the Section 508 standard is specific and direct. The WCAG checkpoints assume technology is not yet supported, like grouping and labelling links. WebCT has added code to allow screen readers to detect and skip navigation links. Blackboard isolates navigation elements in a separate frame, allowing users with screen readers to directly access the main content of a page without listening to the navigation links each time they access a page. For navigation links located in the body of the main content page, such as Breadcrumb navigation, code has been added to allow screen readers to detect and skip the navigation links.

5. **Section 508 §1194.22 (p)** - When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required: There are no comparable checkpoints in the WCAG. Both WebCT and Blackboard provide the option of setting time limits on quizzes. When the time limit is set, the student is warned with a pop-up alert when time is running out. If a student does not submit the assessment before the time limit expires, the assessment is flagged in the grade book. An instructor must then review flagged assessments and determine an appropriate action (for example, docking the grade, or in the case of a student with disabilities, dismissing the flag).

To summarize the accessibility evaluation, authoring tool developers for WebCT and Blackboard have tried to incorporate Section 508 guidelines and their effort to address the issue of accessibility for inclusiveness is acknowledged. Both tools have made significant progress in terms of operability of the interface as they have complied with the guidelines for their interface, but they have not addressed the issue of user uploaded information. Also, they still need to work on learnability issues. Layout and aesthetics of WebCT are pleasing to the eye, but those using adaptive

technologies may experience difficulty navigating a page. Blackboard uses pop-up alert boxes to inform the user of special conditions such as missing information in a form being submitted. When authors utilize scripting languages to display content, or to create interface elements, the information provided by the script is identified with functional text that can be read using assistive devices. Significant improvements in the accessibility compliance of both tools in the most recent releases have been reported, however these have not been verified by the current study.

Table 2.1: Evaluation for accessibility of web-based authoring tools

Criteria			Sub-characteristics	
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability
<b>Guideline 1: provide equivalent alternatives to auditory and visual content.</b>				
1.1 (a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).	WebCT	All WebCT images have alt tags supplied, this allows screen reader users to interact with the software. Content developers should support this standard with Alt Tags included in image files imported into WebCT.	full	none
	BB	All Blackboard images have alt tags supplied. Allows screen reader users to interact with the software. The software prompts the user for a text equivalent when a course designer adds non-textual content to a course.	full	full
1.2 (e) Redundant text links shall be provided for each active region of a server-side image map.	WebCT	WebCT does not use server-side image maps.	partial	none
	BB	Does not use server-side image maps.	partial	none
1.4 (b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.	WebCT	WebCT does not use multimedia that requires synchronization. However, WebCT does support linked multimedia files containing synchronized equivalent alternatives.	partial	none
	BB	Encourages the creation of equivalent content by allowing the course builder to create content folders to group images, videos, sound files, text files, and other rich content. This creates an environment where learners of many types and abilities can absorb and comprehend course information.	full	none
<b>Guideline 2: Don't rely on color alone: Ensure that text and graphics are understandable when viewed without color.</b>				
2.1 Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.	WebCT	Web pages can be designed so that all information conveyed with color is also available without color.	partial	none
	BB	All Blackboard images have alt tags supplied. Allows screen reader users to interact with the software. The software prompts the user for a text equivalent when a course designer adds non-textual content to a course.	partial	partial

Table 2.2: Evaluation for accessibility of web-based authoring tools (cont.)

Criteria			Sub-characteristics		
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability	
<b>Guideline 3: Use markup and style sheets and do so properly.</b>					
<b>Guideline 4: Clarify natural language usage.</b>					
<b>Guideline 5: Create tables that transform gracefully.</b>					
5.1	Row and column headers shall be identified for data tables.	WebCT	The tool has added row and column headers to data tables in order to take advantage of newer screen reader capabilities to read this additional information.	partial	none
		BB	Has added row and column headers to data tables (such as search results) in order to take advantage of new screen reader capabilities to read this additional info.	partial	none
5.2	Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.	WebCT	WebCT has added additional markup where there is uncertainty in the table data structure.	partial	none
		BB	Implemented according to W3C guidelines.	none	none
<b>Guideline 6: Ensure that pages featuring new technologies transform gracefully.</b>					
6.1	Documents shall be organized so they are readable without requiring an associated style sheet.	WebCT	This Application is readable without an associated style sheet.	partial	none
		BB	The application is readable without an associated style sheet.	partial	none
<b>Guideline 7: Ensure user control of time-sensitive content changes.</b>					
7.1	Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	WebCT	WebCT does not include components that cause the screen to flicker.	partial	none
		BB	Does not include components that cause the screen to flicker. Also provides documentation to course builders indicating best practices to meet this guideline for uploaded content.	partial	partial

Table 2.3: Evaluation for accessibility of web-based authoring tools (cont.)

Criteria			Sub-characteristics		
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability	
<b>Guideline 8: Ensure direct accessibility of embedded user interface.</b>					
<b>Guideline 9: Design for device-independence.</b>					
9.1	Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	WebCT	WebCT uses client-side image maps only.	partial	none
		BB	Uses client-side image maps only.	partial	none
<b>Guideline 10: Use interim solutions.</b>					
<b>Guideline 11: Use W3C Technologies and guidelines.</b>					
11.4	A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the	WebCT	WebCT does not currently require text-only pages.	none	none
		BB	Does not currently include text-only pages. If functionality is added to the platform that cannot be rendered in accordance with the 508 standards.	none	none
<b>Guideline 12: Provide context and orientation information.</b>					
12.1	Frames shall be titled with text that facilitates frame identification and navigation.	WebCT	WebCT frame sets are titled to facilitate identification and navigation.	partial	none
		BB	Frame sets have supplemented their original name attribute with more descriptive title attributes. There is also added text describing the frame set and links to Help documentation.	full	partial
<b>Guideline 13: Provide clear navigation mechanisms.</b>					
<b>Guideline 14: Ensure that documents are clear and simple.</b>					

Table 2.4: Evaluation for accessibility of web-based learning tools - partial user agent

Criteria			Sub-characteristics	
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability
<p>Others: Paragraph (l);(m);(n);(o) relate to scripts, applets and forms (partial user agent). Conformity to WCAG 1.0 Level A means meeting all P1 issues. To satisfy Section 508 Guidelines, the paragraphs l,m,n,o,p have to be implemented, in addition.</p>				
<p>6.3 6.4 8.1 9.3</p> <p>Section 508(l): When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.</p>	<p>WebCT</p>	<p>WebCT does not use dynamic scripting for content presentation. It does use java script to conditionally write some content and navigation to a page. Content rendered in this manner is written to the screen before the page loads. The rendering of the output content is seamless to the user.</p> <p>WebCT uses pop-up alert boxes to inform the user of special conditions (such as missing information in a form being submitted). Neither of these uses violates Section 508.</p>	<p>partial</p>	<p>none</p>
	<p>BB</p>	<p>Software uses only minimal dynamic scripting for content presentation. It does use script to conditionally write some content to a page. Content rendered in this manner is written to the screen before the page loads, and can not be distinguished from standard content. In addition, Blackboard uses pop-up alert boxes to inform the user of special conditions (such as missing information in a form being submitted). When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script is identified with functional text that can be read using adaptive technologies.</p>	<p>partial</p>	<p>partial</p>

Table 2.5: Evaluation for accessibility of web-based learning tools - partial user agent(cont.)

Criteria			Sub-characteristics	
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability
6.3 6.4 8.1	WebCT	Supports plug-ins as embedded content in WebCT pages, or as automatically launched files. However, it is necessary for the plug-ins to be installed on users' computers in order for them to view the materials. WebCT has worked to make the Chat and Whiteboard tools accessible by providing an accessible interface to navigate chat applet starting and navigation. If necessary, alternate chat and whiteboard tools with additional accessibility provisions may be incorporated into the course environment.	partial	partial
	BB	Supports many plug-ins as embedded content in Blackboard pages, or as automatically launched files. However, it is necessary for the plug-ins to be installed on users computers in order for them to view the materials. Also, provides description fields to allow course builders to link to the appropriate plug-in.	partial	partial

Table 2.6: Evaluation for accessibility of web-based learning tools - partial user agent (cont.)

Criteria			Sub-characteristics	
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability
10.2 12.4 9.3 Section 508(n): When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.	WebCT	Placed a form label next to the form input elements that are referenced including input boxes and radio buttons. This allows screen reader users to appropriately associate the labels with the form element. The forms can be activated either by keyboard or by mouse input.	partial	partial
	BB	Uses forms to help course builders upload course content. The forms have been made accessible by placing form labels adjacent to the form input elements they reference. In general, text labels immediately precede text input boxes, follow radio button options, and are directly above text area inputs. This allows screen reader users to appropriately associate the labels with the form element. JavaScript is used for form validation, which improves performance of the system. The validation logic runs in the background.	full	partial

Table 2.7: Evaluation for accessibility of web-based learning tools - partial user agent (cont.)

Criteria			Sub-characteristics		
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability	
13.5, 13.6	Section 508(o): A method shall be provided that permits users to skip repetitive navigation links.	WebCT	For navigation links located in the body of the main content page, such as the Breadcrumb navigation, code has been added to allow screen readers to detect and skip the navigation links.	partial	partial
		BB	Isolates navigation elements in a separate frame, allowing users with screen readers to directly access the main content of a page without listening to the navigation links each time they access a page. For navigation links located in the body of the main content page, such as the Breadcrumb navigation, code has been added to allow screen readers to detect and skip the navigation links.	full	partial

Table 2.8: Evaluation for accessibility of web-based learning tools - partial user agent (cont.)

Criteria			Sub-characteristics	
Web Content Accessibility Guideline (WCAG 1.0)	Tools	Comments	Operability	Learnability
Section 508(p): When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.	WebCT	Tool provides the option of setting time limits on Quizzes.If a student submits a quiz after the time limit, the instructor can opt to review the reasoning and opt to accept late submissions, thus allowing students with disabilities to have unlimited time for completion.tool also creates a "session" when a user logs in to their course, for tracking, settings and security reasons. A standard session time-out period is two hours. The session time-out setting can be modified by the system administrator to allow for more time if necessary. If a session does time out, the user must log into the system again before continuing to interact with the application.	full	full
	BB	Instructors have the option of setting time limits on assessments. When a time limit is set, the student is warned with a pop-up alert when time is running out. If a student does not submit the assessment before the time limit expires, the assessment is flagged in the grade book. An instructor must then review flagged assessments and determine an appropriate action (for example, docking the grade, or in the case of a student with disabilities, dismissing the flag).	full	full

## 2.5 Verification and Repair Tools

In an Integrated Design Environment (IDE), all authoring tools would provide utilities to aid in creating accessible online information. However, as yet, only preliminary steps towards creation of accessible web content are evident in the authoring tools currently on the market.

In the absence of an IDE, evaluation and repair tools have been created that examine the accessibility of a web site after it has been created. Verification and repair tools perform a static analysis of pages or sites regarding their accessibility, and return a report and a rating.

There are a few commercially available verification and repair tools such as Bobby, A-Prompt, InFocus, AccVerify and LIFT.

Bobby [44] is a verification only tool that checks for compliance with WCAG recommendations. However, this tool does not repair the content for accessibility and only identifies the compliance violations. It has been developed in Java and available for different platforms. The first version of Bobby was released in 1996. The most recent release (Version 5.0) presents a much improved and user-friendly interface compared to older releases. Bobby produces a large report, much of which consists of recommendations for manual checks by the site's author. A summary report for the entire web material is created.

A-Prompt (Accessibility Prompt) Toolkit [45] is a repair utility and is the product of a collaboration between the Adaptive Technology Resource Centre (ATRC) at the University of Toronto and the Trace Center at the University of Wisconsin and it is freeware. A-Prompt allows the author to select a file for validation and repair, or select a single HTML element within a file. If an accessibility problem is detected, A-Prompt displays the necessary dialogs and guides the user to fix the problem. Many

repetitive tasks are automated, such as the addition of ALT-text or the replacement of server-side image maps with client-side image maps.

AccVerify and AccRepair [46] is a verification and repair software suite developed by HiSoftware. It generates a report with statistics and graphics. The reports generated include a list of all possible errors for the compliance standard, along with a pass and fail notation for each. Automatic repair functions are available, but the software does not advise the user of the specific error being corrected.

InFocus, by SSB Technologies [47], finds and repairs issues related to Section 508 only. Once a file is loaded, a very useful feature is the ability to view the source code within the interface. InFocus allows for an entire web site to be checked and can create reports in multiple formats. Like AccVerify, it is appropriate for large-scale projects and monitoring of large quantities of HTML formatted data.

In January 2002, UsableNet announced the release of LIFT [27], an add-on product to Macromedia Dreamweaver [48]. LIFT enables the process of compliance with W3C and Section 508 guidelines. Macromedia claims that evaluating web pages in monitor mode automates accessibility and saves development time. Tables, images, scripts and links are managed using Fix Wizard, which is a step-by-step tool that guides users through creating accessible content. LIFT is comprised of two programs.

The main body of the system is the Dreamweaver. The main interface holds the plug-in LIFT. As the developer creates a new web application or site they can simultaneously run LIFT as they develop their new system. LIFT is based on a plug-in that is comprised of applets that serve different functions for monitoring certain web standards while the developer is creating their web application or site. Other formats like JSP, JavaScript, XML, HTML are some of the formats that are compatible with LIFT.

Bobby, A-Prompt, AccVerify, InFocus and LIFT are all valuable tools to enhance

accessibility of web content. If the needs of the developer are simple, for a small to medium web site and a small budget, A-Prompt and Bobby are recommended. However, A-Prompt will not satisfy the needs if a report with errors is required and Bobby does not repair errors. For a large web site and if the evaluation and repair process is extensive, AccVerify, InFocus and LIFT should be explored. A more recent study at the University of Toronto provides more details on some of these tools [49].

## 2.6 Synopsis

Ideally, all authoring tools in an integrated design environment would provide utilities and information to prompt creation of accessible resources. In the absence of integrated tools designed for writing accessible HTML code, verification and repair tools are necessary to examine the accessibility of web content after it has been created.

WebCT and Blackboard have their distinct capabilities and characteristics. The review and evaluation of the authoring tools was based on the Section 508 Priority 1 guidelines. Both WebCT and Blackboard have made significant progress in the last few years to address the issue of accessibility for users with visual impairment. However, they lack the learnability features and do not ensure compliance of the user-uploaded material.

The brief review on some of the verification and repair tools provides useful information for authors. In a nutshell, for simple needs, A-Prompt and Bobby are quite adequate, while for large or multiple web sites, AccVerify, InFocus and LIFT are a more suitable choice. LIFT has been selected to verify and repair a course web content in the Department of Computer Science at UVic. This is a necessary step for the implementation of a case study for users with visual impairment and to evaluate the accessibility of the material.

The next chapter provides an overview of the assistive technologies that people with disabilities use to access web-based information.

## Chapter 3

# Assistive Technologies

Computers have made much of the work that people do easier and less time-consuming, and adaptive devices offer people with disabilities the opportunity to use computers and to complete tasks that were previously not possible for them.

For blind people, these include voice recognition systems, alternative pointing devices, screen reader/ magnifiers, and other devices. For example, a computer and a screen reader can give a blind person access to online books and journals; supplying a screen reader, a trackball instead of a mouse, or an infrared keyboard means people with disabilities can have complete access to computers and on-line information.

In this presentation, we are concentrating primarily on the output devices.

## 3.1 Types of Assistive Devices for the Blind

### 3.1.1 Screen Readers

Screen readers are designed for people who are blind or visually impaired. This technology finds information on the computer screen and communicates it to the user. It allows access to the information on the screen with the keyboard only. This task gets accomplished either with text-to-speech software or hardware such as a refreshable Braille display (a display that uses pins that move up and down to spell out words in Braille - the primary means of access to computers for users who are deaf-blind).

It is important to note that screen readers work best with information that is text. Images, image maps, animations, video and audio can be translated if text is written to go with them. Screen readers are designed to work best with standard HTML. Use of properly marked up HTML will significantly improve the usability of Web sites for screen reader users.

A Screen Reader is the commonly used name for Voice Output Technology. Hardware and software produce synthesized voice output for text displayed on the computer screen, as well as for keystrokes entered on the keyboard.

One of the leading screen readers in the market is "JAWS for Windows" [50] which is a product of Freedom Scientific and works closely with Microsoft products, such as MS Office, IE and MS project, and mail applications such as Eudora and MS Outlook. JAWS works with a PC to provide access to software applications and the Internet and works within the operating systems itself. Jaws has its own software voice synthesizer "Eloquence" that runs on the sound card.

With its internal software speech synthesizer and the sound card, information

from the screen is read aloud. JAWS also outputs to refreshable Braille displays. In JAWS 4.01, Windows visual effects that are less than optimal for screen readers are automatically disabled when JAWS starts, and are enabled again when JAWS is closed. These effects are visual features, such as animated menus within Windows XP.

In version 4.5, JAWS now supports navigation of web pages by headings in Internet Explorer 5 and later, and in Microsoft HTML help systems. JAWS recognizes HTML headings at levels 1 through 6. Versions of JAWS 3.7 and up are optimized to take advantage of newer accessibility features in IE 5.5. Newer versions of Jaws support frames, forms, javascript alert boxes, graphics (via alt tags), and multi-columns.

JAWS now supports the "Longdesc" attribute in HTML within Internet Explorer 5 and later. In version 4.5 JAWS is able to go through java code and scripting languages, and it relies on the HTML 4.0 standards.

### **3.1.2 Screen Magnification**

Screen magnification software is used by people with partial visual impairment to access information on a computer screen. The software enlarges the information on the screen by a pre-determined incremental factor. Magnification programs run simultaneously and seamlessly with the computer's operating system and applications.

Most screen magnification software has the flexibility to magnify the full screen, parts of the screen or provide a magnifying glass view of the area around the cursor or pointer. These programs also often allow for inverted colors, enhanced pointer viewing and tracking options. The challenge in the design of these screen magnifiers is to track the cursor or the active region of the screen in order to automatically enlarge the portion of the screen the user needs to see. Therefore, software that uses

a custom cursor may pose a challenge for accessibility since the wrong portion of the screen may be magnified.

The two major software in the markets are MAGic [51] and Zoomtext [52]. Freedom Scientific has designed MAGic, and it is compatible with the JAWS screen reader which includes extensive speech and braille display support. MAGic also has option for inverted colour and cursor enhancements. MAGic 6.x for Windows 2000 does not currently have its own screen reading capabilities and must be used with a screen reader such as WindowEyes or JAWS (if a computer user needs synthesized speech).

Ai Squared designed Zoomtext. The latest version, ZoomText 8.0, enlarges, enhances and reads aloud everything on the screen and it gives access to documents, email and the Internet. It comes in two products; a magnifier which is a standalone screen magnifier, and the magnifier/screen reader. The 8.03 release also offers improved compatibility with other adaptive technology products such as JAWS, improved video card support, and greater stability in Windows 98 and Windows Millennium.

### 3.1.3 Refreshable Braille Displays

Refreshable Braille Displays [53] are electronic devices that are used to read text that a computer sends to the monitor, as illustrated in Figure 3.1 . The device is connected to the computer by a serial cable and produces Braille output on the Braille display. Refreshable Braille displays only read one line of text at a time . These displays generally include directional keys which allow the user to navigate through a document. Larger displays also include a cursor routing function. The Braille instantly translates screen data into 6-dot or 8-dot refreshable Braille for those using software screen readers, such as JAWS for Windows. Braille characters based on a

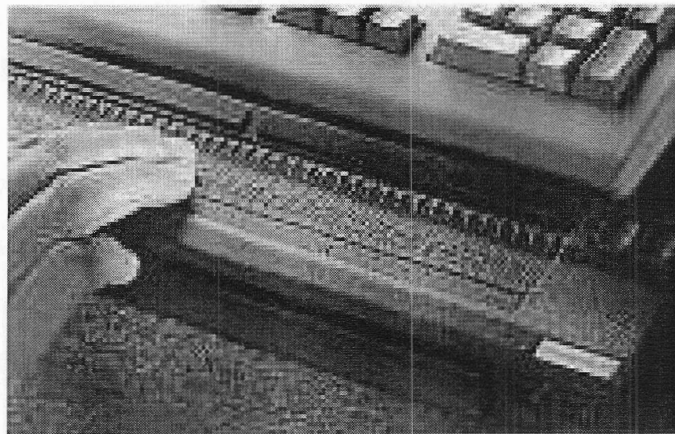


Figure 3.1: Braille display

6-dot braille cell have two parallel columns of three dots each. 8-dot braille is used for some special purposes and computer online refreshable braille displays. 8-dot braille cells have two columns of four dots. 8-dot braille has 256 unique patterns if the blank cell is counted.

Text-to-Speech software is used to convert words from a computer document (e.g. word processor document, web page) into audible speech spoken through the computer speaker. This would be helpful to people who need or want aural verification of what they are seeing in print. Text-to-speech technology can be integrated with optical character recognition systems. Text to Speech software is different from Screen Reading technology in that it does not read any system information (such as file structure or alter boxes). Note that with software that produces a "sound" or voice output, you also need a PC compatible sound card.

### 3.1.4 Braille Embossers

A Braille Embosser is a hardware device for "printing" a hard copy of a text document in Braille. A Braille translation software program is required to translate the text from the computer into Braille. Most Braille translation software programs can translate material into several grades or versions of Braille. There is also a new Music Braille software program as well as scanning software to scan Braille into a computer and have it displayed or translated into text.

### 3.1.5 Haptic Devices

A haptic interface is a device which allows a user to interact with a computer by receiving tactile feedback. There are two main types of haptic devices: the glove/pen-type devices that allow the user to "touch" and manipulate 3-dimensional virtual objects and devices that allow users to "feel" textures of 2-dimensional objects with a pen or mouse-type interface and these can be used to aid computer users who are blind or visually disabled; or who are tactile or kinesthetic learners, by providing a slight resistance at the edges of windows and buttons so that the user can "feel" the Graphical User Interface (GUI). This technology can also provide resistance to textures in computer images which enables computer users to "feel" pictures such as maps and drawings. Examples of haptic devices that include the iFeel Mouse and the iFeel MouseMan [54].

### 3.1.6 Voice Recognition

Voice recognition allows a user to use his/her voice as an input device. Voice recognition may be used to dictate text into the computer or to give commands to the

computer (such as opening application programs, pulling down menus, or saving work). Older voice recognition applications require each word to be separated by a distinct space. This allows the machine to determine where one word begins and the next stops. This style of dictation is called discrete speech. Many people (especially those with learning disabilities) prefer these systems to the newer continuous speech.

Continuous speech voice recognition applications allow a user to dictate text fluently into the computer. These new applications can recognise speech at up to 160 words per minute. While these systems do give the user system control they are not yet hands free. Voice recognition uses a neural net to "learn" to recognize your voice. As you speak, the voice recognition software remembers the way you say each word. This customization allows voice recognition, even though everyone speaks with varying accents and inflection. While the accuracy of voice recognition has improved over the past few years some users still experience problems with accuracy either because of the way they speak or the nature of their voice. Voice Recognition for Windows includes "Dragon Naturally Speaking Professional" [55] from Dragon Systems.

## 3.2 Synopsis

In Chapter 2, we focused on the authoring tools and the role they play in the integrated development environment. In order to make the content accessible to users with visual impairment, the user needs to integrate their respective adaptive technology. For example, totally blind users require the refreshable braille display or JAWS as a screen reader. Partially blind users require a screen magnifier such as Zoomtext to view parts of the screen at a time. In this chapter, we have presented some of the currently available adaptive technologies that have been used in our case study described in Chapter 4. In the next chapter, we now present the methodology used to

qualitatively evaluate the accessibility of the integrated environment composed of the authoring tool (Dreamweaver), the evaluation and repair tool (LIFT), the learning tool (WebCT) and the adaptive technologies (JAWS, Zoomtext and Braille Display)

## Chapter 4

# A Case Study for the Visually Impaired Using WebCT

So far we have presented, analyzed and provided a qualitative evaluation of web-based authoring tools, identified verification and repair tools that are necessary to make web content accessible to users using adaptive technologies, and described adaptive devices that enable visually impaired users to access web content.

This chapter builds upon the information collected in the previous chapters and a test case study of an online course available to students at the University of Victoria has been designed and implemented. To that end, four subjects representing the spectrum of visual disabilities were selected to test the online course for accessibility. One of the goals was to learn how people with low vision and no vision use assistive technologies and the web. From a developer's point of view, in order to make the course content accessible to users with visual disability, the course was verified and repaired for accessibility using LIFT. From the user's perspective, each participant using his/her adaptive device of choice proceeded to access the online course using WebCT and the accessibility barriers experienced by blind users were recorded during

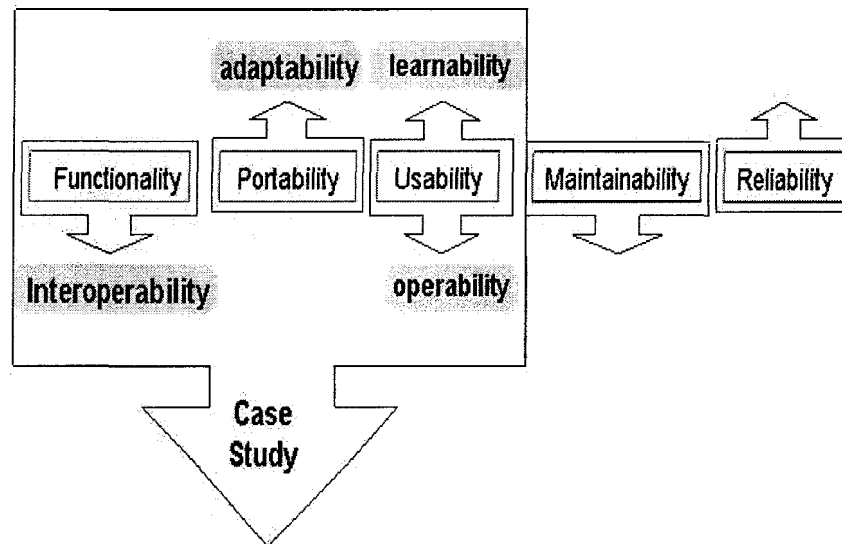


Figure 4.1: Accessibility sub-characteristics considered in the user study for the visually impaired.

the testing phase. The accessibility sub-characteristics considered in this study are illustrated in 4.1.

## 4.1 The Testing Environment

Figure 4.2 illustrates the steps taken to setup the testing environment. In the first step, the types of tools and technologies required by the visually impaired participants were identified for each user. The adaptive devices depended on the extent of disability. For instance, users with partial vision needed a screen magnifier device, while the totally blind required either a text-to-speech or braille devices.

In Steps II and III, the testing environment was defined and setup in terms of the operating system used, and the dependencies of the magnification and text-to-

speech technologies on the hardware. a layered and structured testing environment was designed and implemented to enable a systematic and progressive identification of barriers, as illustrated in Figure 4.3. For example, totally blind users need screen readers that are dependent on the sound card. The sound card in turn has to be compatible with the operating system and had to work concurrently with the screen reader. Next, we ascertained that when using different browsers, the screen reader was able to access the content of the web site.

The next step required verifying and repairing the content of the target course with adherence to the W3C guidelines. The last component added to the testing environment was the web-based learning and teaching tool (WebCT). The learning tool delivers the information to the end users. If all components of the testing environment do not integrate seamlessly, the flow of information to the user can not be ensured.

## 4.2 The Hardware and Software

During the implementation phase, the server was initially set up on a Pentium III PC with the Linux RedHat 6.0 Operating System installed in a laboratory setting in conjunction with the target (authoring) tool (WebCT 3.6). During the testing phase, the course content was moved to the UVic Computer Services Unix server running the Solaris Operating System.

On the client or user side, there were three personal computers configured to handle the various combinations of hardware and software in terms of the operating systems, the assistive device being used during the testing, and the browsers. The PCs were installed with different Windows Operating System (Windows 95, NT, and XP) and one or more of the selected assistive devices from the following categories:

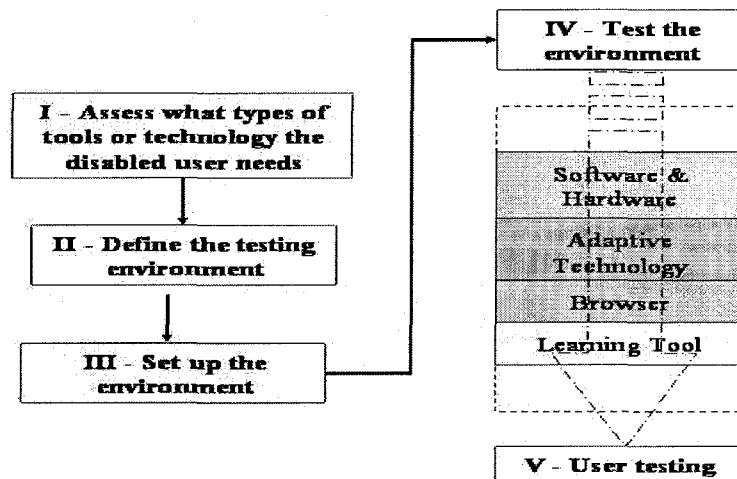


Figure 4.2: The methodology adopted for the user study

Screen Readers (Jaws 3.2, 4.0, 4.5); Screen Enhancers (ZoomText Extra 6.0, 7.0); and Refreshable Braille Display (Braille Note Lite).

Considerable effort was expended when installing and configuring the sound card, the network interface card and the braille device. Netscape and Internet Explorer 5 were the two browsers selected. To verify and repair the course content, the freeware tool A-Prompt 1.0 was used in the first instance, but we soon realized that the save function did not work properly and a decision was made to use a commercial product. Dreamweaver 4.0 with the LIFT plug-in was used instead.

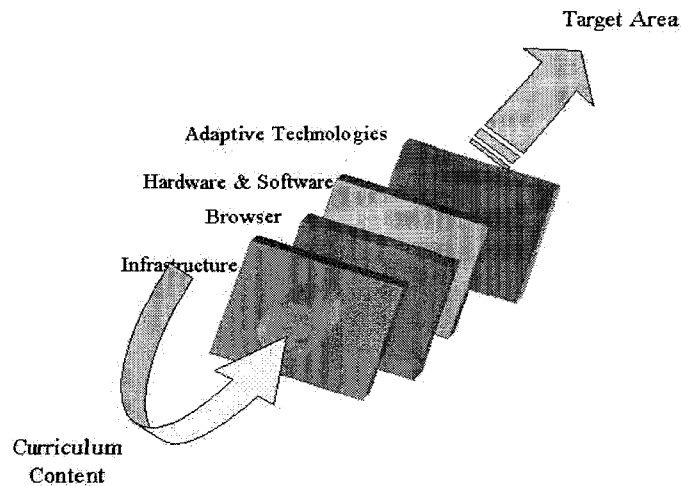


Figure 4.3: The filtered and layered testing environment

### 4.3 Material Selection

The Human Computer Interaction course (SENG310) was selected for this study ([www.csc.uvic.ca/mstorey/teaching/seng310](http://www.csc.uvic.ca/mstorey/teaching/seng310)) This is a third year Computer Science course and as mentioned earlier it was the subject of a usability study at UVic in 2000 [21].

### 4.4 Course Content Verification and Repair

From the authoring perspective, the course content needs to be verified and repaired in order to enhance accessibility. To that end, Dreamweaver 4.0 and LIFT were used. LIFT has two types of issues that arise when verifying the course content: Automatic

and manual accessibility issues. The tool-assisted (automatic) issues allow the use of the wizard applet to correct the HTML code. Please note that the automatic type issue does not imply that they are auto fixing, they just provide a template once the correct section of code has been selected, indicating the issue has been flagged.

The manual issues are suggestions put forward by the verification tool in order for the course designer to exercise judgment and take appropriate steps to correct the situation as deemed appropriate. For example, if no Java scripts links are used, LIFT flags this issue with an error message when using embedded Java scripts for the links. To resolve this issue, the user needs to re-design the layout and the structure of the links altogether.

#### 4.4.1 Tool-Assisted Issues

In our case study, the following tool-assisted (automatic) issues were repaired in order to enhance course accessibility for the visually impaired:

1. Data tables should have headers: (Section 508(g) - Priority 1) As the screen reader reads the header to identify the purpose of the table. LIFT automatically flagged for headers as the tables did not contain a header. To correct this error, the wizard, illustrated in Figure 4.4, allows the user to enter the summary for the table, the table headers and its scope. <sup>1</sup>
2. Cell of data should refer to headers: A reading order is essential when using screen readers such as Jaws. For data tables that have two or more logical levels of row or column headers, LIFT inputs markup to associate data cells

---

<sup>1</sup>The developer/author should always run the evaluation after each "change and save" operation because after a period of time, the changes can be lost.

CHAPTER 4. A CASE STUDY FOR THE VISUALLY IMPAIRED USING WEBCT59

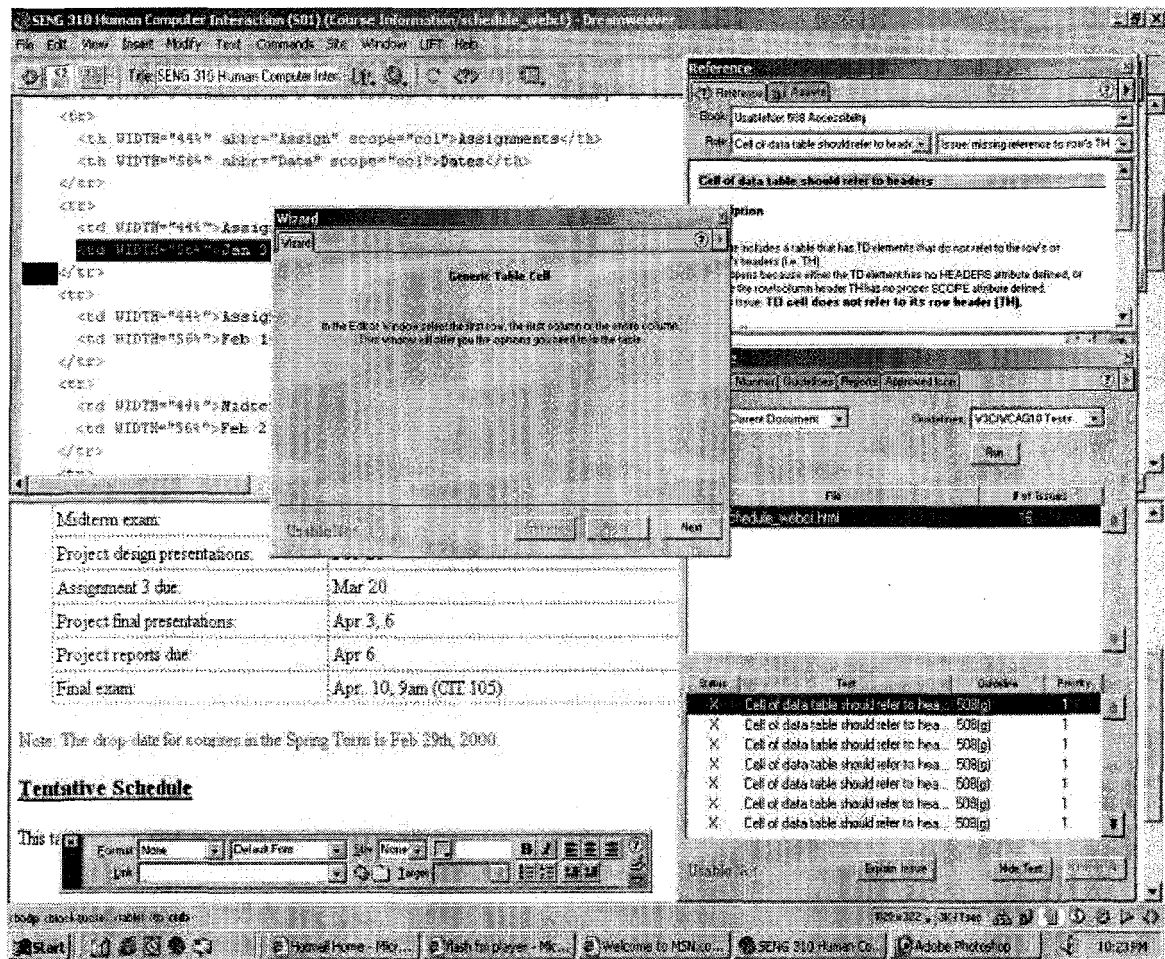


Figure 4.4: The LIFT error message templates showing the wizard applet, the evaluation applet and the reference box

## CHAPTER 4. A CASE STUDY FOR THE VISUALLY IMPAIRED USING WEBCT60

and header cells. Please note that this requirement is not explicit in the Section 508(g) but it is required by the W3C guidelines in the recommendations for aiding the reading order.

3. Skip repetitive links: the applet tried to find the exact position of the code fragment and it allowed us to enter the name of the anchor. After the information was entered correctly, LIFT warned that we had to manually enter the complimentary tag, which bypasses the group of links. Sometimes the evaluate applet does not find the exact position of the code fragment and this had to be done manually.
4. Using Images: The absence of an image descriptor is a priority 1 issue in Section 508. In order to enable a visually impaired user to access and understand the content of an image, there should be a description (LONGDESC/D-LINK and ALT tag) associated with the image. In LIFT, there are two ways of dealing with these issues: using the ALT editor and/or the wizard applet. The ALT editor brings up a list of all the images. The ALT editor attempts to solve this issue automatically by asking the author to fill in the text fields. Alternatively, to activate the wizard applet, the author needs to click on the image itself or the images code segment within the image tag to get the Wizard to respond. In this case, the author should fix the issue and save.

- (i) Image/thumbnail with invalid ALT tag: The best way to deal with this issue is to use the ALT editor. The user can then just fill in the text field provided by the ALT editor applet. When dealing with thumbnails, the user can either opt for the ALT editor or the wizard applet to provide the description of the image. <sup>2</sup>

---

<sup>2</sup>If the author decides to use the Wizard exclusively, the user should save and re-verify. Otherwise, use the (ALT editor) applet, which is excellent for alt tags but it does not solve LongDesc tags.

- (ii) Image/thumbnail with invalid LONGDESC/D-LINK: The author needs to provide a descriptive file for an image or a thumbnail. In this case, the ALT editor will not resolve this issue. The developer needs to use the wizard applet to include the descriptor file. <sup>3</sup>
- (iii) Bullet/spacer image with invalid ALT: Grouping items together using bullets or separating content using spacers presents an accessibility barrier to users with screen readers. This problem can be easily resolved by using the ALT editor to provide a blank string character or a space character.

#### 4.4.2 Manual Issues

When a manual issue is flagged, as shown in Figure 4.5, it is up to the author to confirm that the issue does in fact exist in the current page that they are working on.

To handle the manual issues in Dreamweaver 4.0/LIFT, there is an applet called Reference. It aids the author to fix issues and find information on the issue at hand. To enable the Reference applet, it must be selected from the window in the menu. In the Evaluate Applet, there is an "Explain Issue" button which provides further information to the author.

##### 1. Using Scripts and Applets:

- (i) Spacer with invalid Noscript: Scripts are mini-programs used to handle information submitted in web based forms, such as registration forms and quizzes. They are written in various programming languages such as Visual Basic, C++ and Perl. The course content included quizzes and as-

---

<sup>3</sup>Sometimes the evaluation applet finds the fragment but does not activate the Wizard template and just stays blank. Also, once the author repairs the first issue, and clicks the next button for correcting the next issue. the wizard will not work properly.

signment submissions. If the page has Noscript issues, the author has to make sure that the script does not have a link that it was processing. This error is an important aspect of site accessibility. <sup>4</sup>

- (ii) Applet with valid content: Applets are programs executed or launched from within a web page, and not all browsers support applets. The CONTENT was missing from the applet tag and needed to be added. Here, there was also a need to provide an equivalent alternative inHTML format.
- (iii) No JavaScript links are used: If the page has Javascript, it is necessary to create a logical tab order through links, form controls, and objects. LIFT flags this issue with an error message when using embedded JavaScript for the links. The best way to solve this problem is to re-design the layout altogether and the structure of the links.
- (iv) Script with no Noscript tag: When adding JavaScript to a web page, we needed to make sure to include a Noscript tag so that if the JavaScript can not be accessible then the user can access a new link within the tag or read information about what the JavaScript is supposed to do along with an optional link. The Wizard in this case will only display the selected JavaScript, which does not have the Noscript tag. The author has to come up with the code that will solve the problem.

---

<sup>4</sup>Other than finding out the actual content of the script just add a Noscript tag at the end of the script tag and within the Noscript tag. The author has to provide information as to what the tag does or provide a link that resembles a similar function of the script if applicable or include an email so that the user can submit the information.

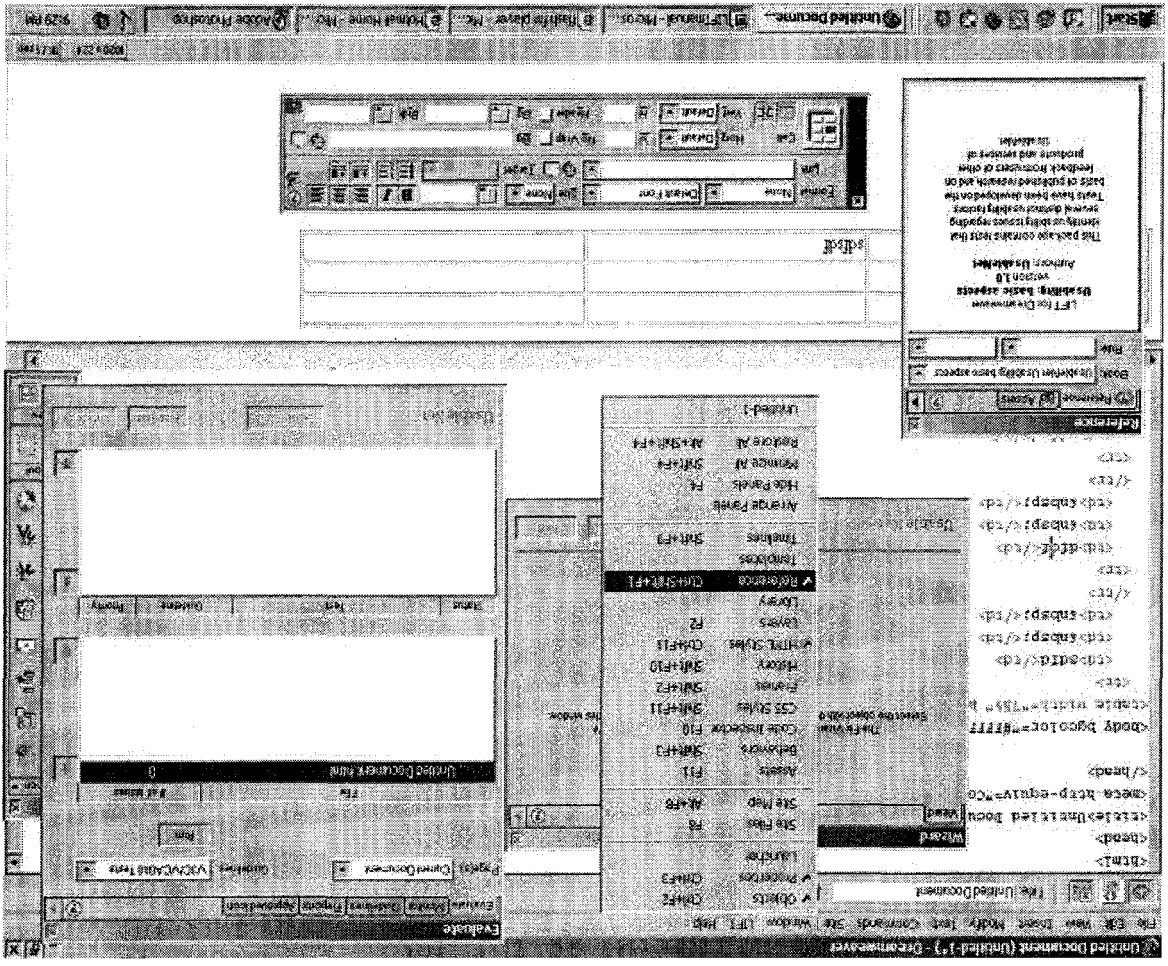


Figure 4.5: The LIFT error message templates showing manual issues

### 4.4.3 LIFT bugs

This section identifies some of the bugs that we encountered when trying to use LIFT and also illustrate some of the difficulties faced by the author as these issues are not discussed in the LIFT manual.

1. Sometimes the next button in the Wizard applet will not work yet the Wizard will have a new template loaded that has nothing to do with the task at hand. It will not take the author to the next issue displayed in the Evaluate Applet. In this case we just fixed one issue at a time, then saved the file and ran the Evaluate applet again. If the issue is repaired properly, next time there will be one less issue to deal within the Evaluate applet which also means that the Evaluate Applet becomes redundant.
2. When using the wizard applet to modify the ALT tags in a HTML page, the author needs to switch between two windows. For example, if we open the browser and the Dreamweaver 4 windows in order to view the picture and describe it for the ALT tag, when switching between windows information was lost.
3. When using the Wizard applet for image LONGDESC, the author selects the error, the applet will sometimes find the code. This only happens if the author performs the change and saves the work for each issue individually.

## 4.5 User Study

Typically, students access the internet to read course descriptions, register for classes, pay tuition, purchase books, submit assignments, take quizzes and check grades, among other tasks that are necessary for adequate delivery of a course.

### 4.5.1 Participant selection

Users with visual impairment were selected as this group represents a segment of the population that can most benefit from the internet and the virtual classroom technology. They also constitute a very challenging group as end users due to the difficulties in representing the visual information in a non-visual form.

In order to recruit users representative of the segment of the population, the Disability Office at the University of Victoria was contacted and they used their expertise in dealing with students with the impairment to identify suitable candidates. The Disability Office was requested to search for 3 to 6 users who fit the following profile: they had to be University of Victoria students, they had to be knowledgeable about computer applications and they must have used the web as a search tool. Furthermore, the subjects had to be knowledgeable in using their respective adaptive devices with some experience.

The users were required to sign a participant consent form approved by the UVic Human Research Ethics Committee Project No. 361-01. The participant consent form is shown in Appendix D. For this study, it was possible to recruit only three participants with visual disability and a fully sighted user was recruited for data control purposes. Fortunately, the disabled users recruited represent a broad spectrum in disability: from partially impaired to totally blind. The three subjects are all UVic students. The names presented are fictitious in order to protect the identity of the participants.

Janice is a Humanities student who has been totally blind since birth. She uses mostly the Braille device with added help and enhancements from the JAWS reader. She likes to switch back and forth between the two. She also mentioned that when she is thinking or visualizing a concept she switches to Braille only, but for navigational

purposes she uses JAWS as this provides audio direction.

James is a senior Science student who has been visually impaired (partial vision) since birth. He uses screen magnifiers to enlarge the screen and he also can work without a magnifier on a 21" screen with his corrective lenses.

John is a junior Arts student, who had a stroke 10 years ago that damaged the optical nerves and he became blind. After struggling with blindness, he learned to use JAWS and he is back at the school to get retrained.

In Summary, the users all had extensive knowledge on their adaptive devices, and they use computers on a daily basis for research or doing their work, checking their email, and chatting online, and reading news.

In this qualitative study, users in all three groups were asked to perform the same tasks. Data collected in this study is summarized for specific concerns stated by the disabled participants, providing first hand experience from which to base recommendations to be made to educators and tool developers.

#### **4.5.2 Design of questionnaire**

The Distance Education Survey (DES) [56], a standardized 88-item questionnaire developed in 1997 through the EvNet project at McMaster University was adapted to the Web to gather participant personal information. The DES questions participants about their experience with distance education, learning preferences, access to computers, and personal attitudes towards computers, thus providing additional qualitative data on participants attitudes and beliefs.

For the user study, questionnaires were designed to make sure that the users were experts in their area of adaptive technology because the study needed to focus on the web learning tool itself. The issues related to access should not be attributed to

the inexperience of the user. The hardware and software requirements as defined by the users were setup and a valid Netlink ID was created. The course was moved to the UVic's server. At the time of testing, the UVic server was running WebCT 3.8. Furthermore, a consent form was distributed to the participants to make sure that they were informed of the testing procedure and the requirements. Next, permission was obtained to video-tape the screen and audio tape the session.

An experimental handbook was designed, and it is included in Appendix C. The handbook includes a pre-questionnaire, the warm-up session and a task list. The pre-questionnaire was designed to obtain the background information on the work habits and the level of comfort with the computers. This activity was followed by an orientation, and this preceded a warm-up session to acquaint the users with the setup. For the testing phase, the types of tasks presented to the participants were designed to reflect a true virtual classroom environment. The questions were intended to identify the accessibility barriers of the learning tool. At the end of the testing phase, a post-study questionnaire was distributed and the users were interviewed.

### **4.5.3 Tasks**

Thus, a typical session started by requesting the participant to start WebCT and login to the main web page of the course. This was followed by asking the participants to collect basic information about the course such as the name of the instructor, the timetable and location of lectures, and the textbook for the course. The participants were next asked to view the lecture notes linked to the home page, find out the date for delivery of the first assignment and complete an online quiz. Finally, the participants were requested to logout of WebCT.

The following questions and tasks were designed with the objective of determining

## CHAPTER 4. A CASE STUDY FOR THE VISUALLY IMPAIRED USING WEBCT68

the accessibility of critical information in the course:

1. Login to the MyWebCT ( this is the main page of users' web page for their courses.)
2. Find out who is the instructor for the course.
3. Find out the days and the times of the lectures.
4. Find the location of the lectures (building and lecture rooms
5. Find the text book for the course.
6. Find the final exam date for the course.
7. Find the deadline for the first assignment.
8. View the first set of lecture notes linked from the home page
9. In this set of notes, find out what are the reading assignments?
10. Review the first lecture notes and find out what are the main topics of the course.
11. After finding the main topics in the first lecture notes, use the note taking tool (My Notes in the top frame) to enter a comment about your experience.
12. Send an email to the instructor about setting up a meeting for the upcoming project.
13. Submit a file, any small file will do, using the "Assignment Dropbox", which is linked from the Tools page.
14. Go to Student Area, and get familiarize with the workspace. Your workspace is linked from the Tools page.

15. View the class list and see who else is in the course.
16. Complete the online quiz, linked from the Home page or Tools page.
17. Logout of My WebCT.

#### 4.5.4 Results

This section provides a narrative on the participants activities and relevant observations have been noted. Activities during tool setup were annotated. James (Zoomtext user) did not need to modify any of the settings on the browser (IE 5.0). We set up ZoomText 7.1, set the magnification setting of 2 (10 is maximum) as he was also using corrective glasses. John and Janice had no difficulty setting JAWS and Janice used Braille as well. There were several issues of compatibility between the versions of Windows, and adaptive tools software and hardware. These were identified and the optimal combination was established for each user. It is noted that a fully sighted user was used to establish a datum for our observations. The fully sighted user typically completed all tasks from seconds to a couple of minutes.

During the warm-up session, we tested the users' comfort level with the browser and see how the user performs various tasks on the internet, the level of comfort with navigating the internet. The users were requested to complete task of searching CNN for Florida's average weather temperature in August. All users were able to complete the first task with the available adaptive tools.

The testing was evaluated in terms of the time taken to accomplish a task by each participant and a ranking number was attributed according to following rationale [2]:

1. Was unable to access. Required observer to intervene physically (inaccessible)

## CHAPTER 4. A CASE STUDY FOR THE VISUALLY IMPAIRED USING WEBCT70

2. Was able to access with instruction, but with no physical help (instructional, how to)
3. Was able to access with a hint (eg. Try tabbing, using the down arrow)
4. Was able to access but asked for clarification/affirmation (eg. Am I in a text area?)
5. Was able to access without instruction or physical help from the observer (accessible)

The results are tabulated in Table 4.1.

When the testing phase began, the participants were asked to login into WebCT. None of the users had any difficulty with this task and they all finished the tasks ranging from 50 seconds for the James (partial blindness) to three minutes for John (total blindness following an accident). Both James and Janice used JAWS to move between frames and this was not a simple task as both struggled to navigate on the main page. The task of finding of the name of the instructor for the course was more successful and they all commented that they liked the fact that this information was on the main page and little or no navigation was required. This task required typically 1 minute for the Zoomtext user to 2 minutes for the JAWS and Braille users.

As the complexity of the tasks increased, such as finding the textbook for the course and the date for the final exam, the participants using JAWS resorted to the 'find' and 'page heading' feature to navigate through the pages rather than listening to the whole text being read. James, The Zoomtext user, became increasingly frustrated as he was not able to keep a visual perspective of the page. This task took him 4 minutes. Next, the participants were asked to find the first set of lecture notes. James needed 3 minutes to locate the notes, and he had no difficulty opening the pdf file.

CHAPTER 4. A CASE STUDY FOR THE VISUALLY IMPAIRED USING WEBCT71

Table 4.1: Time taken and ranking attributed to the participants for task completion

No.	Task Description	Time(minutes)				Rank			
		PB	TBA	TBB	FS	PB	TBA	TBB	FS
1	Login to myWebCT	1	3	3	< 1	5	3	3	5
2	Find out the instructor for the course	1	2	1	< 1	5	5	5	5
3	Find out the days and the times of the lectures	1	2	1	< 1	5	5	5	5
	Find the physical location of the lectures	1	2	1	< 1	5	5	5	5
	Find the textbook for the course	4	1	1	< 1	5	5	5	5
	Find the date for the final exam	1	1	1	< 1	5	5	5	5
	Find the deadline for the first assignment	1	3	5	< 1	5	4	4	5
8	View the first set of lecture notes linked from the home page	3	6	4	< 1	4	2	2	5
9	In this set of notes, find the reading assignments	1	3	2	< 1	5	4	4	5
10	Review the first lecture notes and find out the main topics of the course	1	7	5	< 1	5	3	3	5
11	After you have completed reading the topics and found the information, use the note taking tool (My Notes in the top frame) to enter a comment about your experience	3	5	3	< 1	4	3	3	5
12	Send an email to the instructor about setting a meeting for an upcoming project	2	7	5	< 1	4	3	3	5
13	Submit a small file using the "Assignment Box" which is linked from the Tools Page	2	5	4	< 1	4	2	4	5
14	Go the "Student Area" and familiarize yourself with its structures. Your workspace is linked from the Tools page	2	6	7	< 1	5	4	4	5
15	View the class list while learning about your workspace and see who else is in the course	1	7	nc	< 1	5	3	1	5
16	Complete the online quiz, linked from the home page or Tools page	nc	nc	nc	< 1	1	1	1	5
17	Logout of MyWebCT	2	1	1	< 1	5	5	5	5
<b>Key</b>									
nc = task not completed									
PB = Partially blind user									
TBA = Totally blind due to accident user									
TBB = Totally blind at birth user									
FS = Fully sighted user									
Rank 1-5 , 1- being unable to access and 5- accessed without help									

#### *CHAPTER 4. A CASE STUDY FOR THE VISUALLY IMPAIRED USING WEBCT72*

Janice and John also took 4 minutes to locate the lecture notes. When asked to locate the reading assignments in the lecture notes, James used the search function within the page to complete the task within 1 minute. Janice had a slightly better time in locating the assignment as she used a combination of Jaws and Braille to locate the assignment within the page. Note that the users were informed that the assignment was on that particular page, as the time required would have considerably increased if they had to find out by themselves.

The tasks of using the 'My Notes' capability to enter comments was accomplished without much difficulty by all participants once they identified the icon. The same level of difficulty was encountered when they were asked to send an email to the instructor. Also, the participants were not able to submit a file to the instructor as they had difficulty locating the 'Assignment Drop' box on the page. The common barrier faced by all users was in locating the icons on the page and the question "Am I in the right place" was a common issue with all users, especially with the totally blind.

When trying to complete the online quiz, JAWS had difficulty with the tables. This task was aborted as all the users were becoming increasingly frustrated with tabular text. Finally, the users were requested to log off and again finding the icon was a barrier.

Inspecting the table more closely, it can be inferred that the users were successful completing the easy tasks such as finding the instructor, the day, time and location of the lectures, and the textbook for the course. In tasks 6 and 7 the users had to find the final exam date and the deadline for the first assignment. Here, they were able to complete the tasks successfully by executing search utilities of the browser and JAWS. However, in our opinion, the success could be attributed to the good design of the page where all the elements were located centrally and the users expressed the

view that they liked that all the critical information was located in the same page. If navigation principles had to be applied, the outcome could have been different.

As the tasks became more complex, a different pattern emerged from the tabulated data. It can be observed that the users required more time to complete the tasks and the ranking clearly shows that the users needed guidance and help to navigate the pages (Tasks 9 - 14). Finally, for tasks 15 and 16, that required finding their workspace and the class list, and completing the online quiz, the totally blind users were not able to finish the tasks in finite time. It is noted that the partially blind user was able to complete task 15 successfully but required considerably more time to finish task 16. Note that the fully sighted user completed all tasks in less than one minute and a ranking of five was attributed in all tasks.

Summarizing, the participants voiced their frustrations and concerns when navigating through the pages and frames, however they found WebCT useful as a tool for exchange of information. One of the issues with screen magnifier was the loss of context and the cursor would jump around the screen. From the course layout point of view, they mentioned that they felt more in control when all the information was on a single page, rather than having hyperlinks or having to sort through many pages. They also found the naming of the frames confusing and not very informative. They often required a sighted user to reset the pages so that they could get back on track. The use of tables created many accessibility problems as the adaptive tools were not able to read them properly. In the warm-up session, popup windows were definitely a barrier to accessibility as JAWS often lost its bearing and often confused the user. The users also expressed the lack of human interaction and social aspect of attending classes.

### **4.5.5 Observations**

Many difficulties associated with the learning tools were identified for the users with visual impairment. Qualitative data was collected based on observations during the study for specific concerns expressed by the disabled participants.

Our goal was to identify issues around web-based learning tools. Three categories of issues arose: (i) issues related to the operating systems OS and adaptive tools from the administrator's perspective; (ii) issues related to publishing from the authoring perspective; and (iii) issues related to learning from the user perspective.

#### **Issues related to administrators**

The version of JAWS 3.3, which was provided to students by UVic did work with Windows 95 and 98, at the time of testing in July of 2002. However, JAWS was not compatible with web applications. Therefore, we used the users licence key for JAWS 4.0 which worked with Windows NT, as JAWS licensing requirements are very specific to the operating system. This barrier limited the testing considerably as it was not possible to change workstations during the testing. The more recent version 5.0, released in October 2003 is compatible with Windows 2000 and XP.

#### **Issues related to authors**

We opted out of pop-up windows for this version of WebCT as a previous study, conducted by Wang and Storey, suggested that pop-ups created a barrier to users. Note that the effect of the pop-up windows was evident during the warm-up session as the users lost focus and were frustrated by them. WebCT uses frames which really confuses the students with JAWS and Braille. However, the users could circumvent

this issue with JAWS (CTRL+ feature) which allows navigation between frames to find which frames they should go into.

Another problem is that WebCT does not identify frames very well, so navigation becomes an issue. The WebCT frame titles were not intuitive to the users and this made the tagging of the frames irrelevant. The course front page was setup in a 3 by 3 table to place the icons. Although the icons in the course were labelled, they did not help the Braille user during navigation.

### **Issues related to end-users**

The fully sighted participant not using assistive devices was able to complete all the tasks within the time limits and had no issues related to accessibility. The users who used Jaws and Braille took on average double the time to navigate and find answers to questions than the partially impaired users using screen magnification tools. All users experienced difficulties in navigation and some of the tasks were not completed and frustration was evident during the testing phase.

WebCT uses frames and ZOOMTEXT needs to refresh the frames, so when James tried to scroll vertically and horizontally, the screen refresher created a jitter and James constantly lost his position on the screen and this was obviously a source of frustration to the user.

When uploading documents in WIN 98, PDF documents in ZOOMTEXT 6.0 would not load. It was necessary to quit ZOOMTEXT 6.0 and subsequently load the Acrobat Reader and again initialize ZOOMTEXT and re-load the PDF document. However, ZOOMTEXT 7.0 in the WIN NT environment did not exhibit this problem. It was interesting to observe that for blind users the keyboard and speakers provide the same functionality as the mouse and the monitor provide for the fully sighted

users. Also, braille users have an added advantage as they can turn off the reader and visualize the documents with written words. However, it is very difficult to train recent blind users with brail as it is a language by itself.

## 4.6 Study limitations

Although this study has provided useful insights on the issues related to the delivery of web based courses, it is important to note the small number of users that participated in this study as it was very difficult to recruit participants. The study has used the most recent technology and this created problems of compatibility between the various components of the testing environment. This observation emphasizes the need for implementing the accessibility guidelines in the developmental stage of new technology and for the continued evaluation of these technologies as they are improved and updated with new features.

## 4.7 Recommendations

The main focus of this study was on the author's role in aiding accessibility. The observations and findings substantiate the conclusion that the authoring and learning tools for web-based education need to be improved in order to accommodate people using assistive devices.

To summarize the findings from this study, the following are some of the recommendations proposed for web authors to enhance accessibility for people with visual impairment:

### 4.7.1 Navigation Bar

The navigation bar is the first item in a web page. For users with visual impairment using a screen reader, the same series of links are repeated each time a page is accessed. By containing the links in a logical grouping, the links can be addressed as a unit rather than individually. This can be accomplished by placing a *skipover* link before a grouping. This enables the screen reader to bypass a set of links and start accessing the page where the main content begins.

### 4.7.2 Loading of Pages

It was observed that the users had difficulty realizing when the pages had completed loading. The Zoomtext user did not often see the complete page. The JAWS user thought a page had loaded, but the screen reader never read the page. In several instances, screen readers did not read pages at all, and there was no indication to users that anything was even on the screen.

### 4.7.3 Title Page

It is useful to include a page title for users with screen readers and magnifiers, as the users found it reassuring to hear the page title. It confirmed to the users that they had reached the intended point during navigation.

### 4.7.4 Labels

The screen reader user was often uncertain of the site reached. It is recommended that authors use both TITLE tags to name pages and ALT text to name icons, or

alternatively provide the name of the site at the top of the page. Naming links and graphics is not always enough, and repeating what the visible page shows does not always help.

#### **4.7.5 Images**

Images should be avoided whenever possible. Unfortunately, in this a case a picture is not worth a thousand words. Text is accessible to all users (including those relying on screen readers, braille displays, and non-graphical browsers) and it requires far less download time than a graphic. When writing ALT text, keep the text concise and simple. Use the ALT attribute after the image name to briefly describe the image. Use the LONGDESC attribute to thoroughly describe the image.

#### **4.7.6 Pop-up Windows**

Pop-up windows are disorienting and often frustrating to people using screen readers and JAWS. For people using screen readers, it can be a big problem if they click something, expect a certain page, and then hear something completely different like the contents of an unexpected pop-up window.

During the warm-up session, when accessing the CNN site, at several instances a new browser window opened and covered the window behind it. When users tried clicking BACK it did not seem to work, and they usually ended up closing the browser altogether to recover. At times, people using screen readers did not realize that several browser windows were open at once.

### 4.7.7 Scrolling

Users with low vision may not want to scroll through an entire dialog box. For example, with security alert boxes, some users just press the Enter key without reading the active message. The authors should design web pages that minimize the need for scrolling. Users with screen magnifier already need to scroll many times to see the entire contents of even a small page. When people use the scroll bar to scroll down a page, they must leave wherever they are and scroll to the right hand side, then use the scroll bar to move down the page. This is time-consuming and can be tedious and distracting.

### 4.7.8 Page Layout

For people with partial vision, space between buttons and links is extremely important. Aiming for a specific target can be difficult, and it is troubling and frustrating for people when they accidentally miss the target, or worse, hit another target that is very close. Sighted users also sometimes accidentally hit the wrong links when links appear too close together. Leaving blank space between links and buttons helps alleviate this problem.

Also, when screen reader devices read pages, they usually do not pause in the right places, so unless the page designer adds it, there is no indication for pauses. It is the author's responsibility to separate all commands and links so that people using screen readers can hear important commands in a page.

### 4.7.9 Tables

All tables should include a valid caption element. Web content authors should use CAPTIONS to indicate the purpose of a TABLE within a document. The CAPTION element is important because it attaches an identifiable label to a TABLE and assists adaptive devices in extracting specific information from the TABLE by creating automatic numbering schemes and separating the data contained in cells into individual files.

## 4.8 Synopsis

In this chapter, an integrated environment was setup for a user study to identify the accessibility barriers encountered by users with visual impairment.

Four test subjects representing a spectrum of visual disabilities (two totally blind, one partially blind and one fully sighted) were selected to test the online course content for accessibility. From the author's point of view, in order to make the course content accessible to users with visual disability, the course content was verified and repaired. From the user's perspective, each participant using his/her adaptive device of choice proceeded to evaluate the online course for accessibility and observations were recorded. This limited study provides a set of guidelines to authors of web-based information in order to make the web content more accessible to users with visual impairment.

Although the main thrust of the study was to observe authoring issues, we were able to collect data that helped us formulate a few recommendations for administrators and users as well. From a user's perspective, users with screen readers and magnifiers took longer to complete each task because users with screen readers had to

#### *CHAPTER 4. A CASE STUDY FOR THE VISUALLY IMPAIRED USING WEBCT81*

listen to items being read and users using screen magnifiers had to scroll many times just to see a single page with considerable jitter resulting in confusion and frustration.

This study has raised fundamental new issues. In the quest for universal access and design for all-inclusive virtual classrooms, are we inadvertently creating barriers to other user groups that may also need to have access to the same virtual information? The minimization or elimination of visual information affects the "flow" for fully sighted users. The compliance of Section 508 guidelines does not address this issue. How do we target different users groups while maintaining optimal experience to each and every group? This question takes us naturally to the next chapter, where we attempt to respond to these concerns by looking at two other software design characteristics that are important in the definition of accessibility: changeability and fault-tolerance.

## Chapter 5

### Other Accessibility Issues

In a virtual classroom, the author has to deal with different types of learners that require varying levels and types of accessibility. The findings and recommendations proposed in the previous chapter lead to the question of how to accommodate diversity in a virtual classroom while assuring optimal experience for all users. For example, when web content is designed to make it accessible to blind users, how does it impact fully sighted users?

When trying to reach the widest possible audience in a web medium, and at the same time trying to avoid imposing a single uniform reality to all users [57], we have to design software tools with maintainability (changeability) and reliability (fault-tolerance) in mind. In our definition of accessibility proposed in Chapter 1, these are necessary considerations when designing accessible web based authoring and learning tools. *Changeability* relates to the attributes of the software that bear on the effort needed for modification or for environmental change, and *fault-tolerance* relates to the attributes of software that bear on its ability to maintain a specified level of performance in case of software faults or of infringement of its specified interface. In this chapter, these two issues, illustrated in Figure 5.1 are reflected upon to generate

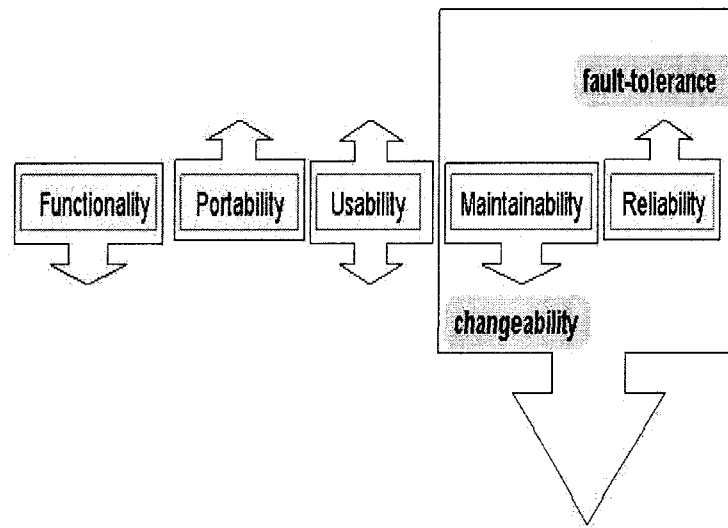


Figure 5.1: Accessibility barriers in web-based education

further discussion and provide direction for future research.

On the issue of changeability, a prototype has been designed to demonstrate the potential use of Intensional Sequential Evaluator (ISE) [58] and its role in customizing accessibility for all users groups by allowing the design of multi-versioned web sites.

Finally, while trying to accommodate all the different user groups, how can we ensure the reliability of the system for real-time and fault-tolerant delivery of web content. Quality of adaptation is one of the methods to improve reliability. Distribution of web content into more than one server by replicating or partitioning is another method to provide services to a large number of users. We refer to a distributed multimedia system that was designed for fault tolerance services using QoS (Quality of Service) adaptation during fault in servers. A simulation program was designed and implemented to illustrate the application of QoS to enhance fault-tolerance of multimedia server systems. We briefly mention this in Section 5.2.

## 5.1 Maintainability and Changeability

In theory, web pages can be updated frequently to provide customized versions on demand. In practice, however, rarely can authors provide multi-versioned web content to accommodate different levels of requirements by users. Also, the ability to maintain such sites is not feasible as HTML provides little support for the design and maintenance of multi-versioned sites.

For example, in a scenario where a multi-lingual web site needs to be developed. For each language, the web site will have roughly the same structure and graphics. However, the text contained within each version of the web site will have to be different to reflect the selected language. The conventional way to structure such a site is to create the English version, and subsequently translate the web content in a parallel version, one for each language. In this scenario, each version would be a complete parallel site, with reuse of the common structure of the web pages [59]. Authors of multi-version sites are required to create copies by duplicating, editing and maintaining many separate and parallel pages.

In view of this problem, Wadge [57] has proposed the Intensional Hyper Text Markup Language (IHTML) that extends the HTML language to allow authors to create web content for environments that require multi-versioned material. IHTML allows the author to specify parameters based on the version and switch statements where the action to be performed in the document would vary depending on the current version. The author can control the extent to which the user can alter the parameters by following links or by using menus and forms. The IHTML implementation uses ISE, a Perl-like CGI language with run time parametrization. The structure of the ISE application is illustrated in Figure 5.2. Here, each language represents a version in the multi-dimensional space. The default language may be English, but

once a different language, say French, is selected by user B, the language dimension assumes the value French, as illustrated in Figures 5.3 and 5.4. Subsequently, ISE packages the modules according to the requests submitted by each individual user, and the web server generates an HTML document to cater to each user requested profile.

With reference to the issue at hand, in a virtual classroom setting there would be a page containing the course outline for the fully sighted and another customized page would present the same information tailored to the visually impaired. The author can specify parameters in order to make available various modules of the same content but in different formats. In turn, the user can request the appropriate content of the course by specifying their user profile in the form of parameters. For example, one user can request content designed for sighted and English speaking people, while another user can request the same content designed for blind and French speaking people.

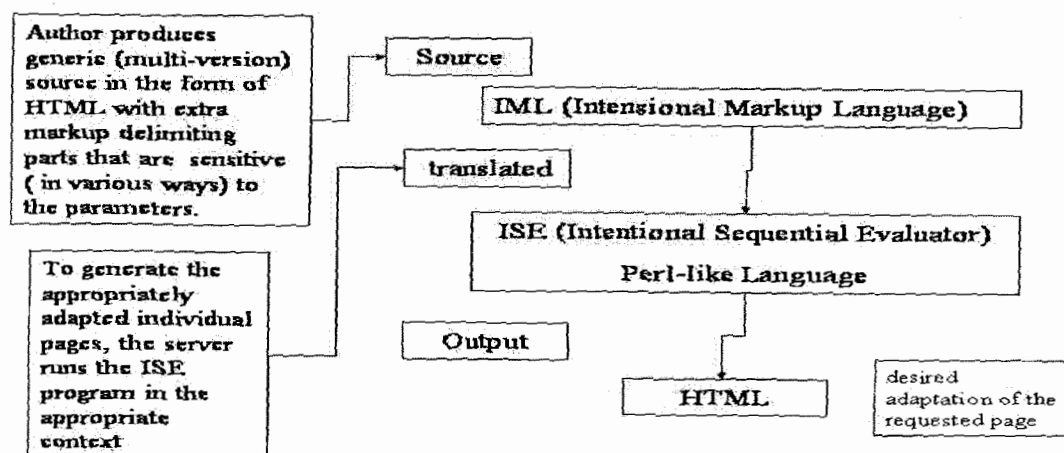


Figure 5.2: The ISE program structure

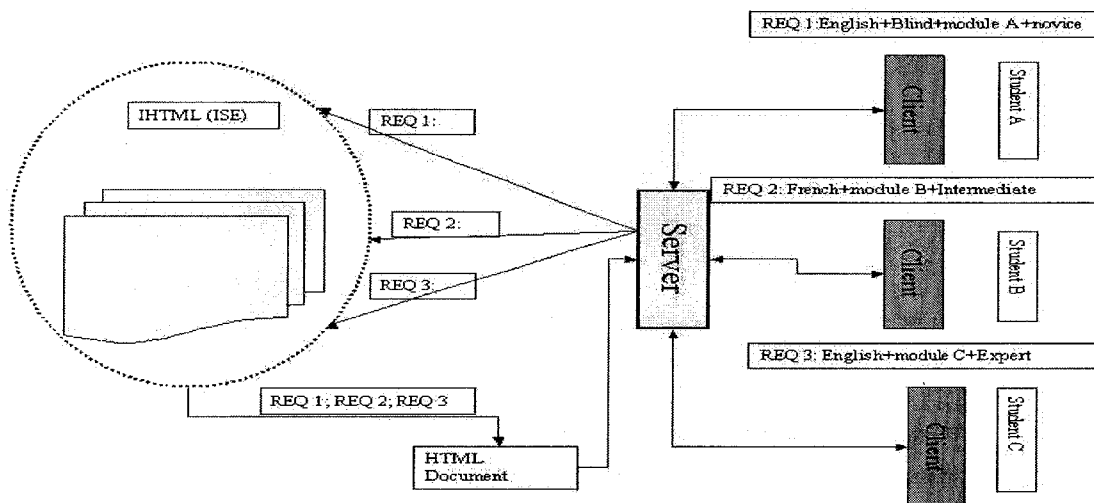


Figure 5.3: The IHTML multiversioning tool

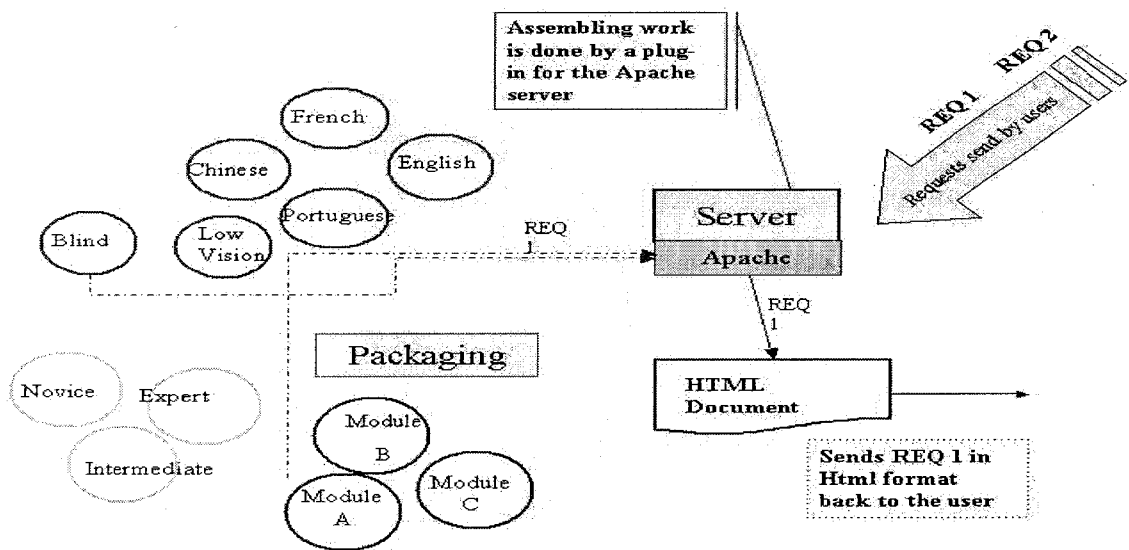


Figure 5.4: The IHTML setup

### 5.1.1 Example: A repository for assistive devices

In order to demonstrate that ISE is ideally suited for making web sites accessible to the widest possible audience in a web medium, we created a prototype for a repository of some available tools and technologies.

There has been a lot of work done in the area of assistive technology, but we felt that there is still a need to provide a repository of all the available tools and technology and to automate this process. Some assistive technologies are obvious, such as wheelchairs and closed-captioned television. Others may be overlooked at first by the general population, but are still familiar to many people, including braille, sign language, and ramps. We developed a search instrument to make it possible to automatically select the best combination of assistive technologies to help the person with a particular disability. In order to complete this prototype, an extensive search of all the assistive technologies in the open literature was performed.

With respect to the details of the prototype, we created two lists, one holding the information on the different types of disability, and the other one containing most of assistive technologies available in the open literature. A query would access these two lists. The output of the query consists of a customized list of assistive tools or products associated with that disability. Once the customized list is produced, the user can then choose the assistive technologies of interest.

The structure and layout of the web page is illustrated in Figures 5.5 and 5.6.

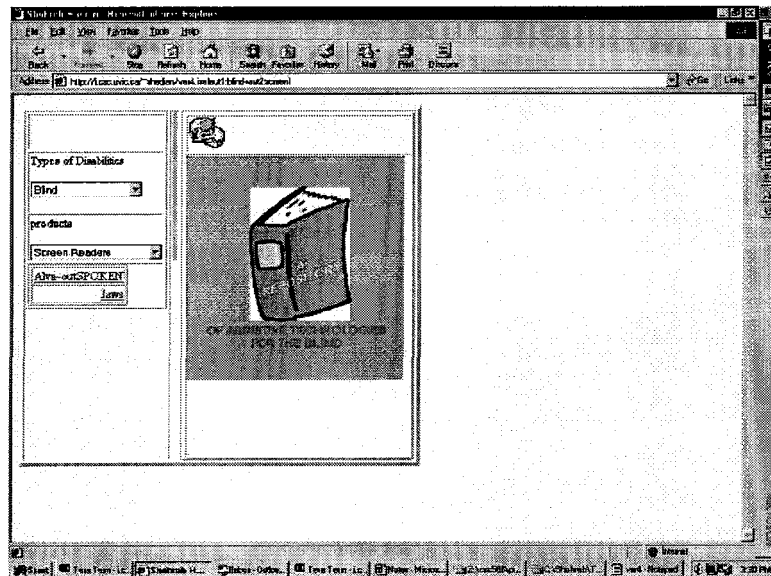


Figure 5.5: The search engine

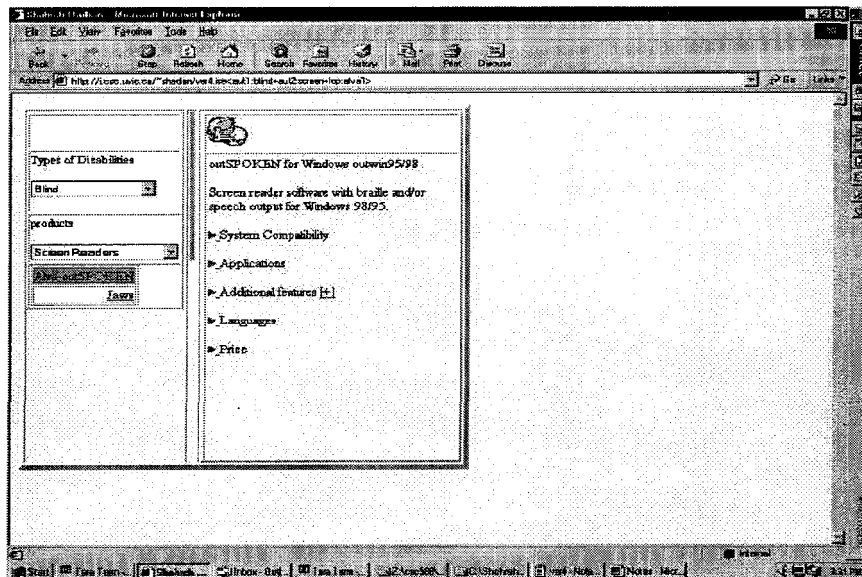


Figure 5.6: The internal structure of the search engine

## 5.2 Reliability and Fault-Tolerance

In this thesis, one last question remains to be probed. If the author is able to multi-version the web material to accommodate the diversity of the end-users, and if this service is implemented in a client-server environment using the web as a medium, how can we ensure that the server that contains all the information is reliable and fault-tolerant so that there is minimal disruption of service?

Fault-tolerance relates to the attributes of software that bear on its ability to maintain a specified level of performance in case of software or hardware faults. While computer users are accustomed to software failures and operating system crashes, users of interactive services are not likely to tolerate anything but rare unavailability of these services. Consequently, servers used for real-time delivery of educational material in a virtual classroom setting must employ techniques to guarantee high availability of service.

Quality of adaptation is one of the methods to improve reliability. Distribution of web content into more than one server by replicating or partitioning is another method to provide services to a large number of users. In other work, we did a study of a distributed multimedia system which was designed for fault tolerance services using QoS (Quality of Service) adaptation during fault in servers.

The QoS adaptation technique was studied to improve the fault tolerance and the capacity of the system to tolerate fault in multimedia systems. A distributed adaptive multimedia system architecture was proposed and fault tolerance parameters of the proposed system were implemented. The goal was to find the reliability of a system therefore the probabilistic nature of every element of the system was examined carefully. The frequency of session requests and the length of the sessions followed a particular probabilistic behavior. Markov models were applied to determine the

reliability of the system. The discrete event simulation was assumed appropriate to simulate the proposed system by generating random numbers.

This aspect of the study on the computational study of reliability and availability was carried out in the context of a graduate course project on Fault Tolerant Computing (CSC 554) at UVic in collaboration with M. Akbar (PhD student) and the results of this study were presented and published in the Proceedings of the International Conference on Computer and Information Technology (ICCIT2000) [60]. This paper is available at <http://www.csc.uvic.ca/shadian>.

### 5.3 Synopsis

This chapter reflects upon two issues that affect accessibility i.e. maintainability and reliability. Maintainability relates to the diversity in the type of users, and the delivery of user-targeted information. To this end, an application based on multi-versioning (customizing the delivered material) for different type of users was designed and implemented. The prototype that was developed demonstrated the application and usefulness of ISE in the design and fast delivery of multi-versioned web sites.

Reliability or fault-tolerance relates to the attributes of software to maintain a specified level of performance in case of software or hardware faults. We refer the reader to the paper outlining the details of a distributed multimedia system that was designed for fault tolerance services using quality of service adaptation during faults in servers.

The findings from these very preliminary studies are aimed to inform the reader on two issues that may influence accessibility and provide a basis for further development in the field.

## Chapter 6

# Conclusions and Recommendations

This thesis has identified barriers to accessibility encountered by users with visual impairment in web-based environments. The problem was approached in two fronts: the author's responsibilities to make their web material more accessible to assistive devices for the blind; and from a user's perspective, the barriers faced by the visually impaired were noted during a case study and recommendations have been noted for administrators, authors and users of teaching and learning tools. Initially, The study defines accessibility in the context of good software design practices. Here, for the first time, accessibility has been looked at from interoperability, adaptability, learnability, operability, changeability and fault tolerance perspectives. From the authoring point of view, two popular teaching and learning tools, *WebCT* and *Blackboard*, were presented and evaluated for accessibility compliance with Section 508 of the Disability Act. On the one hand, the findings substantiate that authors have a greater role in enhancing accessibility of their web material. On the other hand, the tool developers need to design integrated development environments to enable "optimal experience" for the authors in order for them to concentrate on the pedagogical issues of their course content. Ideally, all authoring tools in an integrated design environment would

provide utilities and information to prompt creation of accessible resources. In the absence of integrated tools designed for writing accessible HTML code, verification and repair tools are necessary to examine the accessibility of web content after it has been created. To this end, a few content verification and repair tools were presented. Secondly, a user study was designed and implemented to identify the barriers faced by the visually impaired. The study involved three participants covering a broad spectrum of disability from totally blind to partially blind. A fully sighted user was also selected for control purposes. To make the course material accessible to the blind, LIFT in combination with Dreamweaver was selected for the user study. Also, assistive devices were identified for use by people with visual impairment and three devices were used during the testing phase: JAWS, Zoomtext and Braille. The study enabled us to learn how people with impaired vision use assistive devices and the web. We have provided some recommendations so that developers and authors better understand how people use assistive devices, and how design features can enhance or impede accessibility. Blind users have unique obstacles when accessing web information. It is important that authors are clear and logical and appropriate navigational aids are incorporated for people using screen readers. The use of multiple frames and nested tables should be discouraged. Frames that are either unlabelled or improperly labelled present complications for blind users as they are trying to form a mental picture of the information. A link to a site map page that explains the relationships between frames and navigation structures should be incorporated to enable the blind user the learning structure. People with partial vision have similar concerns, however they are attenuated by the residual vision. This study has raised fundamental new issues. In the quest for universal access and design for all-inclusive virtual classrooms, are we inadvertently creating barriers to other user groups that may also need to have access to the same virtual information? The minimization or elimination of visual information affects the "flow" for fully sighted users. The compliance of Section 508

guidelines does not address this issue. How do we target different user groups while maintaining optimal experience to each and every group? This question forced us to reflect upon two other software design characteristics that were included in our definition of accessibility: changeability and fault-tolerance.

## 6.1 Research Contributions

The original contributions of this thesis are summarized as follows:

1. A clear and concise definition of accessibility is introduced based on good software design characteristics.
2. Design, development and testing of a case study to identify accessibility barriers encountered by users with visual impairment was implemented. The findings and recommendations based on the observations from the user study concur with the W3C guidelines in WAI and in a way further emphasize the importance of including the design guidelines into web based information to enhance universal access. Furthermore, the WAI guidelines are technical and a course developer may not necessarily have the knowledge to interpret the technical information codes. The current findings explain with examples and simple language the implications of non-compliance with the guidelines. Issues related to administrators, authors and users are delineated.
3. This study has been approached from the Computer Science perspective. One fundamental conclusion that resulted from this study is that currently accessibility is provided by gluing together discrete components such as the authoring tools, verification and repair tools, browsers and assistive devices. Instead of

approaching the problem from the user end, it is recommended that the solution be found from the development and authoring perspective, as well.

4. The application of the multi-versioning tool ISE to enhance accessibility has also been introduced as a feasible solution to address the issue of universal access for all.

## 6.2 Future Work

Most of the published literature to date suggests to solve the issue of accessibility from the user's side. However, the conclusions from the current study substantiate that the issue of accessibility as a critical component of usability has to be a collaborative effort between developers, authors and users. This may be possible by creating an Integrated Design Environment (IDE), so that teachers and authors can use an authoring tool to create the code, parse and verify it in a seamless and integrated way, and finally upload the material while fully concentrating on the pedagogical issues at hand. Ultimately this approach will result in a better environment for all end-users. To ensure inclusiveness for all, it is proposed to integrate ISE with Amaya to create and develop a course content for users with diverse and multi-level needs. The W3C has recently released Amaya [61]. Amaya is a Web editor. The main motivation for developing Amaya was to provide a framework that can integrate as many W3C technologies as possible. Finally, both WebCT and Blackboard are based on a client-server architecture. Reliability of the server is an issue for access, specially when dealing with multimedia components for real time delivery. In order to ensure real-time delivery of a virtual classroom, distributed service into different servers using QoS adaptation will improve the reliability of the system, thus enhancing accessibility.

## Bibliography

- [1] B. Landon and R. Robson. Technical issues in systems for www-based course support. International Journal of Educational Telecommunications, 5(4), 1999.
- [2] L. Harrison. Inclusion in an electronic classroom - 2000: the role of the courseware authoring tool developer. Adaptive Technology Resource Centre, University of Toronto, [http://snow.utoronto.ca/initiatives/access\\_study/ATrec.html](http://snow.utoronto.ca/initiatives/access_study/ATrec.html)(retrieved in July 2002), 2000.
- [3] N. Nielsen. Beyond alt text: Making the web easy to use for users with disabilities. Nielsen Norman Group, <http://www.NNgroup.com/reports/accessibility>, 2001.
- [4] B. Bederson. Interfaces for staying in the flow. Technical Report HCIL-2003-37, University of Maryland, October 2003.
- [5] M. Maczewski M-A. D. Storey, B. Phillips and M. Wang. Evaluating the usability of web-based learning tools. Educational Technology and Society, 5(3), 2002.
- [6] M. Hricko. Design and implementation of web-enabled teaching tools. Information Science Publishing, 2003.
- [7] A. Foley. Distance, disability and the commodification of education: Web accessibility and the construction of knowl-

- edge. Journal of Current Issues in Comparative Education, 6(1), <http://www.tc.columbia.edu/cice/articles/af161.htm> 2003.
- [8] S. Burgstahler. Distance learning: Universal design, universal access. Educational Technology Review, International Forum on Educational Technology Issues and Applications, 10(1), 2002.
- [9] R. D. Owston. The world wide web: A technology to enhance teaching and learning? Educational Researcher, 26(2):27-33, 1997.
- [10] ESSI-SCOPE. Information software product quality. <http://www.cse.dcu.ie/essiscope/>, 2000.
- [11] E. Van Veenendaal and J. McMullan. Achieving software product quality: Essiscope. Den Bosch, <http://www.cse.dcu.ie/essiscope/>, 1997.
- [12] National Educational Association of Disabled Student. <http://www.neads.ca>, 2000.
- [13] World Wide Web Consortium. <http://www.w3.org>, 2000.
- [14] Web Accessibility Initiative. <http://www.w3.org/WAI/>, 2000.
- [15] H. Yu. Web accessibility and the law: Issues in implementation. Design and Implementation of Web-Enabled Teaching Tools, <http://www.section508.gov/>, 2002.
- [16] Common Look and Feel for the Internet. Government of Canada, [http://www.cio-dpi.gc.ca/clf\\_nsi/index\\_e.asp](http://www.cio-dpi.gc.ca/clf_nsi/index_e.asp), 2001.
- [17] S. Burgstahler. Disabilities, opportunities, internetworking and technology. <http://weber.u.washington.edu/~doit>, 2000.

- [18] Equal Access to Software EASI and Information. Rochester New York, <http://www.isc.rit/edu/~easi/>, 2000.
- [19] SNOW Special Needs Opportunity Windows. <http://snow.utoronto.ca/>, 2000.
- [20] The Node learning Technologies Network. <http://Thenode.org/>, 2001.
- [21] M. Wang. Evaluating the usability of web-based learning tools. MSc Thesis, Department of Computer Science, University of Victoria, B.C., Canada, 2000.
- [22] P. Pavlik. Collaboration, sharing and society teaching, learning and technical considerations from an analysis of webct, bscw and blackboard. <http://members.fortunecity.com/pgp5/Collaboration.Learning.and.Society.htm>, 2000.
- [23] M. Wang M-A. D. Storey, J. Bavelas and B. Phillips. The importance of usability testing for web-based learning tools. Educational Technology Conference Series, University of Victoria, Victoria, B.C., Canada, May 4-5, 2002.
- [24] M. Lynch and P. DeWitz. Web-based teaching and learning for blind or visually-impaired faculty. Design and Implementation of Web-Enabled Teaching Tools, 7:124-143, 2002.
- [25] Statistics on Blindness and Blinding Diseases in the United States. The university of washington, department of ophthalmology. <http://depts.washington.edu/opthweb/statistics.html>, 2004.
- [26] J. Treviranus. Alternative access to the world wide web. Proceedings of the 10th Annual Conference on Technology and Persons with Disabilities, Los Angeles, CA, 1995.

- [27] UsableNet. Lift. [http://www.usablenet.com/products\\_services/lift\\_dw/lift\\_dw.html](http://www.usablenet.com/products_services/lift_dw/lift_dw.html), 2002.
- [28] A. Guercio P. Maresca T. Arndt, S.K. Chang. An xml-based approach to multimedia software engineering for distance learning. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, Ischia, Italy:525-532, 2002.
- [29] M. Pearrow. Web accessibility handbook. Rockland, MA, Charles River Media Inc, 2000.
- [30] Learning Without Limits WebCT. <http://www.webct.com/>, 2000.
- [31] Blackboard. <http://company.blackboard.com>, 2000.
- [32] Trivantis. Lectora publisher overview. <http://www.trivantis.com>, 2004.
- [33] Emedia Design. Virtual-u. <http://www.emediadesign.com/sfu>, 2000.
- [34] IBM Lotus. Lotus learningspace. <http://www.lotus.com/lotus/offering3.nsf/wdocs/learningspacehome>, 2003.
- [35] Prometheus; the Evolution of Learning. George washinton university. <http://www.prometheus.com/product/>, 2003.
- [36] Darasoft. <http://www.darasoft.net>, 2003.
- [37] Manhattan Virtual Classroom. Western new england college. 2002.
- [38] Intralearn Software. Software for e-learning. <http://www.intralearn.com>, 2004.
- [39] Macromedia. Authorware 7: Courseware for e-learning. <http://www.macromedia.com/>, 2004.

- [40] Digital Workshop. Opus pro 04: Multimedia crativity made easy. <http://www.digitalworkshop.com/index.shtml>, 2004.
- [41] The Kallidus Product Family. Kallidus authoring system. <http://www.e2train.com/>, 2004.
- [42] B. Schneiderman. Designing the user interface, strategies for effective human-computer interaction. 3rd Edition, Addison Wesley, 1998.
- [43] Simth S.L. and Mosier J.N. Guidelines for designing user interface software. Report ESD-TR-86-278, MITRE, Bedford, MA:386-388, 1986.
- [44] Watchfire Products and Services. Bobby. <http://www.cast.org/bobby/>, 1999.
- [45] University of Toronto. A-prompt: Web accessibility verifier. <http://aprompt.snow.utoronto.ca/>, 2000.
- [46] HiSoftware. Accverify/accrepair ds2. <http://www.hisoftware.com/access/>, 2004.
- [47] SSB Technologies. Infocus accessibility software. <http://www.ssbtechnologies.com/>, 2004.
- [48] Macromedia. Dreamweaver mx 2004. <http://www.macromedia.com>, 2004.
- [49] L. Harrison and L. O'Grady. Web accessibility validation and repair: Which tool and why? <http://snow.utoronto.ca/access/evaltoolreview/validation.html>, 2003.
- [50] JAWS for Windows. Freedom scientific. <http://www.freedomscientific.com>, 2003.
- [51] MAGic. Freedom scientific. <http://www.freedomscientific.com>, 2003.
- [52] Ai Squared Products. Zoomtext 8.1 magnifier. <http://www.aisquared.com>, 2004.

- [53] IRTI Products Innovative Rehabilitation Teechnology Inc. Braille voyager. <http://www.irti.net>, 2004.
- [54] Logitech Products. ifeel mouseman. <http://www.logitech.com>, 2004.
- [55] Scansoft Productivity Without Boundaries. Dragon naturallyspeaking. <http://www.scansoft.com>, 2004.
- [56] McMaster University Evnet Group. Distance education survey (des). <http://socserv2.mcmaster.ca/srnet/evnet.htm>, 2000.
- [57] W. Wadge. Intensional markup language. Lecture Notes in Computer Science; also Proceedings of the Distributed Communities on the Web: Third International Workshop, 2000, Quebec City, Canada, 1830/2000, 2003.
- [58] P. Swoboda and W. Wadge. Vmake and ise general tools for the intensionalization of software systems. In M. Gergatsoulis and P. rondogiannis, editors, Intensional Programming II, pages 310–320, 2000.
- [59] M. Schraefel W. Wadge, G. Brown and T. Yildirim. Intensional html. Proceedings of the 4th International Workshop on Principles of Digital Document Processing; also Lecture Notes in Computer Science, 1481:128–139, 1998.
- [60] S. Hadian M. M. Akbar and M. Kaykobad. Fault tolerance of distributed multimedia system using quality adaptation. Proceedings of International Conference on Computer and Information Technology, Dhaka, Bangladesh:55–59, 25-26 January 2001.
- [61] W3C's Editor/Browser. World wide web consortium. <http://www.w3.org/Amaya/>, 2003.

## **Appendix A**

# **Correlation between WCAG and Section 508**

Table A.1: The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act

Correlation between the Web Content Accessibility Guidelines and the section 508 of 1194.22 legislation set by government.			
Criteria		W3C	Section 508
<b>Guideline 1: provide equivalent alternatives to auditory and visual content.</b>			
1.1	A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).	P1	A
1.2	Redundant text links shall be provided for each active region of a server-side image map.	P1	A
1.3	Provide an auditory description of the important information of the visual track of a multimedia presentation.	P1	E
1.4	Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.	P1	B
1.5	Provide redundant text links for each active region of a client-side image map.	P3	
<b>Guideline 2: Don't rely on color alone: Ensure that text and graphics are understandable when viewed without color.</b>			
2.1	Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.	P1	C
2.2	Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. P2 (images) P3 (text)	P2/P3	

Table A.2: The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.)

Web Content Accessibility Guidelines (WCAG 1.0)		W3C	Section 508
<b>Guideline 3: Use markup and style sheets and do so properly.</b>			
3.1	When an appropriate markup language exists, use markup rather than images to convey information.	P2	
3.2	Create documents that validate to published formal grammars.	P2	
3.3	Use style sheets to control layout and presentation.	P2	
3.4	Use relative rather than absolute units in markup language attribute values and style sheet property values.	P2	
3.5	Use header elements to convey document structure and use them according to specification.	P2	
3.6	Mark up lists and items properly.	P2	
3.7	Mark up quotations. Do not use quotation markup for formatting effects such as indentation.	P2	
<b>Guideline 4: Clarify natural language usage</b>			
4.1	Clearly identify changes in the natural language of a document's text and any text equivalents.	P1	
4.2	Specify the expansion of each abbreviation or acronym in a document where it first occurs.	P3	
4.3	Identify the primary natural language of a document.	P3	
<b>Guideline 5: Clarify natural language usage.</b>			
5.1	Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.	P1	C
5.2	Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.		H
5.3	Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent which can be linearized.	P2	
5.4	If a table is used for layout, do not use any structural markup for the purpose of visual formatting.	P2	
5.5	Provide summaries for tables.	P2	
5.6	Provide abbreviations for header labels.	P2	

Table A.3: The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.)

Web Content Accessibility Guidelines (WCAG 1.0)		W3C	Section 508
<b>Guideline 6: Ensure that pages featuring new technologies transform gracefully.</b>			
6.1	Documents shall be organized so they are readable without requiring an associated style sheet.		D
6.2	Ensure that equivalents for dynamic content are updated when the dynamic content changes.	P1	
6.3	Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If not, provide equivalent information on a alternative accessible page. [P3]	P1	
6.4	For scripts and applets, ensure that event handlers are input device-independent.	P2	
6.5	Ensure that dynamic content is accessible or provide an alternative presentation or page.	P2	
<b>Guideline 7: Ensure user control of time-sensitive content changes.</b>			
7.1	Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	P1	J
7.2	Allows users to control blinking, avoid causing content to blink (ie. Change presentation at a regular rate, such as turning on and off.	P2	
7.3	Allow users to freeze moving content, avoid movement in pages.	P2	
7.4	Provide the ability to stop the refresh, do not create periodically auto-refreshing pages.	P2	
7.5	Do not use markup to redirect pages automatically. Instead configure the server to perform redirects.	P2	
<b>Guideline 8: Ensure direct accessibility of embedded user interface.</b>			
8.1	Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies P1 – if functionality is important and not presented elsewhere, otherwise P2.	P1/P2	

Table A.4: The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.)

Web Content Accessibility Guidelines (WCAG 1.0)		W3C	Section 508
<b>Guideline 9: Design for device-independence.</b>			
9.1	Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	P1	F
9.2	Ensure that any element that has its own interface can be operated in a device-independent manner.	P2	
9.3	For scripts, specify logical event handlers rather than device-dependent event handlers.	P2	
9.4	Create a logical tab order through links, form controls, and objects.	P3	
9.5	Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.	P3	
<b>Guideline 10: Use interim solutions.</b>			
10.1	Until User agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.	P2	
10.2	Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.	P2	
10.3	Until user agents (including assistive technologies) render side-by-side text correctly, provide a linear text alternative ( on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.	P3	
10.4	UUA - handle empty controls correctly, include default, place holding characters in edit boxes and text areas.	P3	
10.5	UUA (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.	P3	
<b>Guideline 11: Use W3C Technologies and guidelines.</b>			
11.1	Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.	P2	
11.2	Avoid deprecated features of W3C technologies.	P2	
11.3	Provide information so that users may receive documents according to their preferences. (e.g. language, content type,etc.)	P3	
11.4	A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.		K

Table A.5: The WCAG guidelines and cross-correlation with Section 508 of the Disabilities Act (cont.)

Web Content Accessibility Guidelines (WCAG 1.0)		W3C	Section 508
<b>Guideline 12: Provide context and orientation information.</b>			
12.1	Documents shall be organized so they are readable without requiring an associated style sheet.		I
12.2	Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.	P2	
12.3	Divide large blocks of information into more manageable groups where natural and appropriate.	P2	
12.4	Associate labels explicitly with their controls.	P2	
<b>Guideline 13: Provide clear navigation mechanisms.</b>			
13.1	Clearly identify the target of each link.	P3	
13.2	Provide meta data to add semantic information to pages and sites.	P2	
13.3	Provide information about the general layout of a site.	P2	
13.4	Use navigation mechanisms in a consistent manner.	P2	
13.5	Provide navigation bars to highlight and give access to the navigation mechanism.	P3	
13.6	Group related links, identify the group (for user agents) and UUA do so, provide a way to bypass the group.	P3	
13.7	If search functions are provided, enable different types of searches for different skill levels and preferences.	P3	
13.8	Place distinguishing information at the beginning of headings, paragraphs, lists, etc.	P3	
13.9	Provide information about document collections(i.e. documents comprising multiple pages.)	P3	
13.10	Provide a means to skip over multi-line ASCII art.	P3	
<b>Guideline 14: Ensure that documents are clear and simple.</b>			
14.1	Use the clearest and simplest language appropriate for a site's content.	P1	
14.2	Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.	P3	
14.3	Create a style of presentation that is consistent across pages.	P3	

Table A.6: Section 508 paragraphs related to user agent recommendations

Web Content Accessibility Guidelines (WCAG 1.0)		W3C	Section 508
<b>Others: Paragraph (l);(m);(n);(o) relate to scripts, applets and forms (partial user agent). Conformity to WCAG 1.0 Level A means meeting all P1 issues. To satisfy Section 508 Guidelines, the paragraphs l,m,n,o,p have to be implemented, in addition.</b>			
6.3 6.4 8.1 9.3	When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.		L
6.3 6.4 8.1	When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with 1194.21(a) through (l).		M
10.2 12.4 9.3	When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.		N
13.5, 13.6	A method shall be provided that permits users to skip repetitive navigation links.		O
	When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.		P

## Appendix B

# Experimenter's Handbook for WebCT

Experimenter: _____	Date: _____
User Code: _____	Time: _____

Check the equipment set up for this particular testing session.

Machine	OS	Browser	Version	Adaptive Tool	Version
Fjord	WIN NT	IE		Jaws	
Shohrehpc	WIN 95	Netscape		Zoomtext	
Sulemanpc	WIN XP			Braille	
Other					

Before you start the tool testing double check the following:

<b>Account setup</b>	<b>Yes</b>	<b>No</b>	<b>Comments</b>
User have a NetLink ID?			
WebCT account been setup?			
<b>Forms</b>	<b>Yes</b>	<b>No</b>	<b>Comments</b>
Have Pre-Questionnaire?			
Have Consent form?			
<b>Equipment</b>	<b>Yes</b>	<b>No</b>	
Video camera started?			
Is microphone on?			
Make notes of any difficulties or questions asked?			

## B.1 Pre-Study Questionnaire

A pre-study questionnaire and a consent form will have already been sent out to all the participants via e-mail.

→ Check that we have received both their questionnaire and consent form.

**Q1** What is your sex?

I am a Female    I am a Male

**Q2** What is your age group?

under 25 years    25-45 years    45 years and over

**Q3** What is the language you first learned as a child?

English    French    Other (Specify)

**Q4** Did you or are you considering taking a course over the Internet?

(No) No

(Yes) If yes, then please specify:

Was the material supplemented over the internet?

Was it offer via a web based learning tool like WebCT or Black Board?

**Q5** Do you have access to a computer which you could use for your Internet course-work?

(No) If no, then please skips ahead to Question 12.

(Yes) If yes, then please answer the following questions.

**Q6** Where is this computer located? (Indicate "Yes" or "No" for each location.)

(No) (Yes) At home

(No) (Yes) At work

(No) (Yes) Other (Please specify)

Please answer question 7 through 12 for the computer you would use most often for coursework.

**Q7** Which one of the following best describes this computer? (Check one)

Pentium-based computer

Macintosh

486-based computer

Laptop (Specify: 386/486/Pentium)

Other (Please specify)

**Q8** How many megabytes of RAM (Ramdon Access Memory) is in this computer?

Less than 16

16 to 31.9

32 to 63.9

64 to 127.9

128 or more

**Q9** Does this computer have a modem?

(No) (Yes)

less than 28.8

28.8

33.6

56.6

ISDN

Ethernet connection (high speed, ADSL, cable modem, LAN)

Other (Please specify)

**Q10** Which Web browser do you prefer to use when you are accessing the World Wide Web?

Internet Explorer Netscape Mosaic

Opera Lynx Other Web browser

What version is your Web browser?

**Q11** What operating system is installed in this computer? (Indicate "Yes" or "No" for each operating system.)

- (No)(Yes) MS Dos
- (No)(Yes) Windows 3.x
- (No)(Yes) Windows 95/98
- (No)(Yes) Windows NT
- (No)(Yes) Windows XP
- (No)(Yes) Macintosh
- (No)(Yes) Linux
- (No)(Yes) Other (Please specify)

**Q12** Indicate which of the following computer applications you have used. (Indicate "Yes" or "No" for each application.)

- (No) (Yes) Word processing
- (No) (Yes) Spreadsheets
- (No) (Yes) Databases
- (No) (Yes) Statistical Programs
- (No) (Yes) E-mail
- (No) (Yes) Library electronic searches
- (No) (Yes) Browsing the Web
- (No) (Yes) Web Design
- (No) (Yes) Newsgroups
- (No) (Yes) Gopher
- (No) (Yes) Listserves
- (No) (Yes) Games
- (No) (Yes) Programming
- (No) (Yes) Home based utility programs  
(personal taxes, household budgeting, etc.)
- (No) (Yes) Other (Please specify)

**Q13** Overall, how would you rate your level of comfort level working in front of a computer on this scale from 1 to 6, with 1="Very Uncomfortable" to 6="Very Comfortable". Please select the option that best describes you working in front of a computer. I am

1. Very Uncomfortable
2. Somewhat Uncomfortable
3. Slightly Comfortable
4. Somewhat Comfortable
5. Very Comfortable

**Q14** Overall, how would you rate your level of computer ability on this scale from 1 to 6, with 1="novice" to 6="expert". Please select the option that best describes you.

Your computer ability is:

1. Novice skill level
2. below average skill level
3. Slightly below average skill level
4. Above average skill level
5. Expert skill level

**Q15** On this same scale, how would you rate your computer keyboard skills?

Your keyboard skills are:

1. Novice skill level
2. Below average skill level

3. Slightly below average skill level
4. Above average skill level
5. Expert skill level

**Q16** We would like to know what disabilities/impairments you have and how long you have had them. The first part of this question asks “No” or “Yes” about specific impairments. Just click “Yes” on the ones that apply to you.

- (No) (Yes) totally blind
- (No) (Yes) visually impaired/partially sighted
- (No) (Yes) deaf
- (No) (Yes) hearing impaired/hard of hearing
- (No) (Yes) speech or communication impaired
- (No) (Yes) learning disabled
- (No) (Yes) wheelchair user
- (No) (Yes) mobility impaired (eg. walk with difficulty, use a cane)
- (No) (Yes) difficulty using hands/arms
- (No) (Yes) health or medically related impairment (eg. diabetes)
- (No) (Yes) psychological or psychiatric disability
- (No) (Yes) other (specify impairment/disability)

**Q17** In this part we want to know how long have you had your disability or multiple disabilities? Please use the 3-point scale where 1=all or most of my life (before age 10), 2=part of my life and 3=recent (past 3 years)

1. all or most of my life (before age 10)
2. part of my life
3. recent (past 3 years)

**Q17a** In this part please describe your disability or multiple disabilities in the box below, that you think might affect your success with online education.

**Q18** Which one of the following adaptive computer technologies you use in order to get your work done:

(Please indicate product names if you know them.)

a. **screen reader (software that reads what's on the screen)**

product name: \_\_\_\_\_

b. **software that enlarges what is on the screen**

product name: \_\_\_\_\_

c. **a scanner**

product name: \_\_\_\_\_

d. **Braille translation software**

product name: \_\_\_\_\_

e. **a portable note taking device (not referring to a laptop)**

product name: \_\_\_\_\_

f. **a large screen monitor**

product name: \_\_\_\_\_

g. **a Braille printer**

product name: \_\_\_\_\_

**h. a spell checker**

product name: \_\_\_\_\_

**i. other specialized software for learning disabilities (eg. word prediction)**

product name: \_\_\_\_\_

**j. keyboard adaptations (eg. sticky keys)**

product name: \_\_\_\_\_

**k. mouse adaptations (eg. head mouse, track ball)**

product name: \_\_\_\_\_

**l. dictation software (voice recognition software that type what you say)**

product name: \_\_\_\_\_

**m. voice control software (you give voice commands like "file", "open", eg. VoicePad)**

product name: \_\_\_\_\_

**n. having material available in electronic format (eg. books, handouts)**

product name: \_\_\_\_\_

**o. other (specify adaptive hardware/software and product name)**

product name: \_\_\_\_\_

**Computer Skills Assessment Checklist**

For the remaining 13 items on this page, please indicate your assessment of your current skill level for each task on a three point scale: don't know how, have tried it but not sure how, or do know how.

**Q19** resize/move windows

- a. Don't know how to resize/move windows
- b. Tried before but not really sure how to resize/move windows
- c. Know how to resize/move windows

**Q20** work with multiple applications at the same time

- a. Don't know how to work with multiple applications at the same time
- b. Tried before but not really sure how to work with multiple applications at the same time
- c. Know how to work with multiple applications at the same time

**Q21** cut and paste within and between applications

- a. Don't know how to cut and paste within and between applications
- b. Tried before but not really sure how to cut and paste within and between applications
- c. Know how to cut and paste within and between applications

**Q22** format and copy disks

- a. Don't know how to format and copy disks
- b. Tried before but not really sure how to format and copy disks
- c. Know how to format and copy disks

**Q23** save a word processing document to disk

- a. Don't know how to save a word processing document to disk

- b. Tried before but not really sure how to save a word processing document to disk
- c. Know how to save a word processing document to disk

**Q24** open an e-mail program

- a. Don't know how to open an e-mail program
- b. Tried before but not really sure how to open an e-mail program
- c. Know how to open an e-mail program

**Q25** read messages in an e-mail program

- a. Don't know how to read messages in an e-mail program
- b. Tried before but not really sure how to read messages in an e-mail program
- c. Know how to read messages in an e-mail program

**Q26** send e-mail messages

- a. Don't know how to send e-mail messages
- b. Tried before but not really sure how to send e-mail messages
- c. Know how to send e-mail messages

**Q27** delete e-mail messages

- a. Don't know how to delete e-mail messages
- b. Tried before but not really sure how to delete e-mail messages
- c. Know how to delete e-mail messages

**Q28** get on to the Internet

- a. Don't know how to get on to the Internet
- b. Tried before but not really sure how to get on to the Internet
- c. Know how to get on to the Internet

**Q29** follow links

- a. Don't know how to follow links
- b. Tried before but not really sure how to follow links
- c. Know how to follow links

**Q30** do simple searches

- a. Don't know how to do simple searches
- b. Tried before but not really sure how to do simple searches
- c. Know how to do simple searches

**Q31** go to specific sites through URLs

- a. Don't know how to go to specific sites through URLs
- b. Tried before but not really sure how to go to specific sites through URLs
- c. Know how to go to specific sites through URLs

## B.2 Orientation

1. Introduce yourself to the user, and all other observers. Make the user relaxed.
2. Review the participant's consent form.
  - Check if the user requested not to be videotaped. If so make sure the recorder is turned off.

3. Their pre-study questionnaire should be attached to the back of this package.
  - (a) If not, please ask them to sign the consent form and fill in the questionnaire. Then briefly go over it with them and fill in any missing or ambiguous answers.
4. Briefly state the purpose of the experiment:
  - “We want to test the accessibility of Web-based learning tools with the adaptive tools for browsing and understanding course content in order to participate in an online course.”
5. Ask the user again to relax, state the follow:
  - “We are not testing the user; we are testing the user interface and the accessibility and usability of the tool. The key point is to observe how the tool’s interface helps or hinders the user’s performance while carrying out the tasks in the allotted time.”
  - “We are interested to see how user attempts to complete a task with the given tool and the respective assistive device.”
6. Outline the phases of the experiment (see below) and clearly state that the time to complete these phases and the tasks is limited.

Break down of the users voluntary time commitment.

Objectives	Approx. Time	Expected tasks
Orientation	30 min	There will be a short orientation to the study and the environment they will be working on.
Adaptive equipment setup	30 min	Session to setup the users equipment according to their likes or what they are comfortable with.
Warm up	30 min	The participant will be given the program "Hangman" to test the tools' functionality.
Access Tasks	60 min	Completion of a set of course content accessibility testing tasks to find where barriers exist within the testing environment.
Post questionnaire/Interview	15 min	Completion of a short questionnaire/interview, concerning the tasks they were asked to perform.

7. State that "the user is not expected to be able to complete all the tasks in the given time."

### B.3 Assistive Tool Setup

**Objective:** To observe the users habits in setting up their work environment; therefore, each participant will setup their perspective assistive tool.

**Observations:**

Browser: Was there any settings that was modified here?
AT: (ie. What settings was changed, color scheme or the jaws reader)
Other

## B.4 Warm-up Session

**Objective:** To double check the users' comfort level with the browser and see how the user performs various tasks on the internet. This session will help ensure that the user is comfortable with navigating the internet and help us to determine if a user's

difficulty was due to a lack of access with some of the target tool features.

In the practice phase, the user will do some hands-on practice an exploration of the Internet features to complete some low-level tasks.

1. Search for a specific topic (find a specific url for this task) on the internet.

Rank:

Comment:

2. Download selected paper (give them a paper's title to search) for review.

Rank:

Comment:

## B.5 Access Tasks

During the task phase of the experiment, the user will attempt some program understanding tasks on a larger program using the features of the tool.

### Access Scale Ranking

1. Was unable to access. Required observer to intervene physically (inaccessible)
2. Was able to access with instruction, but with no physical help (instructional, how to)
3. Was able to access with a hint (eg. Try tabbing, using the down arrow)
4. Was able to access but asked for clarification/affirmation (eg. Am I in a text area?)
5. Was able to access without instruction or physical help from the observer (accessible)

1. Login to the My WebCT.

Rank:

Comment:

2. Find out who is your instructor for the course.

Rank:

Comment:

3. Find out the days and the times of the lectures.

Rank:
Comment:

4. Find the location of the lectures (which building and lecture room).

Rank:
Comment:

5. Find the text book for the course.

Rank:
Comment:

6. Find the final exam date for the course.

Rank:

Comment:

7. Find deadline for the first assignment.

Rank:

Comment:

8. View the course outline linked from the Homepage.

Rank:

Comment:

9. Find lecture notes **for the last class (I have to have a specific lecture notes)**.

Rank:
Comment:

10. Review the lecture note and look for the **(I have to ask for a specific information within the lecture that they have to look for)**.

Rank:
Comment:

11. After you've finished reading the notes and found the information, use the note taking tool (My Notes in the top frame) to enter a comment about your experience.

Rank:
Comment:

12. Send an email to the instructor about setting up a meeting for the upcoming project.

Rank:
Comment:

13. Submit a file, any small file will do, using the "Assignment Dropbox", which is linked from the Tools page.

Rank:
Comment:

14. Access your work space, and familiarize yourself with it. Your work space is linked from the Tools page.

Rank:
Comment:

15. View the class list while learning about your work space to see who else is in the course.

Rank:
Comment:

16. Complete the online quiz, linked from the Home page or Tools page.

Rank:
Comment:

17. Logout of My WebCT.

Rank:
Comment:

## **B.6 Post-Questionnaire/Interview**

1. Read the questions to the user and audio tape the response.
2. Encourage the user to answer the questions quickly; first impressions are fine.

3. Do not talk to the user as he or she answering the Post Study questionnaire.
1. Did you find WebCT useful while solving the prescribed tasks?
2. What did you find difficult or cumbersome during the assigned tasks?
3. Which part of the exercise presented accessibility barriers?
4. What features do you think the tool is missing?
5. Do you have any suggestions on how the course content accessibility can be improved?
6. Would you be inclined to use a distributed learning tool such as WebCT as a mean for taking distant courses?

## Appendix C

# Participant Consent Form

You are being invited to participate in a study entitled: “**Accessibility Issues in the Virtual Classroom Environment for the Blind - A Case Study**” that is being conducted by the Master of Science graduate student Shohreh Hadian under the supervision of Dr. M. Storey.

Dr. Storey is an Associate Professor in the of Computer Science at the University of Victoria and you may contact her if you have further questions by calling 721-8796 or e-mailing [mstorey@csc.uvic.ca](mailto:mstorey@csc.uvic.ca). Shohreh Hadian will be conducting and supervising the tests and her telephone number is 472-4239 and e-mail is [shadian@uvic.ca](mailto:shadian@uvic.ca).

The objectives of this study are to address the accessibility issues related to blind persons in Web-based learning (virtual classrooms) environments. The assistive technologies used to access the courseware tools will depend on the nature of the disabilities. The results of this study will aim to solve the accessibility problems faced by blind people in a virtual classroom. We are recruiting all of our participants from University of Victoria. Secondary objectives include the provision of directives for future

development of online courseware and for providing tools to educators who work in a Web setting with persons with disabilities. You will be asked about your impressions and opinions of the tools' effectiveness, and to indicate which tool features, if any, caused difficulties.

If you agree to voluntarily participate in this research, your participation will require approximately 4 hours in the study. The following table details how this time will be spent:

<b>Approx. Time</b>	<b>Objective</b>	<b>Expected tasks</b>
45 min	Pre-study questionnaire	An e-mail containing a consent form and a pre-study questionnaire will be sent out. It will ask for and the type of assistive tools being used.
2.5 hrs	Tool testing	Each participant should complete the testing of the selected tool using their respective assistive technology which is appropriate to the disability. (3 sessions or 2 sessions of 1.5 & 1 hours)
30 min	Orientation	There will be an orientation to the study.
30 min	Setup assistive tool	Each participant will modify their perspective assistive tool to their liking.
30 hrs	Warm up session	Users will have a chance to practice on sample tasks.
60 min	Access tasks	Use the distributed learning tool to complete a series of course content accessibility testing tasks. We will ask you to think-aloud (verbalize your thoughts) as you complete these tasks.
30 min	Post questionnaire/interview	You will complete a short questionnaire/interview, concerning the tasks you were asked to complete.

There are no known or anticipated risks in this research. The potential benefits of your participation in this research include: you may find the tasks enjoyable and challenging. You may also find the debriefing explanation of the study and the distributed learning tool interesting.

Participation in this study may cause some inconvenience to you. Participants may find the tasks boring and/or difficult. Participants may experience frustration-

using tools under investigation with unknown errors.

Your participation in this research must be completely voluntary. You may withdraw at any time without any consequences or any explanation. If you choose to withdraw from the study, we may ask permission to keep the data we collected from you. However, you may also choose to have the data destroyed and no one will see it.

Eliminating all identifying information from the data will protect anonymity. For example, data will be stored separately from participant's names, and any statements made during the study that are potentially identifying will be deleted from the video/audio records. However, as we will be videotaping the sessions, only the screen and your voice will be recorded.

Keeping the information in a secure location will protect your confidentiality, and the confidentiality of the data. Even though, we are not capturing the participants face and only the screen movements are being captured. Following the study, participants will be allowed to listen to their comments during the session. Participants will be asked if the recordings can be viewed by:

1. experimenters only
2. experimenters and other researchers, or
3. experimenters, other researchers, and shown for educational purposes. Participants will determine the permissions for who will view their tapes.

Data from this study will not be disposed of. This means that the videotapes, transcriptions of the session will not be destroyed.

It is anticipated that the results of this study will be shared with others through published articles and presentations and may be published on the web.

In addition to being able to contact the researcher at the above phone numbers, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Associate Vice President Research at the University of Victoria (250)-721-7968.

Your signature below indicates that you understand the above conditions of participation in this study and that you have had the opportunity to have your questions answered by the researchers.

Permissions for videotape (to be completed after the study):

The videotape of my session may be viewed by:

1. experimenters, other researchers, and it may be shown to others for educational purposes;
2. experimenters and other researchers; or
3. experimenters only.

<i>Name of Participant</i>	<i>Signature</i>	<i>Date</i>

***A copy of this consent will be left with you, and a copy will be taken by the researcher.***

## PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my dissertation to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this dissertation for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this dissertation for financial gain shall not be allowed without my written permission.

Title of Dissertation:

Accessibility in a Virtual Classroom: a Case Study for the  
Visually Impaired Using WebCT

Author:

*Sh. Hadian*  
Shohreh Hadian

*Sep 7, 2004*  
Date