



# **IMS Guidelines for Developing Accessible Learning Applications**

**Version 1.0 White Paper**

**Copyright © 2002 by IMS Global Learning Consortium, Inc.  
All Rights Reserved.**

The IMS Logo is a trademark of IMS Global Learning Consortium, Inc.  
Document Name: IMS Guidelines for Developing Accessible Learning Applications  
Date: 27 June 2002

# Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>1. INTRODUCTION</b> .....	<b>5</b>
1.1 ROLES AND RESPONSIBILITIES .....	5
1.2 ISSUES WITH LEARNING TECHNOLOGIES FOR PEOPLE WITH DISABILITIES.....	6
1.3 ACKNOWLEDGEMENTS .....	7
1.4 NOMENCLATURE.....	7
<b>2. PRIMER ON ACCESSIBILITY</b> .....	<b>8</b>
2.1 DISABILITIES, FUNCTIONAL LIMITATIONS, AND ACCESSIBILITY TIPS.....	8
2.1.1 <i>For People Who Are Blind</i> .....	8
2.1.2 <i>For People with Low-Vision</i> .....	9
2.1.3 <i>For People with Color Blindness</i> .....	9
2.1.4 <i>For People Who Are Hard-of-Hearing or Deaf</i> .....	9
2.1.5 <i>For People with Physical Disabilities</i> .....	10
2.1.6 <i>For People with Language or Cognitive Disabilities</i> .....	10
2.1.7 <i>General Accessibility Improvements</i> .....	11
2.2 TOOLS FOR ACCESS - TYPES OF ASSISTIVE TECHNOLOGIES .....	11
2.2.1 <i>Screen Readers</i> .....	11
2.2.2 <i>Refreshable Braille Displays</i> .....	12
2.2.3 <i>Screen Magnifiers</i> .....	12
2.2.4 <i>Adaptive Keyboards</i> .....	12
2.2.5 <i>Voice Recognition Software</i> .....	12
2.2.6 <i>Single Switches</i> .....	12
2.3 EQUIVALENT ACCESS VERSUS ALTERNATIVE ACCESS .....	13
2.4 DIRECT ACCESS VERSUS COMPATIBLE ACCESS .....	13
<b>3. PRINCIPLES FOR ACCESSIBILITY IN ONLINE DISTRIBUTED LEARNING</b> .....	<b>15</b>
3.1 ALLOW FOR CUSTOMIZATION BASED ON USER PREFERENCE.....	15
3.2 PROVIDE EQUIVALENT ACCESS TO AUDITORY AND VISUAL CONTENT BASED ON USER PREFERENCE.....	15
3.3 PROVIDE COMPATIBILITY WITH ASSISTIVE TECHNOLOGIES AND COMPLETE KEYBOARD ACCESS.....	16
3.4 PROVIDE CONTEXT AND ORIENTATION INFORMATION .....	16
3.5 FOLLOW IMS SPECIFICATIONS AND OTHER RELEVANT SPECIFICATIONS, STANDARDS, AND/OR GUIDELINES.....	16
3.6 CONSIDER THE USE OF XML.....	17
<b>4. USING XML FOR ACCESSIBILITY</b> .....	<b>19</b>
4.1 XML IS CUSTOMIZABLE AND FLEXIBLE .....	19
4.2 XML ALLOWS FOR STANDARDIZATION FOR ACCESSIBILITY, VALIDATION, AND WELL-FORMEDNESS .....	19
4.3 RECOMMENDED XML-BASED LANGUAGES.....	19
4.3.1 <i>XHTML</i> .....	20
4.3.2 <i>XML Mark-up and Stylesheets</i> .....	20
4.3.3 <i>Scalable Vector Graphics (SVG)</i> .....	20
4.3.4 <i>Synchronized Multimedia Integration Language (SMIL)</i> .....	21
4.3.5 <i>XML E-Book Formats</i> .....	22
4.3.6 <i>Other Mark-up Languages</i> .....	22
<b>5. GUIDELINES FOR ACCESSIBLE DELIVERY OF TEXT, AUDIO, IMAGES, AND MULTIMEDIA</b> .....	<b>23</b>

5.1	COMMON TYPES OF MEDIA DELIVERY AND ASSOCIATED PRESENTATION FORMATS .....	24
5.1.1	Text .....	24
5.1.2	Audio .....	24
5.1.3	Images .....	25
5.1.4	Multimedia .....	25
<b>6.</b>	<b>GUIDELINES FOR DEVELOPING ACCESSIBLE ASYNCHRONOUS COMMUNICATION AND COLLABORATION TOOLS .....</b>	<b>27</b>
6.1	THREADED MESSAGE BOARDS .....	27
6.2	E-MAIL MESSAGING.....	28
6.3	DOCUMENT REPOSITORIES .....	28
6.4	ORGANIZERS, SCHEDULERS, AND CALENDARS .....	29
6.5	PRESENTATION TOOLS .....	30
6.5.1	Microsoft PowerPoint.....	30
6.5.2	Alternatives to Microsoft PowerPoint for Accessibility.....	30
<b>7.</b>	<b>GUIDELINES FOR DEVELOPING ACCESSIBLE SYNCHRONOUS COMMUNICATION AND COLLABORATION TOOLS .....</b>	<b>31</b>
7.1	SYNCHRONOUS TEXT CHAT .....	31
7.2	AUDIO-CONFERENCING.....	31
7.3	VIDEO-CONFERENCING .....	32
7.4	WHITEBOARDS.....	33
7.5	MULTI-USER DOMAIN OBJECT ORIENTED ENVIRONMENTS (MOOS).....	33
<b>8.</b>	<b>GUIDELINES FOR DEVELOPING ACCESSIBLE INTERFACES AND INTERACTIVE ENVIRONMENTS.....</b>	<b>35</b>
8.1	INTERFACE CONTROLS.....	35
8.2	NAVIGATING THE INTERFACE.....	35
8.3	FORMS .....	36
8.4	INTERACTIVE EXERCISES: DRAG-AND-DROP EXERCISES, SIMULATIONS, AND TIMED TESTS .....	36
8.5	INTERACTIVE TUTORIALS .....	37
8.6	DVDS, CONSUMER ELECTRONICS, AND HANDHELD DEVICES .....	38
8.7	ACCESSIBILITY INFORMATION FOR OPERATING SYSTEMS AND DEVELOPMENT PLATFORMS .....	39
8.7.1	Microsoft's Windows OS Accessibility.....	39
8.7.2	Apple's Macintosh OS Accessibility.....	39
8.7.3	Web Accessibility.....	40
8.7.4	The Java Platform.....	40
8.7.5	Other Operating Systems .....	40
<b>9.</b>	<b>GUIDELINES FOR TESTING AND ASSESSMENT.....</b>	<b>41</b>
9.1	TESTING AND ASSESSMENT CHALLENGES.....	42
9.2	PRINCIPLES FOR THE ACCESSIBILITY OF TESTING AND ASSESSMENT .....	42
9.2.1	Consider Accessibility Standards from the Earliest Design Stages.....	42
9.2.2	Follow Standards for Testing and Assessment That Promote Test Validity .....	42
9.2.3	Build a Coherent Rationale for the Assessment.....	44
9.2.4	Develop a Reuse Strategy That Includes Test Content, Designs, and More.....	45
9.3	DELIVERY AND AUTHORING TOOL PRINCIPLES FOR TESTING AND ASSESSMENT.....	45
9.4	CONTENT DEVELOPMENT PRINCIPLES FOR TESTING AND ASSESSMENT.....	45
9.4.1	Provide Information about Purpose.....	45
9.4.2	Provide Information about Accessibility of Content at Varying Levels.....	45

<b>10.</b>	<b>GUIDELINES FOR DEVELOPING ACCESSIBLE AUTHORING TOOLS.....</b>	<b>47</b>
<b>11.</b>	<b>GUIDELINES FOR TOPIC SPECIFIC ACCESSIBILITY .....</b>	<b>48</b>
11.1	MATHEMATICS.....	48
	11.1.1 <i>Chunking of Mathematical Expressions.....</i>	49
	11.1.2 <i>Inaccessible Mathematical Notation.....</i>	49
	11.1.3 <i>Encoding Mathematical Expressions Using MathML .....</i>	50
	11.1.4 <i>Localized Applications for Mathematical Accessibility.....</i>	50
	11.1.5 <i>Accessible Mathematical Applications and Devices.....</i>	51
11.2	SCIENCE .....	51
	11.2.1 <i>Chemistry .....</i>	52
	11.2.2 <i>Physics.....</i>	52
11.3	SIMULATIONS AND IMMERSION .....	53
11.4	ROBOTS AND TELEPRESENCE.....	53
11.5	CHARTS, DIAGRAMS, AND TABLES .....	54
	11.5.1 <i>Haptic Perception .....</i>	54
	11.5.2 <i>Haptic Image Sources .....</i>	55
	11.5.3 <i>Text Representations of Charts, Diagrams, and Tables.....</i>	55
11.6	GEOGRAPHY AND MAPS.....	56
11.7	MUSIC .....	56
11.8	LANGUAGES .....	57
	11.8.1 <i>Ruby Annotation as a Solution for Language Accessibility .....</i>	58
	<b>APPENDIX A – LEGAL ISSUES FOR ACCESSIBLE DISTANCE LEARNING .....</b>	<b>59</b>
	<b>APPENDIX B – RESOURCES .....</b>	<b>61</b>
	<b>ABOUT THIS DOCUMENT .....</b>	<b>68</b>
	LIST OF CONTRIBUTORS .....	68
	<b>REVISION HISTORY .....</b>	<b>69</b>
	<b>INDEX.....</b>	<b>70</b>

# 1. Introduction

The evolution of the Internet and other technologies has led to tremendous growth in opportunities to teach and learn outside of the traditional classroom. Over the past decade, the Internet has brought learning “online” and offers many advantages: it is convenient, available at any time of the day, and can be accessed nearly anywhere in the world. Clearly, online or distributed learning, offers tremendous potential to increase the availability and convenience of education. And if the technologies behind distributed learning are made universally accessible, it will also have the potential to reach a significant percentage of those with disabilities.

As with many types of products and technologies, including those used in online distributed learning, people with disabilities may be inadvertently excluded if accessibility is not considered and incorporated into products and technologies. If steps are not taken to incorporate accessibility into distributed learning, people with disabilities may be excluded from the many benefits offered by online technologies. However, accessibility is not only of concern to those with disabilities. The potential for online distributed learning expands when developers embrace the widest possible range of individual learning styles, preferences, and abilities.

Accessible design grants a wider range of learners more options and greater flexibility in learning. Inclusive design strategies, also known as “universal design,” can help meet the varying needs and limitations of end users, including limitations created by available computer power and bandwidth. Universal design can make stand-alone courseware, Web-based e-learning systems, and other content more accessible to a larger population, including people with disabilities. Presenting educational material in a variety of formats will also provide benefits to those with differing learning styles (visual, auditory, tactile) and will allow people to learn in their preferred learning style.

The guidelines developed by the IMS Accessibility Project Group and presented in this document, will provide a framework for the distributed learning community. This framework will highlight existing solutions, discuss the opportunities and strategies for their implementation and will identify areas where further development and innovation are required to ensure education that is truly accessible to anyone, anytime, anywhere.

Other standards and guidelines currently exist; however, the IMS Accessibility Guidelines are targeted at the distributed learning community and specifically address the challenges that exist in online education. The IMS Accessibility Guidelines are not meant to replace existing standards and guidelines, but instead to reference those resources and provide additional information and solutions compatible with existing recommendations. That said, some topics addressed in this document, such as mathematics, scientific, and musical notation guidelines, do not yet have mainstreamed or widely adopted solutions. In these cases, this document offers suggestions and indicates the direction of current research.

## 1.1 Roles and Responsibilities

There are existing policies and laws that govern accessibility for people with disabilities. But they vary greatly from country to country, state-to-state, province-to-province, and even from one educational institution to another. Even the motives behind the desire to address accessibility concerns vary from person-to-person and from company-to-company. Yet in order to make learning technologies accessible to everyone, all parties must share responsibility.

The stakeholders include:

- courseware and software vendors
- educational publishers
- authoring tool developers/vendors
- authors/content developers
- educational institutions (including administrators)
- educators/instructors
- administrative staff
- students

Each setting is unique, requiring its own mix of roles and responsibilities. Consider the challenge of achieving accessibility in a university that delivers online education.

To begin, the manufacturer of courseware development tools must make the authoring environment itself accessible. Vendors must also provide tools within that environment to help educators and other courseware authors to meet universal design criteria.

Even if a vendor has made all features of an authoring environment accessible, and has provided the tools that support creating accessible courseware, there is no guarantee that an authoring professor will take advantage of them, and students with disabilities may still be excluded. Therefore, responsibility falls heavily on educators as well.

Educational institutions are responsible for following their own government's regulations. Educational institutions can go further and create their own accessibility policies in support of students, faculty, administrators, and staff with disabilities. These policies might include timeframes that establish the amount of advance notice a student must give that institution in order for the institution to be able to provide accessible materials. Policies can also establish just who must be informed of these needs and the procedures that students must follow in making their needs known.

Finally, students should also be considered a crucial part of this process. They must take responsibility for informing professors and school administrators about their particular needs and preferences and to do so in a timely manner. The school may establish limits within which students must communicate their accessibility needs and the student may be expected to work within those limits.

Making each constituent aware of their role is a responsibility shared by all. Creating courseware is a collaborative process and not all participants recognize that their contribution may have an impact on accessibility. By fostering an environment where accessibility is openly supported and discussed, institutions can help to ensure that all participants fulfill their individual obligations with accessibility in mind.

This document is targeted at the online distributed learning community, which includes software developers, courseware vendors, educational publishers, content developers, administrators, educators/instructors, and students. It is intended to be useful to a wide audience, including readers with varying levels of technical expertise.

## 1.2 Issues with Learning Technologies for People with Disabilities

Teaching and learning materials are delivered in many forms: paper, audio, and videotape, CD-ROM, television, and over the Internet. However, online education has become the preferred means of retrieving up-to-date information. With its characteristic speed and flexibility, online learning takes advantage of a variety of technologies to facilitate learning and interaction between participants.

Online education technologies include:

- synchronous and asynchronous communication and collaboration tools such as e-mail, listservs, bulletin boards, whiteboards, chat rooms, videoconferencing, and teleconferencing.
- interactive environments, such as simulations, immersive experiences, and games.
- testing and evaluation tools, including self-assessment and multiple-choice testing.

Today, online content is varied and can include: text on a website, digital audio, digital video, animated images, and virtual reality environments. This content can be created in a variety of ways, utilizing a variety of authoring tools.

As technology advances, accessibility questions proliferate. Video is everywhere, yet are all videos captioned for deaf and hard-of-hearing viewers? Do all videos carry an audio component that describes visual images for blind and low-vision users? Do all software applications offer keyboard equivalents appropriate for users with mobility restrictions or vision impairments. Do all images on websites have labels (alt-text) for users who access content with the help of screen reading software? How can blind students access mathematical equations presented as graphics, which cannot be read by screen readers? Are testing protocols flexible enough? Are administrative matters such as course listings and course registration accessible?

Not all of these questions have answers based in existing technologies. But we need to examine all accessibility issues in order to ensure access to online education for everyone. This document will lay out guidelines for accessible learning. These guidelines include:

- common accessibility problems associated with each technology.

- practices that learning system developers can implement to enhance accessibility for all users.
- practices that content creators and/or educators can implement to enhance accessibility for all users.
- resources that provide best practices and solutions in use.

### 1.3 Acknowledgements

This project is supported by a grant awarded to the CPB/WGBH National Center for Accessible Media from the Learning Anytime Anywhere Partnerships (LAAP), a program administered by the Fund for the Improvement of Postsecondary Education (FIPSE), U.S. Department of Education (<http://www.ed.gov/offices/OPE/FIPSE/LAAP/>).

Additional support is provided through participation and contributions from the following organizations:

Blackboard	USA	<a href="http://www.blackboard.com/">http://www.blackboard.com/</a>
CETIS (Centre for Educational Technology Interoperability Standards)	UK	<a href="http://www.cetis.ac.uk/">http://www.cetis.ac.uk/</a>
DEST (Department of Education, Science, and Training)	Australia	<a href="http://www.dest.gov.au/">http://www.dest.gov.au/</a>
ETS (Educational Testing Service)	USA	<a href="http://www.ets.org/">http://www.ets.org/</a>
IMS Global Learning Consortium	USA	<a href="http://www.imsglobal.org/">http://www.imsglobal.org/</a>
Industry Canada	Canada	<a href="http://www.ic.gc.ca/">http://www.ic.gc.ca/</a>
NCAM (The CPB/WGBH National Center for Accessible Media)	USA	<a href="http://ncam.wgbh.org/">http://ncam.wgbh.org/</a>
The Open University	UK	<a href="http://www.open.ac.uk/">http://www.open.ac.uk/</a>
Sheffield Hallam University	UK	<a href="http://www.shu.ac.uk/">http://www.shu.ac.uk/</a>
UK eUniversities Worldwide	UK	<a href="http://www.ukeu.com/">http://www.ukeu.com/</a>
University of Toronto ATRC (Adaptive Technology Research Centre)	Canada	<a href="http://www.utoronto.ca/atrc/">http://www.utoronto.ca/atrc/</a>

### 1.4 Nomenclature

ASL	American Sign Language
ATRC	Adaptive Technology Research Centre, University of Toronto, Canada
AT	Assistive Technology
CETIS	Centre for Educational Technology Interoperability Standards (UK)
DEST	Department of Education, Science, and Training (Australia)
DTD	Document Type Definition
ETS	Educational Testing Service
NCAM	The CPB/WGBH National Center for Accessible Media
SVG	Scalable Vector Graphics
TTY	Text Telephone
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative of the W3C
WGBH	Boston, Massachusetts public broadcaster

## 2. Primer on Accessibility

### 2.1 Disabilities, Functional Limitations, and Accessibility Tips

This section introduces the reader to several types of disabilities affecting vision, hearing, physical capacity, and cognitive skills. Each disability presents unique challenges to computer users. The following subsections also include general strategies to meet those challenges and to increase accessibility for those users.

#### Resources:

See the Trace Center Accessible Software Guidelines for a comprehensive guide to creating accessible software at: [http://www.trace.wisc.edu/world/computer\\_access/software/](http://www.trace.wisc.edu/world/computer_access/software/)

This section is adapted with permission from a previous publication of the Trace R&D Center, the Application Software Design Guidelines written in 1994 and published at: [http://trace.wisc.edu/docs/software\\_guidelines/toc.htm](http://trace.wisc.edu/docs/software_guidelines/toc.htm)

#### 2.1.1 For People Who Are Blind

Many people who are legally blind retain some residual vision. Some may be able to see objects with the help of magnification. Others may be able to sense light and dark but little else. Because of the wide range of visual sensitivity found among those who are legally blind, a well-designed interface should assume that the end user has no vision, while allowing that person to make use of whatever residual vision they possess.

To access online material, blind users depend on screen reading software that digests the contents of the computer screen and sends information to a text-to-speech synthesizer or refreshable Braille display.

Application software developers can do much to support screen reading software and to help blind users perceive and understand screen content.

#### To support screen reading software developers can:

- use standard system tools to draw and erase all on-screen text and to display all cursors and pointers.
- use system standard on-screen controls whenever possible.
- define tools in toolbars, palettes, and menus as separate items. Do not create single graphics containing multiple objects. By keeping tools and other objects separate, the screen reader is better able to identify and name each tool for the user.
- embed descriptive text in graphic images in such a way as to make the text known to screen reading software. This addresses the problems that can arise when text is rendered as a graphic image and cannot be read by software.
- assign logical names to controls, even if the name is not visible on the screen. Screen readers can access this information and use it to describe the type and function of the control on the screen.
- track the system cursor with the mouse, even if the cursor is invisible. This allows the screen reading software to detect the mouse position when customized highlighting or focusing techniques are in use.
- use consistent and predictable screen and dialog layouts.
- avoid the use of “help” balloons that disappear whenever the hot spot or focus of the mouse changes. Try instead to lock the “help” balloon in place so that the user can move the cursor and continue to read the balloon.
- use single column text whenever possible.
- provide keyboard equivalents for all tools, menus, and dialog boxes.

#### Since screen readers can only read text (or give names to separately identifiable icons or tools), it is a good idea to:

- avoid assigning unlabeled “hot spots” to pictures for use as controls, unless they are redundant with menu selections.
- avoid non-text menu items when possible or at least incorporate visible or invisible text cues to accompany these items. Screen readers can see text even if that text is written to the screen invisibly.



- avoid non-redundant graphic tool bars. Make any tool bar command available in a menu.

**Finally, documentation and training materials are always more accessible when:**

- documentation and online help can be understood independent of graphics. Text descriptions should stand on their own.
- synchronized audio descriptions are available to play alongside animated graphics or movies.

### 2.1.2 For People with Low-Vision

**“Low-vision” refers to a range of vision problems including:**

- poor acuity, meaning blurred or fogged vision.
- loss of all central vision, the ability to see only the outer ring of the visual field.
- tunnel vision, the ability to see only the center of the normal visual field.
- loss of vision in other parts of the visual field.
- other problems, including sensitivity to glare, night blindness, or reduced contrast.

Computer users with low-vision often depend on the ability to enlarge or otherwise enhance areas of on-screen information. Screen enlargement software can be tremendously helpful.

**To make on-screen information easier to see, developers can:**

- increase the contrast between text and the background.
- place text over a solid-color background. A patterned background can make text harder to discern.
- create consistent layouts for all screens and dialogs within the program.
- provide access to tools via a menu bar.
- follow line width guidelines when drawing lines on screen. Use the line width information provided by operating system settings. This will ensure that the learning application will increase all lines proportionally should a user choose to enlarge the view.
- allow the user to zoom in to or magnify portions of the screen.

**To make their software more compatible with other applications that offer low-vision access features, developers can:**

- use the system pointers whenever possible, as well as the system caret or insertion bar, if available.
- include a highlight or focus indicator when dragging the system cursor, even at those times when the cursor is invisible. This adjustment will help screen enlargement software using “pan and zoom” features to track the user’s movements more accurately.
- add support for a “high contrast” setting.
- protect users from the need to simultaneously monitor two or more events occurring far apart from each other on the screen.

### 2.1.3 For People with Color Blindness

**To make it easier for color blind users to access their software, developers can:**

- make color-coding a redundant or secondary means of conveying information.
- ensure that the program will run in monochrome mode.
- use variations in contrast and brightness in addition to color variations.

### 2.1.4 For People Who Are Hard-of-Hearing or Deaf

Many users with hearing impairments need to have some method for adjusting the volume or for coupling audio output more directly to their hearing aids. These are hardware requirements that can be met by systems with built-in volume controls and headphone or audio output jacks. Users who have more severe hearing impairments may choose to combine these solutions with visual display technologies designed for people who are deaf.

**To increase the accessibility of software to users with hearing impairments, developers can:**

- provide all auditory information visually.
- ensure that all visual cues are noticeable even if the user is not looking straight at the screen. Important information should catch the user's attention, even through peripheral vision.
- support the ShowSounds feature that allows a user to assign a visual and caption for each audio event.

Finally, telephone support staff should also have text telephone (TTY) available in order to be able to assist deaf or hard-of-hearing customers.

**2.1.5 For People with Physical Disabilities**

The nature of physical disabilities varies widely. Some physically disabled users experience complete paralysis in some portion of their bodies. Others are restricted by muscular weakness. Some may have a very limited range of motion, but may possess very fine movement control within that range. Others may have feeling in their limbs, but little control over them. Others have to work with uncontrolled, sporadic movements that interfere with their purposeful movements. Users with arthritis report that the joints in the hand and elsewhere are restricted in their range of motion both mechanically and by pain. Some users have posture difficulties that can only be solved by mounting the computer screen in a non-standard orientation.

Physical disabilities, by themselves, do not usually affect a person's ability to perceive information displayed on the computer screen. They can interfere with the interactive experience by preventing users from easily manipulating the interface.

**To increase the accessibility of software to people with physical disabilities and ensure compatibility with assistive technologies, developers can:**

- avoid timed responses or when they cannot be avoided, lengthen the time allowed for a user to respond.
- provide keyboard access to all toolbars, menus, and dialog boxes.
- allow the user to access helpful features already built into the operating system, such as StickyKeys, SlowKeys, Key Repeating.

**2.1.6 For People with Language or Cognitive Disabilities**

Language and cognitive disabilities are very difficult for developers to address, partly because of the diversity of the category. The group includes individuals with:

- general processing difficulties such as mental retardation, brain injury, and others.
- specific deficits such as lack of short-term memory, the inability to remember proper names and others.
- learning disabilities such as dyslexia, dyscalculia, dysgraphia, auditory perceptual disabilities, cognitive disorganization, and visual perceptual disabilities.
- language delays.

In addition, the range of impairment within each of these categories can vary from minimal to severe. In general, software designed to be as user-friendly as possible will improve accessibility for those with language or cognitive impairments.

**To improve accessibility for people with language or cognitive disabilities, developers can:**

- allow all message alerts to remain on the screen until dismissed by the user.
- make language and instructions as simple and straightforward as possible, both on-screen and in documentation.
- use simple and consistent screen layouts.

It is important to bear in mind that those with language and cognitive disabilities often have difficulty processing print. To increase accessibility for this population, developers should take steps to make their software compatible with screen reading software (for more information, see subsection 2.1.1 For People Who Are Blind.)

Dyslexia is one of the most common learning disabilities. Affecting between 4 and 15 percent of the population, this neurological disorder interferes with the acquisition and processing of written language. Dyslexia is associated with difficulties in both receiving and expressing language, including phonological processing, in reading, writing, spelling, handwriting, and sometimes in mathematical reasoning. Dyslexic students can benefit from many of the solutions developed to help users with other disabilities.

**To improve accessibility for people with dyslexia, developers can:**

- give users the ability to select their preferred font.
- make their products compatible with screen reading software and voice recognition software.
- use consistent layouts and formats.
- use plain, uncomplicated backgrounds behind text.
- avoid flashing, moving, or animated text.
- List hyperlinks at the end of the relevant paragraph or section instead of within the general text.
- Describe the type of information or website pointed to by a link, before inserting the link into the text.

**Resources:**

U.S National Institute of Neurological Disorders and Stroke resource on dyslexia:

[http://www.ninds.nih.gov/health\\_and\\_medical/disorders/dyslexia\\_doc.htm](http://www.ninds.nih.gov/health_and_medical/disorders/dyslexia_doc.htm)

Learning Disabilities Resource Community: <http://www.ldrc.ca>

Trace Center resources on Learning Disabilities: [http://trace.wisc.edu/docs/quick\\_sheets/qs17.html](http://trace.wisc.edu/docs/quick_sheets/qs17.html)

### **2.1.7 General Accessibility Improvements**

**To increase overall accessibility, developers can:**

- make documentation available in electronic form, accessible by screen reading software, thus making it easier to use for people who cannot handle or read printed materials.
- make product support staff aware of disability access issues and that people with disabilities routinely depend on their software products.
- establish a protocol to forward accessibility and compatibility problems from product support staff to product designers. Keep the trigger-level for such referrals low.

Finally, all properly trained support staff should be able to handle routine product use questions from users who have disabilities. Ideally, some individual should also seek to specialize, to learn how to resolve specific incompatibility issues between software and technologies designed to assist those with disabilities. In this way, developers and support staff can create pools of expertise to deal with the widest possible range of problems and suggestions brought forward by users.

## **2.2 Tools for Access - Types of Assistive Technologies**

Assistive technology (AT) is an umbrella term used to describe any product or technology-based service that helps disabled people to live, learn, work, and enjoy life. In the context of online education, AT refers to hardware and software technologies that enable people with disabilities to use a computer more effectively. The following is a brief overview of the main categories of these assistive technologies.

### **2.2.1 Screen Readers**

Screen readers are software products designed for blind users. Screen readers are also useful to users with learning disabilities.

Screen readers locate information on the computer screen and vocalize it for the user using text-to-speech audio hardware and software. Most screen readers work in close concert with the operating system, relying on the computer's built-in capabilities. Applications and software that conform to the standards of the operating system are more likely to be accessible. Applications and software that ignore the requirements of screen readers and the operating systems that support them may well prove unusable by some disabled students.

### 2.2.2 Refreshable Braille Displays

Refreshable Braille displays are tactile devices that move small pins up and down to create Braille letters. These Braille displays are the primary means of access to computers for users who are deaf-blind.

### 2.2.3 Screen Magnifiers

Screen magnifiers are software solutions for users with low-vision. These products allow the user to enlarge the size of images and text displayed on screen. Screen magnifiers may also permit the user to change the default colors of the display. For example, reversing the colors and blacks of the video signal can make the text more readable for some users.

Compatibility between screen magnifiers and software can be an issue for developers. Typical screen magnifiers track the cursor or the active region of the screen and will automatically enlarge that portion of the display. Applications that use a custom cursor design may cause the magnifier to enlarge the wrong portion of the screen. Developers can avoid this problem by relying on standard interface practices, particularly those that apply to cursor control and display.

### 2.2.4 Adaptive Keyboards

Adaptive keyboards are designed for users with physical disabilities who cannot use a standard keyboard. Users with reduced range of motion may require smaller keyboards. Conversely, those without fine motor control may require a keyboard that is somewhat larger. Keyboards that offer fewer choices are helpful to users who benefit from a more structured learning environment and one-handed keyboards are helpful for those who can only type with one hand.

For users who are only able to use a mouse (or an AT that emulates a mouse) the keyboard itself can be represented on-screen using software. Pointing at individual letters replaces the physical act of typing.

Developers can take steps to support users of these technologies. Applications and software that employ the operating system's standard methods of reading input from the keyboard should be compatible with all adaptive keyboards. Those applications that bypass the operating system and attempt to interrogate the keyboard directly will probably not be accessible to users who wish to substitute an adaptive keyboard.

### 2.2.5 Voice Recognition Software

Voice recognition software allows the user to input data or control the computer by speaking. Voice recognition software benefits users who have difficulty typing or using their hands. Generally, applications and software that allow full access through keyboard commands are well suited for use with voice recognition software.

### 2.2.6 Single Switches

Single switches are hardware solutions for users with physical disabilities who can only control the computer with one or two specific movements. Single switches are used with software that works by scanning through preset options on screen. The user triggers the switch when the option they wish to choose has been highlighted during scanning. Single switches can be used in conjunction with on-screen keyboards and word prediction software. Scanning software can be used to create customized screen layouts for use with a variety of applications. However, every clickable spot in the layout must be identified in advance in order for the scanning software to find it.

#### Resources:

A *Technical Glossary of Adaptive Technologies* is available from the Adaptive Technology Resource Centre at the University of Toronto at: <http://www.utoronto.ca/atrc/reference/tech/techgloss.html>

## 2.3 Equivalent Access Versus Alternative Access

When considering accessibility of learning applications, it is important to understand the differences between two types of access: equivalent and alternative.

**Equivalent access** provides the disabled user with content identical to that used by the non-disabled user. For the disabled user however, that content is presented using a different modality. Providing a course textbook in Braille format, on audiotape, or in digital format are examples of equivalent accessibility.

**Alternative access** provides the disabled user with a learning activity that differs from the activity used by the non-disabled user. However, the alternative activity is designed to achieve the same learning objectives. For example, a mobility-impaired student might be given the option of conducting a science experiment in a virtual laboratory, where the levels of dexterity, strength, and physical access are different from those required in a physical laboratory.

Equivalent access should be provided whenever possible. Alternative access should be provided only if equivalent access is not possible. That said, there are numerous examples where software developed for alternative access has become the mainstream choice when its value to all learners was recognized. For example, the “virtual microscope” developed by The Open University, UK for disabled students, proved better able to achieve key learning objectives than its mainstream counterpart and it came to be used by all students.

## 2.4 Direct Access Versus Compatible Access

Solutions designed to make education accessible can be grouped into two categories: direct access and compatible access.

A **directly accessible** product allows a person with a disability to operate all on-screen controls and access all content without relying on the aid of an AT. For example, to be accessible to users with low-vision, directly accessible applications, software, or websites offer features that enlarge all controls and on-screen text. They are also designed using high contrast colors or provide features that allow users to choose appropriate colors.

To be accessible to users who are blind, a directly accessible product should have a keyboard interface with audio output. Such a keyboard interface can also provide access for users with physical disabilities. Audio output should announce the presence and status of all on-screen controls and convey the atmosphere of the application, software, or website. A built-in method of using a single key to scan through choices in the application or software will provide access for users who can only use a single switch as input. Teachers of students who are visually impaired report that their younger students receive only limited training with ATs. For this reason, it is particularly important to provide direct access in products targeted at elementary and middle school students.

Direct access brings many benefits. Most importantly, the user is able to access educational material without needing special assistive hardware and software. In this way, direct access helps to reduce cost for schools and individuals, and eliminates the technical difficulties associated with using AT. Direct access also gives students with disabilities the option to use any computer, freeing them from dependence on adapted workstations.

Ideally, the directly accessible interface should be designed by the same people who create the application, software, or website. These are the content experts and when they apply their understanding of educational goals to designing an accessible interface, the resulting educational experience will certainly be superior to the alternative; assistive technology paired with software not designed with users with disabilities in mind.

Alternatively, the **compatibly accessible** application, software, or website is an application designed with AT in mind. This level of access assumes that the user has a preferred AT package installed and is relatively competent and comfortable with the AT they are using. A compatibly accessible product is designed with “hooks” built into the software that facilitate the use of a screen reader, screen magnifier, or alternative input devices such as adapted keyboards or single switches. These “hooks” are provided by developers using software engineering tools such as Microsoft Active Accessibility (MSAA) and the Java Accessibility API from Sun Microsystems (discussed in more detail in Section 8.7 Accessibility Information for Operating Systems and Development Platforms). Developers of compatibly accessible applications should also remember to always expose the cursor, use standard controls and fonts, and follow the operating system’s human interface guidelines.

Compatible access offers some advantages. It provides consistency of operation between applications for users who already know how to navigate with their AT or who can become competent doing so. In some cases it may be less expensive to develop applications, software, and websites in this way. Relying on the AT package for text-to-speech capability, rather than adding it into the product itself, for instance, can save on disk space. In reality, compatibly accessible products may be the only means of access for some users, such as deaf-blind Braille users who depend on screen readers to interact with computers. Developers who are designing applications, software, and websites to be compatible with ATs should use proven programming techniques to create software that works consistently well with the range of screen readers, alternative input devices (e.g., switches, on-screen keyboards, voice recognition), and any other input or output device that is not part of a standard computer.

## 3. Principles for Accessibility in Online Distributed Learning

The following principles present best practices for producing accessible software applications and accessible content for online distributed learning. Developers, content providers, and educators involved in the creation of learning products should adhere to these guidelines from the outset of the design process, since retrofitting an inaccessible product is almost always significantly more difficult, labor-intensive, and expensive.

We offer six principles that address accessibility for people who have sensory or mobility disabilities. These principles also address accessibility issues faced by people with cognitive disabilities, though often to a lesser extent.

- 1) allow for customization based on user preference.
- 2) provide equivalent access to auditory and visual content based on user preference.
- 3) provide compatibility with assistive technologies and include complete keyboard access.
- 4) provide context and orientation information.
- 5) follow IMS specifications and other relevant specifications, standards, and/or guidelines.
- 6) consider the use of XML.

### 3.1 Allow for Customization Based on User Preference

When applications make it possible to present information in a versatile manner, content becomes more accessible, reaching a wider variety of users. Some examples of items that should be customizable by the users fall into two categories:

**Customizable display elements include:**

- font, font style, font color, and font size
- cursors size, style, and blink rate
- size of text and images, including video
- screen layout, colors, and backgrounds

**Customizable interface features include:**

- timing of events
- keyboard settings

Allowing the user to customize elements of the application and its content is essential for accessibility. Hard-coding presentation elements such as fonts, may make access impossible for some users. Depending on a user's abilities and preferences, they may seek to change settings for presentation style, size, or timing. For example, a user with low-vision may want to change the style of a font and enlarge the size of text.

The user should also be able to change other application features, such as the timing of events. For example, dialog boxes and alerts should remain on the screen until they are cleared by the user. Programmed activities that require several actions to be taken within a fixed time limit can create problems for users whose ATs provide less efficient ways of interacting with the software. Those timing requirements should be customizable by the user.

### 3.2 Provide Equivalent Access to Auditory and Visual Content Based on User Preference

To be fully accessible to deaf or hearing-impaired users, applications should provide equivalent access to all auditory aspects of learning technologies and content.



**To make applications accessible to those with hearing impairments, developers can:**

- caption all auditory content.
- provide a text transcription of auditory content.

**For blind or visually impaired users, applications should provide equivalent access to all visual aspects of learning technologies and content. Specifically, developers should:**

- add text descriptions (alternative text or alt-text) to all static images (e.g., pictures, logos, charts, links, other graphics) so the text can then be read by a screen reader or output to a Braille display.
- utilize the “longdesc” attribute for images that have useful content and require more lengthy descriptions.
- provide audio description tracks for multimedia, describing visual aspects of the content.

For users who are deaf, hearing-impaired, blind, visually impaired, or deaf-blind, applications should combine equivalent access, as detailed above, for all auditory and visual aspects of learning technologies and content. Deaf-blind users in particular, need text equivalents for all audio and visual material.

### 3.3 Provide Compatibility with Assistive Technologies and Complete Keyboard Access

Applications, software, and content must be compatible with all types of ATs including: screen readers, screen magnifiers, adaptive keyboards, voice recognition software, and single switches.

Developers should provide complete keyboard access to all elements of an application and its content, including menus, help directories, toolbars, and dialog boxes. Never assume that all users can operate a mouse.

### 3.4 Provide Context and Orientation Information

Applications and software are made more usable when developers provide context and orientation information to users. They should design their products to:

- teach users how to navigate.
- inform the user of the length of the document. For example, page numbers can be expressed as: “Page X of Y pages”.
- provide a way for users to skip standard page headers and navigation links. Users who are already familiar with the page layout should be able to skip directly to the primary content.
- maintain a consistent layout between pages. Content becomes more accessible when users can depend on consistency of presentation.
- provide alerts/text warning whenever a new browser window will be automatically opened.

Applications that implement these features will help users of screen reading software work more efficiently, saving time. They will be able to listen to content rather than rely on graphics or visual aids to navigate through content. In addition, these features benefit all users by increasing usability and minimizing the learning curve. And adhering to a consistent design between pages makes information generally easier to find for all users.

### 3.5 Follow IMS Specifications and Other Relevant Specifications, Standards, and/or Guidelines

Developers who follow relevant specifications and guidelines increase accessibility in two ways. Obviously, guidelines that provide information on how to implement accessibility within applications offer useful techniques and suggestions to programmers. But because accessibility often relies on interoperability *between* learning applications, software, content, and ATs, following other relevant industry specifications will also lead to improved access. These broad guidelines help ensure that applications, software, and content conform to standard operating system protocols, and thus make it more likely that ATs will be able to operate with them.



Developers within the online distributed learning community who seek to promote more fully interoperable technologies should fully adopt IMS Global Learning Consortium specifications.

Through the work of the IMS Accessibility Project Group, in conjunction with other IMS Project Groups, these specifications will continue to evolve and will include additional, specific sections intended to improve accessibility of online distributed learning technologies.

**Many IMS specifications are currently available:**

- The *IMS Profiles and Enterprise System Specification* organizes learner information and specifies interfaces for exchanging student, course, and administrative data. They make learning systems more responsive to the specific needs of each user, as well as better integrated with administrative software systems.
- The *IMS Content Packaging Specification* makes interchange of content possible between different learning systems.
- The *IMS Question & Test Interoperability Specification* allows institutions to share test items and other assessment tools across different systems.
- The *IMS Learning Resources Meta-Data Specification* creates a uniform way for describing learning resources.

**Several IMS specifications are currently in development (at time of publication of this document):**

- The *IMS Accessibility Project Group* is currently working on producing additions to the Learner Information Profile (LIP) Specification and including provisions for accessibility in the Meta-Data Specification.
- The *IMS Learning Design Project Group* is producing an information model that will extend IMS Content Packaging to describe a learning experience and the activities that contribute to it.
- The *IMS Digital Repositories Project Group* is producing an information model that specifies a means to perform searches, gather information, and set up alerts across different kinds of repositories.
- The *IMS Simple Sequencing Project Group* is working towards a final specification which provides a rule-based method for ordering and presenting learning objects to a user.

Throughout this document there are references to additional industry specifications, standards, and guidelines, which are also listed in Appendix B of this document.

**Resources:**

IMS specifications are available to the public, free of charge, at: <http://www.imsglobal.org/specifications.html>

## 3.6 Consider the Use of XML

XML (Extensible Mark-up Language) has been selected by the IMS Global Learning Consortium as the basis for all of its specifications. XML provides a widely accepted method for defining abstract data structures that can be easily parsed, used, and validated. All of the major computer platforms support XML, making it ideal for solving interoperability problems.

XML lends itself to accessibility for these and other related reasons. An AT often requires interoperability between the application delivering the actual course content or services and the specialized application, used by a learner with a disability, to access that content.

Materials authored in XML also offer the following benefits:

- **Transformable and flexible:** XML encourages separation of content and presentation. It allows user agents to transform the presentation of content to meet users' individual needs, even if the author didn't consider those transformations.
- **Structured and validated:** XML encourages use of consistent nested structures, which make it easier to navigate complex learning materials. Also, XML documents can be validated to ensure they use mark-up appropriately.
- **Text based:** XML documents are likely to contain reusable text content since they are created as text documents. Providing text improves accessibility as described in Section 5 Guidelines for Accessible Delivery of Text, Audio, Images, and Multimedia.

XML does not guarantee accessibility. To avoid creating accessibility problems, developers should follow guidelines accompanying any published XML language (see the WAI for information on W3C languages). Also, developers should use existing XML languages whenever possible, especially if they have built-in accessibility features, such as those from the W3C (see Section 4 for more details). When designing a new XML language, developers should follow the W3C's recommendations to ensure that the new language is as accessible as possible.

For more information, see the W3C's XML accessibility guidelines at: <http://www.w3.org/TR/xmlgl>

## 4. Using XML for Accessibility

As mentioned in Section 3, XML can facilitate accessibility through transformable, structured, text-based content. XML allows a range of flexible stylesheet transformations. It allows simple changes to font size and color as well as the use of complex translation grammars used to translate a presentation into entirely different modalities. Because all content in an XML document is declared and labeled, authors can create content that later can be re-styled in ways that the author never imagined.

### 4.1 XML is Customizable and Flexible

XML is the W3C recommended metalanguage for the Web. Using XML, authors can develop content that is controlled by languages of their choice and even of their own making. For the purposes of these Guidelines, authors are advised to use XML-based languages in preference to plain HTML. In addition, they should seek to confine their selection of special-purpose XML languages to those already in general use, especially those languages designed with accessibility provisions.

XML has significantly fewer limitations than HTML. While the tags and attributes within HTML are restricted to a pre-defined set, XML offers expert developers the capability to employ author-defined tags. This flexibility makes XML more fully customizable.

Headers and paragraph tags can be made more descriptive. Instead of the standard header and paragraph tags found in HTML, XML allows the developer to define tags, making them more intelligent and intelligible. Thus it is possible for an author to tailor data files to meaningfully describe the type of data in any particular set of tags in a data file.

This flexibility also allows the author to mark up data files with tags that clearly differentiate content intended for different users with different needs. For example, a single data file can contain content designed both for blind or visually impaired users and sighted users. Subsection 4.3.2 provides an example of how this can be achieved using XML and stylesheets. These flexible tags also support the editing process by permitting authors to use a different 'editing' view than they intend to display for their readers. A visually impaired author, for example, could use a large print view but publish in a smaller print.

Another important difference between HTML and XML lies in the way that the two languages handle the structure and display of data. HTML combines structure and display data. XML separates structure information from display information. The XML metalanguage can create languages that describe and manage the structure of data or that format data. The semantics of an XML document are, therefore, defined by the applications that process them and/or by stylesheets (Extensible Stylesheet Language Transformations, XSLT). Stylesheets render the content of the XML file appropriately for differing output formats.

### 4.2 XML Allows for Standardization for Accessibility, Validation, and Well-Formedness

XML-based mark-up languages, including those created by authors, depend upon online DTDs or schemas that provide rules for their interpretation. This means that the tags, including those in XHTML, must be well-formed and validated. This additional strictness, not required by HTML, imposes a discipline on content developers, but it is not burdensome. There are code validators available from W3C, and both the content files and the associated stylesheets can be validated for correct code. As described in the principles on standards, validity and well-formedness generally contribute to accessibility.

#### Resource:

For information on accessible use of XML, consult the resources from the W3C at: <http://www.w3.org/TR/xmlgl>

### 4.3 Recommended XML-Based Languages

The following subsection considers some of the XML-based languages used in online education.

### 4.3.1 XHTML

XHTML offers persuasive advantages over HTML for the production of accessible online media.

Well-formed HTML can be translated easily into XHTML, which then refers the user agent to standardized, online interpretation specifications. Developers need not fear that XHTML will cause compatibility problems with older browsers. XHTML will default to HTML when it encounters non-XHTML browsers. This back-up feature comes at a small cost to the author but it offers important support for user agents that are designed to increase accessibility using XML.

To use XHTML is to use a mark-up language that is future-proofed. Developers do not have to make significant changes to existing resources and activities; they need only follow the rules for producing correct HTML 4.1 and then add an extra line to their resource that identifies it as being accessible to XML-using agents. This added line also includes instructions for finding online, machine-readable information for interpreting the XML version in use. XHTML is particularly useful to user agents that have work to do: to find alternative resources, to synchronize files such as caption and video files, or to transform the content in some way to make it more accessible.

### 4.3.2 XML Mark-up and Stylesheets

Cascading Stylesheets (CSS), as used with HTML, are able to alter the display and formatting of all or part of the contents of the HTML file to which they apply. Thus, for example, <H1> tags in the HTML file can be designed to appear as bold, 15 point, and blue. But a different CSS can instead render them as 30 point for a low-vision learner. CSS are, however, capable of altering the formatting and display of selected aspects of the entire contents of the HTML file. The browser will display the whole HTML file to every user.

Extensible Stylesheet Language Transformations (XSLT) are a more powerful tool. They allow the selection and formatting of particular parts of an XML file. They also allow the addition (within the XSLT file itself) of other elements such as images, text, or video not included in the default presentation. Thus, authors need only write data files once but can specify a variety of presentations to suit different needs. For example, using HTML to create multi-language websites requires separate files for each language for each page. XSLTs allow the stylesheet to select each language's text from the contents of a single file for the page and display it together with any language-specific images, audio, or video.

Besides handling multiple languages, XML can be used to differentiate content that is intended for a variety of users including blind learners, visually impaired learners, and sighted learners, all within a single data file.

Here is an example from a biological sciences course:

```
<A_robustus_intro>
  <audio description> This page contains information relating to the fossils of
  Australopithecus robustus. It shows an image of a skull specimen discovered in 1950
  by Robert Broom showing a slight sagittal crest and large zygomatic arches that
  project forwards, hiding the sunken nasal area.</audio description>
  <large print> The above image shows the skull of Australopithecus robustus,
  discovered in 1950 by Robert Broom. Look at the distinctive features and note them.
  </large print>
  <standard> The above image shows the skull of Australopithecus robustus, discovered
  in 1950 by Robert Broom. Look at the distinctive features and note them. </standard>
</A_robustus_intro>
```

The various versions about *A. Robustus* can be added to the learner's view of the lesson via XSLT, perhaps in response to a preference setting where the user can indicate the need for large print and/or audio description. This brief example illustrates the capacity of XML to include author-tailored tags that increase both the flexibility and readability of the file.

### 4.3.3 Scalable Vector Graphics (SVG)

Scalable Vector Graphics language (SVG) is a vector graphics language recommended and supported by W3C, and is thus used by many of the W3C member companies who develop relevant authoring and user agent software.

Vector graphics differ from raster graphics. Raster, or bit-mapped graphics store images as a simple collection of bits, making manipulation of those images difficult. Vector graphics, however, define images descriptively. For example, a vector image of a cube is defined by a mathematical description of its shape. In that way, the cube can be enlarged, reduced, or rotated on the screen. The description of the object can be interpreted by other output devices, such as a haptic display. In the case of technical drawings, the displayed image is often derived from a database of information. That information can be reproduced graphically, but also as a table or even represented by sound.

The data that defines a vector image can also include a detailed description of the data itself. Thus both the ALT text and the LONGDESC can be part of the ‘image’ object. In addition, SVG supports meta-data about the object so that an image object can point to an alternative description of the object located elsewhere.

Anywhere images are constructed from other image elements, the individual objects retain their own descriptions. In an engineering diagram, for example, items included in a flowchart can each be described separately. In this way, the content producer is able to group sets of objects to create a more complex graphical object. If the graphical elements can be described and their relationship described, then even the user who cannot use a graphical interface still has a good chance of being able to decode the image.

Authors need to work carefully with the power of SVG. Complex images can contain many sub-elements and thus accumulate a large number of descriptions associated with the entire image. Authors should organize sub-element information hierarchically so that a user who cannot see the entire image can listen to or read the text, and still be able to create a suitable mental model of the image.

SVG provides a format for images that supports a number of accessibility features. These features include:

- text information for blind or low-vision users.
- scalability that allows high quality zoom functionality. Lines stay sharp when magnified.
- labels on objects. If the objects are also grouped into a hierarchy, the hierarchical relationship between labels will convey more fine-grained detail than a single text description.
- support for stylesheets, DOM2 for customizing presentation and meta-data as well as RDF for efficient cataloguing.
- support for device-independent events allowing ATs and other input devices.

**Resources:**

Further information about SVG and SVG accessibility is available at: <http://www.w3.org/TR/SVG-access/>

Also see the W3C’s “Accessibility Features of SVG” at: <http://www.w3.org/TR/SVG/struct.html#GEElement>

#### 4.3.4 Synchronized Multimedia Integration Language (SMIL)

Synchronized Multimedia Integration Language (SMIL) is another XML language. SMIL, pronounced “smile”, is recommended for the creation of multimedia presentations for the Web and is supported by popular user agents, including RealPlayer and QuickTime Media Player.

Many of the solutions to problems discussed in these Guidelines require the author to provide equivalent alternative formats such as text, sounds, or haptics alongside graphics. The integration and synchronization of these elements can be managed by SMIL. Throughout the text of these guidelines, there is reference to SMIL and there are some examples that show how it can be used, where appropriate.

**Resources:**

For information on accessibility features of SMIL see the W3C’s “Accessibility Features of SMIL” at: <http://www.w3.org/TR/SMIL-access/>

For an authoring tool for accessible SMIL presentations see the National Center of Accessible Media’s MAGpie (Media Access Generator) at: <http://ncam.wgbh.org/webaccess/magpie/>

### 4.3.5 XML E-Book Formats

Two major e-book formats are XML based: the Digital Audio-based Information System (DAISY) and the Open eBook Publication Structure.

DAISY's mission is to develop international standard and implementation strategies for the production, exchange, and use of Digital Talking-Books in both developed and developing countries. To ensure access to information for people with print disabilities, DAISY is designed to enhance integration with mainstream technology.

More information is available on the DAISY website at: <http://www.daisy.org>

The purpose of the Open eBook Publication Structure is to provide a specification for representing the content of electronic books. Specifically:

- the specification is intended to give content providers (e.g., publishers and others who have content to be displayed) and tool providers minimal and common guidelines that ensure fidelity, accuracy, accessibility, and presentation of electronic content across various electronic book platforms.
- the specification seeks to reflect established content format standards.
- the goal of this specification is to provide the purveyors of electronic-book content (publishers, agents, authors, et al.) a format for use in providing content to multiple reading systems.

The Open eBook Publication Structure is based on the premise that in order for electronic-book technology to achieve widespread success in the marketplace, reading systems must have convenient access to a large number of titles across a variety of genres. The specification therefore is based on HTML and XML, the same core languages that define the World Wide Web, and is designed to allow publishers and authors to deliver their material in a single format. This information and more is available from the Open eBook site at: <http://www.openebook.org/>

### 4.3.6 Other Mark-up Languages

MathML, ChemML, MusicML, and others are discussed in Section 11 Guidelines for Topic Specific Access. These are mark-up languages, developed in XML format, and they can be mixed and matched with XHTML.

## 5. Guidelines for Accessible Delivery of Text, Audio, Images, and Multimedia

Every person learns differently. At its best, online learning allows users to interact with lesson material in their preferred way, relying on their individual strengths while discounting as much as possible their weaknesses. The principles of excellent software design call on developers to work in full knowledge of the range of human skills and limitations. Software designers of teaching materials and activities, in particular, must strive to achieve this high standard.

When a user has a disability, access to learning software may depend entirely on how flexibly that product can deliver its content. Some users may need only to modify the parameters in which media is presented; other users may require entirely different media. Developers who achieve the kind of flexibility that diversity requires will enhance the accessibility of their product.

At a minimum, developers should provide text representations for all media types. This baseline will help address access for many users. It is important to note that users with learning disabilities benefit from graphical presentations. For this reason, the practice of providing text-only content as an alternative to inaccessible multimedia content may not be an effective solution for users with cognitive disabilities.

A number of resources that address flexible media delivery are currently available. The W3C's Web Accessibility Initiative provides accessibility guidelines for W3C technologies such as HTML, XML, SMIL, CSS, and SVG. It also provides more general guidelines for Web content accessibility, authoring tool accessibility, and user agent accessibility. More information links are available in the appendix. Two other comprehensive guides are referenced here.

The National Center for the Dissemination of Disability Research publication, "User-Friendly Materials and Alternate Formats" provides information on implementing several formats and modes including:

- large print
- audiotape
- Braille
- diskette
- CD-ROM/DVD
- video and descriptive video
- captions
- Internet/World Wide Web

Visit the National Center for the Dissemination of Disability Research online at:

<http://www.ncddr.org/du/products/ufm/ufm.html>

The California Community Colleges have produced a comprehensive set of guidelines that offers alternative formats for presenting printed materials. Their document, "Guidelines for Producing Instructional and Other Printed Materials in Alternate Media for Persons with Disabilities" can be found at:

<http://www.htctu.fhda.edu/amguidelines/am33000.htm>

The California Community Colleges document provides a comprehensive resource for producing several types of alternative media including:

- audio
- Braille
- tactile graphics
- large print
- electronic text



## 5.1 Common Types of Media Delivery and Associated Presentation Formats

### 5.1.1 Text

When text is correctly structured and formatted, it can be the most flexible way to present content. To make distributed online learning accessible, developers of learning platforms must provide a means to render digital text in alternative formats.

Specifically, it should be possible to render text as:

- **Visual information.** Text can be displayed on computer screens or other electronic devices (e.g., personal digital assistants, cell phones, e-book readers).
- **Audio information.** Text can be translated into speech using recordings or via synthesized speech provided by a computer.
- **Tactile information.** Text can be displayed on refreshable Braille displays or printed using a Braille embosser.

#### **Common text accessibility problems include:**

- hard-coded fonts that prevent users from changing style, size, or color.
- text presented with background images or poor contrast colors that hinder readability.
- text presented in an image format that screen readers and Braille displays cannot transform.
- multi-column formats (including some tables) that screen readers cannot process in the correct order.

#### **Learning system developers may enhance the accessibility of text for all users by following these practices:**

- offer features that allow the user to customize fonts and backgrounds.
- allow ATs to have access to the source code.
- use validated XHTML.

#### **Content creators or educators may enhance the accessibility of text for all users by following these practices:**

- choose text formats that offer the most accessibility (e.g., XHTML, plain text).
- use true text and not graphical representations of text.
- structure the text appropriately, identifying headings and other structural elements.
- use styles or stylesheets to provide a flexible display.

### 5.1.2 Audio

Audio elements can add to the general appeal of online learning materials while making them more accessible to those who are print-impaired learners, such as those with visual impairments or dyslexia. However, developers should provide alternatives to ensure that learners who are deaf or hard-of-hearing are not disadvantaged.

#### **Common audio accessibility problems include:**

- lack of captions and/or transcripts.
- poor sound quality.
- inability to control volume.

#### **Learning system developers may enhance the accessibility of audio for all users by following these practices:**

- provide a means to include captions and/or transcripts.
- provide volume controls.
- provide visual equivalents to audio alerts (e.g., show a text alert on the screen whenever an error beep is played).

#### **Content creators or educators may enhance the accessibility of audio for all users by following these practices:**

- provide transcripts or captions for all essential audio.
- consider providing other forms such as ASL or captions with images.



**Resources:**

An authoring tool to add captions, subtitles, and audio descriptions to digital media is available free-of-charge for content creators and educators. This tool, MAGpie (Media Access Generator), is available from the National Center for Accessible Media (NCAM) at: <http://ncam.wgbh.org/webaccess/magpie/>

CAST (the Center for Applied Special Technology) offers “Tips on Presenting Alternatives to Sound” at: <http://www.cast.org/udl/index.cfm?i=352#tips>

Proprietary software exists that allows authors to add sign language displayed using animated characters. One company developing this software is Vcom3D. They have created “SigningAvatars” that are able to render content in Signed English and Pidgin Signed English (and possibly ASL in the future). More information is available at: <http://www.signingavatar.com>

**5.1.3 Images**

Images can provide essential information. But without text support, images are not accessible for users who are blind or have low-vision. Developers must provide users with a way to access visual information. Providing text identification, or alternative text, will also benefit users of text-only browsers, such as mobile phones. Developers should ensure that images are scalable so that users can enlarge them for better clarity.

**Common image accessibility problems include:**

- failure to provide alternate text.
- poor image resolution that restricts the ability of low-vision users to enlarge images.

**Learning system developers may enhance the accessibility of images for all users by following these practices:**

- provide a means to include text alternatives of images.
- provide a zoom feature.

**Content creators or educators may enhance the accessibility of images for all users by following these practices:**

- provide text alternatives for images.
- use SVG features that improve accessibility (see subsection 4.3.3).
- use the highest practical resolution for bit-mapped graphics when use of SVG is not possible (e.g., photographs).

**Resources:**

The University of Toronto’s Special Needs Opportunity Windows (SNOW) on making images accessible at: <http://snow.utoronto.ca/access/higher/images.html>

The W3C WAI Techniques for Web Content Accessibility Guidelines 1.0 section 2.1.1 for providing alternatives to auditory and visual content at: <http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#images>

The Web Design Group’s article on use of alt-text in HTML at: <http://www.htmlhelp.com/feature/art3.htm>

WebAIM’s tutorial, including code examples, illustrating the W3C Web Content Accessibility Guidelines 1.0 recommendation checkpoint for providing “equivalent alternatives to auditory and visual content” at: <http://www.webaim.org/tutorials/alt>

SVG is a language used in XML to describe two-dimensional graphics. SVG 1.0 is recommended by W3C and is available from the W3C website at: <http://www.w3.org/TR/SVG/>

Accessibility features of SVG are described at: <http://www.w3.org/TR/SVG-access/>

**5.1.4 Multimedia**

Multimedia is the combination of text, graphics, video, animation, and sound. Thus, a given piece of multimedia content combines the access needs of each media type represented. Multimedia can be useful for many groups of learners, since a multi-modal presentation of information can be easier to understand. In general, users benefit when alternatives are available for each media type.

**Common multimedia accessibility problems include:**

- digital videos without captions, transcripts, or audio descriptions.

**Learning system developers may enhance the accessibility of multimedia for all users by following these practices:**

- provide the means to include accessibility features, such as captions, transcripts, and audio descriptions, within the multimedia format provided by the application.

**Content creators or educators may enhance the accessibility of multimedia for all users by following these practices:**

- follow all relevant suggestions for enhancing accessibility for text, audio, and images, since multimedia can combine all of these elements.
- provide audio description describing essential visual elements for video content.
- consider the importance of the timing of media delivery when planning access features. For example, a “talking head” video may need only a stand-alone transcript of the audio, but a documentary including graphics and other important visuals may require captions in order to maintain the link between visuals and narration.

**Resources:**

The WGBH National Center for Accessible Media’s “Rich Media Accessibility” website (<http://ncam.wgbh.org/richmedia/>) provides information of a wide range of digital media formats, including:

- A tutorial on Creating Audio Descriptions for Rich Media at: <http://ncam.wgbh.org/richmedia/audiodesc.html>
- A tutorial on Creating Captions for Rich Media at: <http://ncam.wgbh.org/richmedia/captioning.html>

## 6. Guidelines for Developing Accessible Asynchronous Communication and Collaboration Tools

This section covers the development of asynchronous communication and collaboration tools, including:

- threaded message boards
- e-mail and listserv messaging
- document repositories
- organizers, schedulers, and calendars
- presentation tools

Asynchronous communication and collaboration tools such as threaded message boards, e-mail messaging, listservs, document repositories, calendar systems, and presentation tools must be rendered in formats that facilitate the full participation of learners with disabilities in online interactions. The navigation system for these utilities should allow users of ATs to operate without encountering barriers to any aspect of the functionality. Specifically, the software should allow them to:

- scan items and locate content easily.
- follow the thread of an ongoing discussion topic.
- post responses, or add additional information.

Users must have flexibility in their choice of browser, input and output modalities, and be able to use ATs. All Web-based components of asynchronous communication and collaboration tools should be compliant with the W3C Web Content Accessibility Guidelines 1.0, available from the WAI website: <http://www.w3.org/TR/WCAG10/>

### 6.1 Threaded Message Boards

For discussion forums, bulletin boards, and other utilities that allow posting of text messages, content providers should render both the navigation system and the content of the messages in standard XHTML. Requiring use of proprietary client software or non-standard file formats may create barriers by preventing users from establishing individual preferences, as described in the introduction.

#### **Common threaded message board accessibility problems include:**

- navigation systems using complex framesets that fail to include title or name attributes.
- lack of ALT text on buttons that function with JavaScript to expand or contract threaded discussions (i.e., blue triangles are commonly used).
- form fields that do not accommodate keyboard navigation.
- illogical tab sequence when moving between elements on the page.

#### **Learning system developers may enhance the accessibility of threaded message board applications for all users by following these practices:**

- ensure that all actions can be completed using the keyboard.
- provide easy-to-use features that allow users to configure the interface according to individual preferences.
- provide help files, including an orientation to the interface and its functionality.
- use text instead of images of text for navigation links or buttons.
- use stylesheet attributes and text to create attractive form buttons instead of using images.
- provide bypass links or some other method to allow users to skip repetitive navigation links and go directly to the main content of the page.

**Content creators or educators may enhance the accessibility of threaded message board applications for all users by following these practices:**

- provide informative names to identify discussion threads and message subject lines.

**Resource:**

Examples of accessible threaded message boards are available on the Internet Accessibility Discussion at: <http://www.worldenable.net/iadiscuss/>

## 6.2 E-Mail Messaging

E-mail systems may be used either for one-to-one exchanges, or used to distribute and receive messages to a larger user group via a listserv. To use an e-mail system, users select proprietary client software according to their individual needs and according to the compatibility needs of their preferred AT hardware and software. To ensure accessibility, developers should make the interface of their e-mail clients compliant with the latest Authoring Tool Accessibility Guidelines, available from the W3C's WAI website at: <http://www.w3.org/TR/WAI-AUTOOLS/>

To further ensure that e-mail messages are accessible, users must always have the option to send plain ASCII text.

**Common e-mail accessibility problems include:**

- non-plain text components within the body of the e-mail, including XHTML mark-up code for fonts, colors, and bold formatting.
- inclusion of background images that obscure text for users who are low-vision.
- attachment of non-standard file formats that cannot be accessed by the recipient of the e-mail.

**Learning system developers may enhance the accessibility of e-mail applications for all users by following these practices:**

- ensure that e-mail clients allow e-mail content in a plain text format that does not allow fonts, colors, bold formatting, background images, and so on.
- provide easy-to-use features that allow the users to configure the interface according to their individual preferences (e.g., a simple or expert option).
- provide help files that include an orientation to the interface and its functionality.
- follow the latest version of the WAI Authoring Tool Accessibility Guidelines.

**Content creators or educators may enhance the accessibility of e-mail applications for all users by following these practices:**

- ensure that the subject line accurately reflects the content of the message.
- do not attach files to listserv messages. Some recipients may not be able to access the content.
- use a signature file or Vcard to provide information about the sender's name, title, and address.

## 6.3 Document Repositories

Designers of document repositories must ensure that the indexing system is accessible to all users and that it follows logical document structure. Where possible, store documents in standard XHTML format. Using proprietary or non-standard file formats may create barriers. Other formats are less accessible and might limit the user's ability to access information according to individual needs or preferences. Search functions and display options must be accessible to users of ATs.

**Common document repository accessibility problems include:**

- indexing or navigation systems use complex frames that do not include the title or name attributes.
- form fields in search utilities do not accommodate keyboard navigation.
- display options are hard to locate or are not accessible via the keyboard.

**Learning system developers may enhance the accessibility of document repository applications for all users by following these practices:**

- ensure that all actions can be completed from the keyboard.
- provide easy-to-use features that allow users to configure the interface according to individual preferences.
- provide help files and include an orientation to the interface and its functionality.

**Content creators or educators may enhance the accessibility of document repository applications for all users by following these practices:**

- ensure that document titles accurately reflect the content of the document.
- provide documents in standard XHTML format. Avoid the use of proprietary or non-standard file formats.

**Resource:**

Examples of accessible document repositories are available at WAI Mail Archives at:

<http://lists.w3.org/Archives/Public/w3c-wai-ig/>

## 6.4 Organizers, Schedulers, and Calendars

Some applications, such as organizers, schedulers, and calendars allow users to post shared information. To make these utilities accessible, developers and content providers should seek to structure posted information in a logical manner, encoded using standard XHTML. Non-visual users are at a disadvantage when confronting complex tables or scripting languages. For these and other users, information must linearize correctly in order to be translated accurately into other forms, such as audio output. When providers fail to properly format structured data, dates and events can become improperly juxtaposed. Traditional calendar layouts are particularly susceptible to this sort of error, so content providers ought to include alternative linear output whenever possible.

**Common organizer, schedule, and calendar accessibility problems include:**

- scripts that use event handlers that are device-dependent (i.e., require a mouse).
- the use of incorrect mark-up that inhibits the user's ability to navigate among table cells and access headers and other cell information.
- table cells whose content does not linearize correctly.

**Learning system developers may enhance the accessibility of organizers, schedules, and calendars for all users by following these practices:**

- ensure that tools are usable even when scripts are not supported.
- provide equivalent information on an alternative accessible page.
- for data tables, identify row and column headers. Use abbreviations for header labels and identify structural groups of rows or columns.
- test table content to ensure it linearizes correctly. If it does not, provide a linear alternative.

**Content creators or educators may enhance the accessibility of organizers, schedules, and calendars for all users by following these practices:**

- provide informative names for events or other posted items.

**Resource:**

The Wave is an accessibility evaluation tool that analyzes the linearized reading order. It is available at the Web address: [http://www.temple.edu/inst\\_disabilities/piat/wave/](http://www.temple.edu/inst_disabilities/piat/wave/)

## 6.5 Presentation Tools

### 6.5.1 Microsoft PowerPoint

Microsoft PowerPoint has been widely adopted because it is easy to use. With a little practice, most authors can easily create presentations that look “professional”. One of its strengths is the fact that Microsoft PowerPoint allows users to mix and match, to drag and drop all sorts of objects into their presentations. The resulting collection of elements often contain no reference to their origins, how they came to be included, what relationship they have to each other and so on. That is a problem for those who require accessibility. It is certainly possible that a presentation slide contains only a set of text outline statements that are easily accessed by a person with vision impairments. But in fact it’s all too likely that a given slide will include many inaccessible objects.

This same problem makes converting Microsoft PowerPoint slides to Web pages difficult. If the screen presentation of the Microsoft PowerPoint display is simply converted into a single graphic image and is then connected by links from one screen to another, all embedded information is lost. Often it is just this information that those with disabilities depend on.

For these reasons, the gains made by presenters using Microsoft PowerPoint that allow them to present their material visually often come at a cost to those with special needs or who rely on alternative access technologies.

Below are three possible alternative approaches. Accessibility Wizard, which provides a way to convert Microsoft PowerPoint slides into accessible HTML. The second, W3C Slidemaker, which converts HTML into slides, and finally WimpyPoint, a tool that replaces Microsoft PowerPoint.

### 6.5.2 Alternatives to Microsoft PowerPoint for Accessibility

#### 6.5.2.1 Accessibility Wizard (for Microsoft PowerPoint)

Accessibility Wizard software was created at the University of Illinois. It provides an alternative to Microsoft PowerPoint’s built-in Web Publishing feature.

Using Microsoft PowerPoint’s Web Publishing feature creates content that can only be viewed using Microsoft’s Internet Explorer. Even if non-XML options are selected, users cannot easily add information that is required for accessibility. Accessibility Wizard simplifies the task of converting Microsoft PowerPoint presentations into text pure HTML.

Accessibility Wizard is available free-of-charge from The University of Illinois at: <http://www.rehab.uiuc.edu/ppt/>

#### 6.5.2.2 W3C Slidemaker

Slidemaker converts a single long HTML or XHTML page into a set of slides. It makes it easy for the author to separate text from presentation, which it handles using stylesheets. Slidemaker’s ease of use also encourages good use of meta-data and helps presenters ensure that their presentation will be accessible to alternative devices. For more information visit the slidemaker website at: <http://dev.w3.org/cvsweb/slidemaker/>

#### 6.5.2.3 WimpyPoint

ArsDigita has created an open source alternative to Microsoft PowerPoint called WimpyPoint.

WimpyPoint replaces Microsoft PowerPoint and allows users to build a slide presentation using their preferred browser. Presentations are stored remotely and can be accessed from anywhere. WimpyPoint allows multiple users, even when separated geographically, to share presentations and to collaborate in their creation.

WimpyPoint is ideal for creating accessible presentations since it is based on HTML and supports cascading stylesheets.

WimpyPoint is available at the ArsDigita website at:

[http://www.arsdigita.com/acs-repository/one-version?version\\_id=1114](http://www.arsdigita.com/acs-repository/one-version?version_id=1114)

## 7. Guidelines for Developing Accessible Synchronous Communication and Collaboration Tools

This section covers guidelines for the development of synchronous communication and collaboration tools, including:

- synchronous text chat
- audio-conferencing
- video-conferencing
- whiteboards
- Multi-user domain Object Oriented environments (MOOS)

Synchronous communication and collaboration tools, such as synchronous text chat, audio-conferencing, video-conferencing, and whiteboards, are increasingly important components of online learning. In order to ensure the full participation of learners with disabilities, these synchronous tools must be made accessible to them. In general, this means ensuring that the user interface of the tool, as well as the real-time communication managed by the tool, are both accessible, for input and output equally.

### 7.1 Synchronous Text Chat

Synchronous text chat tools (such as Internet relay chat (IRC), ICQ, and others) allow two or more users to communicate via typed text messages in real-time. For a chat tool to be accessible, both the navigation system and the messages themselves must be accessible.

**Common synchronous text chat accessibility problems include:**

- complex and non-standard user interfaces.
- some exclusively mouse-driven functions.
- need for simultaneous user attention to multiple areas (i.e., message composition and message monitoring).
- fast pace of conversation may limit access by users who communicate slowly.

**Learning system developers may enhance the accessibility of synchronous text chat applications for all users by following these practices:**

- simplify the user interface and follow appropriate platform-specific user interface guidelines.
- provide help files, including an accessible orientation to the interface and its functionality.
- ensure that all mouse actions can also be completed, effectively, from the keyboard.
- provide keyboard accessible mechanisms to enable the user to quickly shift the focus between the message composition and the message monitoring areas.
- provide the choice not to automatically move focus to new messages as they arrive.
- provide an option to manually refresh the list of messages.
- provide mechanisms that allow users who communicate slowly to participate effectively (e.g., speaker designation).

### 7.2 Audio-Conferencing

Audio-conferencing (via phone or Internet) allows two or more users to collaborate via real-time speech. In order to have access, users with hearing and speech disabilities must be accommodated by the system.

**Common audio-conferencing accessibility problems include:**

- audio output that is inaccessible to users who are deaf or who have profound hearing loss.
- speech input that may be difficult for users with speech disabilities.

- some exclusively mouse-driven functions.
- fast pace of conversation that may limit access by users who communicate slowly.

**Learning system developers may enhance the accessibility audio-conferencing applications for all users by following these practices:**

- ensure that a real-time text transcript is available to participants. The transcript can be provided by any one of a number of methods (e.g., generated by effective minute taking, telecom relay service, or reliable voice recognition).
- ensure a real-time text-to-speech capability is available to participants via telecom relay service or automated text-to-speech.
- ensure that all mouse actions can also be completed, effectively, from the keyboard.
- provide mechanisms that allow users who communicate slowly to participate effectively (e.g., speaker designation).

## 7.3 Video-Conferencing

Video-conferencing allows two or more users to collaborate via real-time interaction that includes both video and sound. In order to be accessible, users with visual, hearing, and/or speech disabilities must be accommodated.

**Common video-conferencing accessibility problems include:**

- additional visual information such as graphs and charts that may be inaccessible to users who are visually impaired.
- audio output that is inaccessible to users who are deaf or who have profound hearing loss.
- speech input requirements that may be difficult for users with speech disabilities.
- videoconference systems that preferentially encode stationary elements of a scene, rather than dynamic ones. These automated choices can disrupt effective transmission of sign languages, since they depend on movement.
- some exclusively mouse-driven functions.
- fast pace of conversation that may limit access by users who communicate slowly.

**Learning system developers may enhance the accessibility of video conferencing applications for all users by following these practices:**

- provide a mechanism for describing visual support elements and encourage the use of this function.
- ensure that a real-time text transcript can be made available to participants. These transcripts can be generated by effective minute taking, telecom relay service, or reliable voice recognition. Once generated, this text transcript might be displayed by a sign language “avatar” (computer-generated signer). Real-time text transcription services can be accessed remotely.
- enable the addition of sign translation services provided remotely. The tool should be designed to accept a second video stream arriving from another location.
- ensure that a real-time text-to-speech capability is available to participants. Transcripts can be generated by a telecom relay service or automated text-to-speech software.
- investigate video encoding standards that facilitate sign language transmission, such as MPEG-4.
- ensure that all mouse actions can also be completed effectively from the keyboard.
- provide mechanisms that allow users who communicate slowly to participate effectively (e.g., speaker designation).



## 7.4 Whiteboards

Whiteboards are the graphical equivalent of synchronous chat tools. They allow multiple users to work on collaborative drawings in real-time. They often include functionality for drawing, painting, and importing existing graphic files. In order for the system to be accessible, both the navigation system and the content of the workspace must be accessible.

**Common whiteboard accessibility problems include:**

- exclusively graphical workspace that is inaccessible to people with visual disabilities.
- when text tools are available, they often produce text in such a way that it is not accessible by screen readers.
- many exclusively mouse-driven functions.

**Learning system developers may enhance the accessibility of whiteboard applications for all users by following these practices:**

- integrate an accessible synchronous text chat with the whiteboard and encourage users to use this function to describe their graphical work.
- ensure that all mouse actions can also be completed effectively from the keyboard.
- implement the workspace using SVG in order to take advantage of the access features of this mark-up language (see subsection 4.3.3 on Scalable Vector Graphics).

**The access features of the SVG language include:**

- SVG images are represented as objects, so lines stay sharp when magnified.
- SVG objects may be labeled. Also objects can be grouped hierarchically. The hierarchical structure of these text equivalents works better than a single text description to convey fine-grained detail.
- SVG supports stylesheets, DOM2 for customizing presentation and meta-data and RDF for efficient cataloging.
- SVG supports device-independent events so operation is not limited to the mouse.

## 7.5 Multi-User Domain Object Oriented Environments (MOOs)

MOOs are multi-user virtual environments in which users often control “avatars” (computer-generated actors) that move through the virtual world, interacting and communicating via speech generated from user-typed instructions. In order to be accessible, both the environment navigation system and the communicative exchanges between avatars must be accessible.

**Common MOO accessibility problems include:**

- visual information in the MOO, including the appearance of the physical environment and the other avatars, cannot be accessed by users who are blind.
- audio output is inaccessible to users who are deaf or who have profound hearing loss.
- some exclusively mouse-driven functions.
- complex spatial navigation of the MOO.
- fast pace of conversation may limit access by users who communicate slowly.

**Learning system developers may enhance the accessibility of MOOs for all users by following these practices:**

- provide a mechanism for automated text description of the virtual world.
- ensure a real-time text transcript can be made available to participants. This text transcript might be translated into Braille or even (from the listener’s perspective only) signed by a speaking avatar.
- ensure that all mouse actions can also be completed effectively from the keyboard.
- include powerful navigation tools for accomplishing high-level navigational instructions (e.g., “take me to the library”).

- provide mechanisms that allow users who communicate slowly to participate effectively (e.g., speaker designation).

**Resources:**

- Center for Applied Special Technologies: <http://www.cast.org>
- Equal Access to Software and Information: <http://www.rit.edu/~easi/>
- Special Needs Opportunity Windows: <http://snow.utoronto.ca>
- University of Washington DO-IT Program: <http://www.washington.edu/doi/>
- The CPB/WGBH National Center for Accessible Media: <http://ncam.wgbh.org/>

## 8. Guidelines for Developing Accessible Interfaces and Interactive Environments

Interactivity in online learning is the real power behind the medium. Compared to a passive text, a well-designed and highly interactive online lesson offers clear advantages in flexible content delivery as well as assessment.

Developers who create accessible interfaces, simulation, and interactive exercises for mainstream computer environments face considerable challenges in making their applications accessible.

The following section offers accessibility guidelines for developers of applications delivered via the Web, as well as on platforms for which AT is not available, such as DVDs and kiosks.

Section 3 Principles for Accessibility in Online Distributed Learning, presents some general principles to guide developers when creating accessible e-learning software, applications, and content.

Section 5 Guidelines for Accessible Delivery of Text, Audio, Images, and Multimedia, presents guidelines for creating accessible multimedia presentations.

This section provides additional information beyond that supplied in Section 3. However, it does not address specific accessibility issues of operating systems or programming languages. Subsection 8.7 lists resources that support specific development environments.

A comprehensive guide to application software accessibility is available from the Trace Center at: [http://trace.wisc.edu/docs/software\\_guidelines/toc.htm](http://trace.wisc.edu/docs/software_guidelines/toc.htm)

### 8.1 Interface Controls

When designing an interface, developers should consider compatibility. It is important to understand how people with disabilities actually use their ATs and how those technologies integrate with software and underlying operating systems. Course designers, educators, and students all need to use the interface of a learning application, including buttons, text fields, text field labels, menus, and other components.

**Common interface control accessibility problems include:**

- controls that are labeled with images rather than text.
- controls that require the use of a mouse.
- display options that are not easily located or are not accessible using keyboard navigation.

**Learning system developers may enhance the accessibility of interface controls for all users by following these practices:**

- follow accessibility guidelines for the operating system or development platform in use.
- use the standard components available for the operating system or development platform, or follow accessibility guidelines on how to create custom controls.
- ensure that all actions can be completed from the keyboard.
- provide easy-to-use features that allow configuration of the interface according to the user's preferences.
- provide help files, including an orientation to the interface and its functionality.
- test interfaces using actual ATs.

### 8.2 Navigating the Interface

Many users of ATs encounter difficulty trying to use features normally accessed only by a mouse. Making the interface overly mouse dependant will surely cause difficulties for AT users, especially when navigating through content and other elements such as menu bars, tables of contents, and frames.

**Common interface navigation accessibility problems include:**

- indexing or navigation systems that use complex frames where the title or name attribute is absent.
- tables of contents with expand/collapse features (e.g., blue triangles, plus-minus signs) that lack text labels.
- menu bars built using scripting languages that ATs cannot understand.

**Learning system developers may enhance the accessibility of interface navigation for all users by following these practices:**

- provide names, titles, or text labels for each element of the interface.
- ensure that the keyboard can access all parts of the interface. Clearly document all appropriate keystrokes.

## 8.3 Forms

Forms must follow interface control guidelines in general. But when a large number of interface elements are displayed together in a single form, problems can arise that require other solutions.

**Common form accessibility problems include:**

- illogical tab order among controls.
- complex layout of controls that makes it difficult to determine which label matches which control, or to determine how a series of related controls are connected.
- form fields in search utilities that do not accommodate keyboard navigation.
- visual-only modifications to form fields that users must perceive in order to proceed. Principle among these modifications are color-coded areas or areas flagged with an image meant to signal that a given field must be filled in or that an error has been entered.

**Learning system developers may enhance the accessibility of forms for all users by following these practices:**

- ensure that the tab order makes sense.
- use programmatic means, when available in the development environment, to indicate which label applies to which field.
- ensure that all actions can be completed from the keyboard.
- do not rely on color alone to differentiate information. For example, if the interface uses red text to indicate that a field is required, also include the word “required” next to those items.
- provide a means for users to easily locate and correct form entries that have errors.

**Content creators or educators may enhance the accessibility of forms for all users by following these practices:**

- use clear labels for each form item.
- place a default value in edit boxes or on the top line of drop-down lists.

## 8.4 Interactive Exercises: Drag-and-Drop Exercises, Simulations, and Timed Tests

When designing complex interactive activities, developers should take care that interfaces remain independent of the input and output requirements of different users. For example, drag-and-drop activities should be usable with either mouse or keyboard, and keyboard commands should be made as easy to execute as possible.

Also consider the manner in which information is displayed. Simulations can be made accessible by allowing for multi-modal output. For example, a computer-simulated chemistry experiment shows its results by changing the color of the liquid in an image from colorless to blue, be sure that the same information is available in text presented nearby or in a text tag associated with the image. Finally, remember that some users may respond more slowly than average to on-screen prompts, or may use an AT that slows down their rate of response. To accommodate this difference, always allow the user to customize timing requirements.

**Common interactive accessibility problems include:**

- requiring the user to use the mouse even when not specifically drawing on screen.
- activities that require monitoring information at one side of the screen while simultaneously entering information on the other side.
- numerical information that is shown visually (e.g., thermometer readouts).
- lesson directions provided in audio without text transcript or captions.
- text that is displayed over a patterned background.

**Learning system developers may enhance the accessibility of interactive exercises for all users by following these practices:**

- ensure that all actions can be completed from the keyboard.
- provide features that allow users to access multiple sources of information separately, even when they are delivered simultaneously.
- allow users to customize any timing of events, including having unlimited time to complete a task.

**Content creators or educators may enhance the accessibility of interactive exercises for all users by following these practices:**

- present information in ways that are accessible to both blind and deaf users and to deaf-blind users. Include an accessible text representation of all information.
- ensure readability of screens containing complex backgrounds by providing a visually simpler version as well.

**Resources:**

NCAM prototypes that demonstrate a talking math game and a talking science simulation at:  
<http://ncam.wgbh.org/cdrom/guideline/download.html>

The Digital Field Trip to the Rainforest CD from Digital Frog International “AT Version” with an accessible interface, descriptions of each image, and other access features at: <http://www.digitalfrog.com/products/rainforest.html>

## 8.5 Interactive Tutorials

Interactive tutorials present images of a software application in action are combined with audio or text narration, in order to teach users how to use that software. While adhering to interface guidelines, developers must also be sure to provide information in multiple modalities in order to meet the needs of all users. It is also important to remember that different users will need to know how to operate the software in different ways, depending on their accessibility needs.

**Common interactive tutorial accessibility problems include:**

- tutorial tasks that can only be completed using a mouse even though the skill being taught can be completed using a keyboard in the full program.
- narration that is provided in audio only without a text version for users with hearing impairments.
- references to items by spatial location without orientation for users who cannot see the screen (e.g., “select the window on the left”).

**Learning system developers may enhance the accessibility of interactive tutorials for all users by following these practices:**

- ensure that interactive tutorials can be operated by keyboard as well as by mouse.

**Content creators or educators may enhance the accessibility of interactive tutorials for all users by following these practices:**

- provide directions for using the keyboard, in addition to the mouse.
- provide text directions in place of or alongside audio directions.
- provide descriptions of screen layouts in text to assist visually impaired users.

## 8.6 DVDs, Consumer Electronics, and Handheld Devices

Learning tools not available on computers may also present accessibility issues. Other solutions, for set-top boxes such as digital cable, are in development. Access techniques for devices such as handheld devices or personal digital assistants, require research. Accessibility techniques for these devices will inevitably be different from those provided for mainstream computer applications, since AT is not currently available on these platforms.

### **Common DVD, consumer electronics, and handheld device accessibility problems include:**

- on-screen menus that are visual and cannot be used by blind users.
- touch screen interfaces that are inaccessible to blind users and users with physical disabilities.
- handheld device interfaces with small buttons that may be difficult to operate.

### **Technology developers may enhance the accessibility of DVD, consumer electronics, and handheld devices for all users by following these practices:**

- provide talking interfaces through built-in features. For example, DVDs and set-top boxes can provide talking menus (see “Examples and Resources” below).
- consider using the emerging Alternative Interface Access Protocol for home networking and other IT products (see “Examples and Resources” below).
- when including multimedia presentations, ensure that access to content is provided by following Section 5 Guidelines for Accessible Delivery of Text, Audio, Images, and Multimedia.
- choose a platform that offers as many access features as possible given the range of learning features required.
- prepare content so that it can be viewed and used on personal computers or other devices, in addition to the target device.

### **Examples and Resources:**

- DVD Accessibility:
  - Technical details on how to create talking DVD menus are forthcoming from NCAM at: <http://ncam.wgbh.org/convergence/>
  - Two accessible DVDs are available from PBS: “Abraham and Mary Lincoln: A House Divided”, and “Marcus Garvey: Look for Me in the Whirlwind” and include talking menus for access to navigation by blind viewers and access features for the program for both blind and deaf viewers. Available from PBS at: <http://www.shop.pbs.org>
  - Universal Studios’ Dr. Seuss’ “How the Grinch Stole Christmas” DVD also includes talking menus and other access features.
- Set-Top Box Accessibility:
  - Information and a prototype Electronic Program Guide are available from NCAM at: <http://ncam.wgbh.org/convergence/>
- Accessible Kiosks and Touch Screen Interface Accessibility:
  - Information is available from the Trace Center on their EZ Access Interface at: <http://trace.wisc.edu/world/kiosks/ez/>
- Alternative Interface Access Protocol:
  - The V2 committee of the US National Committee for Information Technology Standards (NCITS) is charged with developing national standards that will provide access features for users with disabilities. Information about Technology Access Interfaces is available at: <http://trace.wisc.edu/world/v2/index.htm>

## 8.7 Accessibility Information for Operating Systems and Development Platforms

Some techniques for creating accessible software are specific to the development environment or operating system (OS). This section provides resources for developers working on the Windows OS and Macintosh OS, for Web developers, and for Java developers.

### 8.7.1 Microsoft's Windows OS Accessibility

Microsoft provides detailed information on building accessible software for the Windows platform. The Microsoft Accessibility and Disabilities Group has created tools, documents, and APIs that offer ways to take advantage of access features in the OS. These tools also suggest other ways to make software more accessible. The Microsoft Windows Guidelines for Accessible Software Design provides comprehensive guidelines for creating accessible software.

In particular, the Microsoft Active Accessibility API (MSAA) uses programmatic means to help software communicate with ATs. MSAA exposes elements of the screen and presents their current state. It also exposes the focus, the active area of the screen. Using MSAA, software developers can use entirely custom graphical interfaces while still making each element known to an AT that has been programmed to read this information and convey it to the user.

#### Resources:

- Microsoft Accessibility Home Page at: <http://www.microsoft.com/enable/>
- Microsoft Accessibility: Information for Developers at: <http://www.msdn.microsoft.com/at/>

### 8.7.2 Apple's Macintosh OS Accessibility

All Macintosh computers ship with several accessibility features already installed. These features support users with sensory or physical disabilities. Developers may want to test their products with these features activated in order to determine whether their software is operable by users requiring AT. Some of these pre-installed accessibility features include:

- text-to-speech technology including the PlainTalk speech synthesizer.
- voice recognition technology.
- CloseView, a built-in screen magnifier for low-vision users.
- StickyKeys, software that allows users to strike keys one at a time in cases where two or three keys would normally be pressed simultaneously, such as Shift+F9.
- MouseKeys, software that allows users to control all mouse movements by typing on the numeric keypad.

In addition to the built-in accessibility features for the Macintosh, Apple maintains a list of Mac-based ATs available from vendors outside of Apple. OutSPOKEN is the only screen reader developed for the Macintosh platform. To ensure product compatibility on the Macintosh platform, developers should test their products using OutSPOKEN. Macintosh users who are blind will not be able to use educational software if the product is not compatible with this screen reader.

Apple maintains a developer website that offers an array of resources including the Macintosh Human Interface Guidelines. This document provides “authoritative information on the theory behind the Macintosh ‘look and feel’ and guidelines for using individual interface components. This book includes many examples of good design and explains why one implementation is superior to another.”

#### Resources:

- Apple Education Disability Resources contains links to information about built-in accessibility features for the Macintosh, as well as a link to the Mac Access Passport, a database of third-party disability products for the Macintosh at: <http://www.apple.com/education/k12/disability/>

- Macintosh Human Interface Guidelines. Accessibility information is referenced throughout this document and is available comprehensively at:  
<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-2.html#avail1-0>
- Universal Access information is found in Macintosh Human Interface Guidelines available at:  
<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-40.html#MARKER-9-55>
- The Human Interface Design Principles are found in Macintosh Human Interface Guidelines. Key accessibility information is included in the following section on the Apple website:  
<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-16.html#HEADING16-0>
- The home page for Alva Access Group, developers of the outSPOKEN screen reader and inLARGE, a screen magnifier. Programmers should download Alva's time-based demos of their products for testing purposes at:  
<http://www.aagi.com>

### 8.7.3 Web Accessibility

The definitive source for information on accessibility in Web technologies is the Web Access Initiative at the World Wide Web Consortium (the W3C WAI). The WAI provides recommendations for achieving accessibility using W3C technologies for Web content, authoring tools, and user agents such as browsers and media players. Information is available on W3C technologies including, XML, HTML, SMIL, CSS, and SVG.

#### Resource:

- The Web Access Initiative website: <http://www.w3.org/wai/>

### 8.7.4 The Java Platform

The Java platform is an attractive development environment for creating accessible educational software. Java's strongest feature is the accessibility support that is built into Java's core structure. Its Swing user interface components support accessibility. They include an effective keyboard interface. The Java accessibility API, a standard extension in the Java 2 platform, enables Java-based ATs to interact effectively with mainstream applications. The Sun access team has also developed the Java Access Bridge, which allows users to run Java applications with their platform-specific ATs. The Java accessibility API also contains several properties that enable developers to fine-tune the manner in which assistive technologies present their interfaces.

#### Resources:

- Sun Microsystems' Accessibility Program - Developer Information at: <http://www.sun.com/access/developers/>
- The Java Access Bridge: <http://www.java.sun.com/products/accessbridge>
- Java Accessibility Helper identifies areas in an application's UI where the accessibility support has been improperly used. Follow the link from Sun's Accessibility Program at: <http://www.sun.com/access/>
- The Java Tutorial - Accessibility Section at: <http://java.sun.com/docs/books/tutorial/uiswing/misc/>
- Guidelines for Writing Accessible Applications using 100 percent Pure Java at:  
<http://www-3.ibm.com/able/snsjavag.html>

### 8.7.5 Other Operating Systems

A longer list of accessibility techniques for additional development environments is available at:  
<http://www.w3.org/TR/ATAG10-TECHS/#check-use-system-conventions>



## 9. Guidelines for Testing and Assessment

The guidelines in this section are intended to promote access to tests and assessments by people with disabilities. We use the terms “assessment” and “test” virtually interchangeably, though “assessment” is often considered a more general term that includes tests. While distance learning presents important new opportunities for delivery of tests and assessments, much of this section pertains to almost any computer-based delivery or even non-computer-based delivery methods.

Since online assessment or testing takes place in a context where media is presented, even if that media is just text, most of the principles and guidelines pertaining to content described in other sections of this document also apply here. This section outlines some of the differences between the needs of testing and assessment and those of content-only delivery.

Much work has been done throughout the world to make media accessible to all, including these very guidelines. Testing and assessment involve a step beyond media because they are more intimately tied-in with processes, activity, and organizations. Many of the central problems in achieving widespread accessibility of online assessment systems are, at this stage, problems needing research to clarify the issues and identify appropriate technological solutions. This section presents the state of current issues and some principles that can be identified to point the way for future work.

This document is addressed to various stakeholders who provide the assessment, including assessment designers, content authors, validation bodies, organizations providing assessment delivery, and so forth. Different stakeholders may be involved in different parts of the process and concerned with very different kinds of assessment with different constraints. For example, a test designed to detect dyslexia in young children and a self test for Computer Science undergraduates to diagnose their own programming skills may need to handle accessibility very differently.

As discussed below, one over-arching consideration is that in high-stakes testing, where the consequences for individuals are great, there needs to be a very careful consideration as to the potential effects of accessibility provisions on the validity of the scores derived from the testing.

There are two broad classes of assessment, and their accessibility requirements differ.

### Low-Stakes Assessment

Low-stakes assessment is a form of assessment encompassed by the immediate process of learning, often in a very short feedback loop, such as exercises or quizzes. Sometimes this is called “formative” assessment or even just “feedback”. The essential characteristics are immediacy and the lack of serious consequences contingent on performance.

Making low-stakes assessment accessible means providing equivalent content modalities and access mechanisms wherever possible. For example, buttons in a form or other interactive controls should not be labeled only with images and without accompanying text. A good solution would provide the label information in a form that allows for runtime generation of equivalents or alternative renderings. Controls must also be accessible through the keyboard as well as the mouse. The overall goal is to make systems and content as flexible and adaptable as possible. All of the solutions explained elsewhere in this guide apply at this media level. The goals of this kind of assessment are intimately tied up with learning. In general, accessibility requirements can be in conflict with the need to ensure assessments are valid, but here we are chiefly concerned with learning and so this is not a serious issue.

### High-Stakes Assessment

High-stakes assessment has consequences that may have a serious impact on the life-course of the participant. An example might be a university entrance examination.

It is important that a high-stakes assessment be fair to all candidates and not offer advantages to one group over another. The goal is to make systems and content as accessible as possible while adhering to validity requirements. Ideally, for example, an examination candidate should not be excluded from an examination or have performance disadvantaged where the candidate’s disability is unrelated to the skills or knowledge being tested.

Assessment providers need to be able to control, in a flexible manner, which alternatives are available in order to protect the validity of inferences made from the test scores. It may be that allowing one set of alternatives has no consequences of fairness for particular candidates for that assessment while allowing others does. What is needed is

fine control and rigor in the use of alternatives, and building-in this kind of control can affect design choices for both the content, as well as the delivery system and authoring tools. Taking the images and alternatives problem described above as an example, one delivery system may provide for particular alternatives to be enabled or disabled under system control at runtime, whilst another may not. For some group of candidates and assessments this may or may not be important, but the widest reuse of accessible assessment material for high-stakes assessment can only be obtained with the provision of appropriate mechanism. There are however many other considerations in building-in such a mechanism.

Developers and educators should make good use of alternative media and delivery technologies when designing assessment systems and tools. They will likely need to allow assessors greater control over alternatives as well as better use of accommodations and easier integration of disabled users. Accommodations refer to changes in the content, format, and/or administration procedure for individuals who are unable to take assessments under standard (or default) conditions.

## 9.1 Testing and Assessment Challenges

There are several challenges in developing and delivering tests and assessments that can be used by people with disabilities, including:

- tests and assessments too often lack important accessibility features.
- accessibility features may conflict with validity considerations.
- some testing organizations may lack a coherent rationale for test access decisions.
- reuse of test content may be limited by a variety of factors.

## 9.2 Principles for the Accessibility of Testing and Assessment

The following are four principles to help guide the implementation of solutions for ensuring access to testing and assessment by people with disabilities. Developers should:

- consider accessibility standards from the earliest design stages.
- follow standards for testing and assessment that promote test validity.
- build a coherent rationale for the assessment.
- develop a reuse strategy that includes test content, designs, and more.

The relative emphasis among these principles is different for low-stakes than for high-stakes assessment.

### 9.2.1 Consider Accessibility Standards from the Earliest Design Stages

From the earliest design stages, test developers consider accessibility standards. For example, they need to consider including text to accompany images, as well as captions and auditory descriptions to accompany video. Such alternative content is usually best created by those individuals who are well acquainted with the original content. Designing alternative content *early* in the process can help eliminate costly retrofits. Details about such accessibility standards are found throughout other parts of these guidelines.

### 9.2.2 Follow Standards for Testing and Assessment That Promote Test Validity

Validity is a preeminent concern in tests and assessments. The *Standards for Educational and Psychological Testing*, (hereafter called Standards), published by the American Educational Research Association (AERA), American Psychological Association (APA), and National Council on Measurement in Education (NCME) defines validity as:

The degree to which accumulated evidence and theory support specific interpretations of test scores entailed by proposed uses of a test (Standards, 1999).

According to Willingham and Cole (1997):

Validity is the all-encompassing technical standard for judging the quality of the assessment process. Validity includes, for example, the accuracy with which a test measures what it purports to measure, how well it serves its intended function, other consequences of test use, and comparability of the assessment process for different examinees. (p. 228).

Validity is a major theme in chapter 10 of the Standards, which focuses on testing individuals with disabilities. Standard 10.1 (the first of 12 disability-related standards) is essentially about validity. It states: “In testing individuals with disabilities, test developers, test administrators, and test users should take steps to ensure that the test score inferences accurately reflect the intended construct rather than any disabilities and their associated characteristics extraneous to the intent of the measurement” (Standards, 1999, p. 106).

While accessibility features are essential for *overcoming threats* to validity, some accessibility features can actually *pose threats* to validity. Oftentimes, the nature of the threat and the proper solution can be ascertained through basic logical analysis. For example, if a physical disability would prevent a person from recording their responses on a math test when administered with the default response method, then the incompatibility between their disability and the default response format constitutes a major threat to validity. It is fairly obvious that providing an alternative response method such as using an AT or a human scribe to record the answers would likely be a proper accommodation. Providing such an accommodation could improve validity.

But sometimes an accessibility feature can diminish validity. Suppose, for example, that an art test that is intended to measure a student’s ability to write descriptions of artwork (such as paintings), and that the questions present visually displayed pictorial images of the artwork as prompts. Standards for accessibility might require that text descriptions of the artwork be made available to the test taker. However, to do so would make it virtually impossible to detect the test taker’s proficiency in producing the text descriptions, thereby posing a threat to validity. For a test taker who is blind, a better approach might be to provide a tactile graphic of the artwork rather than the text description. Or alternatively, if the questions with pictorial images constitute only a small portion of the total art test, one might consider *not* administering such questions to test takers who are blind. Thus, for any given test and test-taker, not every accessibility feature necessarily promotes validity. Accessibility features need to be examined for their impact on validity.

Another example concerns the use of a readaloud accommodation in a test of reading comprehension. The test performance of a person who is blind might benefit from having content read aloud to them by synthesized speech or by prerecorded or live human speech. However, suppose that this test of reading comprehension includes decoding words from letters as part of the construct (intent of measurement). In this case, a readaloud accommodation would essentially prevent student performance from serving as a source of evidence about decoding ability since hearing the words spoken would not require decoding in order to answer questions correctly. This scenario would suggest that providing the readaloud accommodation may invalidate the test and that, therefore, the readaloud accommodation should *not* be allowed. However, the decision about whether to actually allow a readaloud accommodation might involve additional considerations. For example, one might ask additional questions:

- How *central* is decoding to what the test is attempting to measure? If decoding is not central, then this factor might tend to favor allowing the accommodation.
- What does research say about what this particular test *actually* measures (as opposed to what it is *intended* to measure)? If research showed that known differences in decoding ability did not actually affect performance on this test by people without disabilities, this might suggest that the test was not *actually* measuring decoding (even though the test was *intended* to measure it). These circumstances would tend to favor allowing the readaloud accommodation for the student who is blind, simply as a matter of equity. (However, this discrepancy between the intended and actual influence of decoding on test performance for non-disabled test takers would suggest that decoding is under-represented in the test scores, which constitutes another validity-related problem.)
- Are there *other ways* of making the desired inferences about proficiency? If the student reads Braille, then presenting the test in Braille format might be a better alternative to a readaloud accommodation, since the test taker would need to possess decoding skill in order to perform well.

The need to examine many factors in making decisions regarding test takers with disabilities points to the need to be able to call on experts when needed. As suggested earlier, the rigor required for accommodation decisions will tend to be greater in high-stakes than in low-stakes testing. Thus, in many instances, basic logical analysis of what the test is intended to measure will make clear what kinds of accommodations will promote validity; in other cases, a more careful analysis and consultation with experts may be essential.

### 9.2.3 Build a Coherent Rationale for the Assessment

To the extent possible, one should seek to build a coherent rationale for the claims that one makes about the proficiencies (i.e., knowledge, skills, and abilities) of test takers, including test takers with disabilities. Evidence-centered assessment design (Mislevy, Steinberg, & Almond, in press), a design method that emphasizes explicit evidentiary argument, may help strengthen the technical quality of test or assessment. In the case of individuals with disabilities, this argument must encompass not only claims about student proficiencies, but also about comparability of scores based on different formats or conditions of administration for people with disabilities. Evidence-centered design focuses on developing a coherent argument involving:

- a) the specific claims that one wishes to make about test-taker proficiencies.
- b) the test-taker behaviors that reveal those proficiencies.
- c) the tasks that will elicit those behaviors.
- d) the provisions that address alternative explanations.

Consider, for example, a person who consistently answers test questions incorrectly. Generally, one would claim that the test taker lacks the targeted proficiency. However, an alternative explanation for poor performance would be that the test taker has a disability (e.g., blindness) that prevents the test taker from manifesting their true proficiency level under the default testing conditions (regular-sized visually-displayed text). Addressing such an alternative explanation might involve ensuring that (a) candidates are made aware how they can request testing accommodations, (b) the requests for accommodations are examined by a panel of qualified experts, and (c) if the accommodation is approved, the test taker has adequate opportunity to become familiar with the accommodation before using it in the actual test. Obviously, the efforts to address alternative explanations are all the more important in high-stakes settings. Note that the activities to address alternative explanations are diverse and that they often occur in the pre-assessment phase of testing, before the candidate has answered any questions.

Evidence-centered assessment design, particularly its domain modeling stage, can provide a systematic approach for examining accommodations and their appropriateness. Provision of appropriate accommodations seek to match test-taker background characteristics (e.g., blindness, knowledge of Braille) to task features (e.g., readaloud or Braille versions) that are intended to overcome access hurdles without conflicting with the claims that one wishes to make about test-taker proficiencies. The explicit modeling of proficiencies (both those targeted for measurement and those not targeted) can assist in identifying features of the task performance situation that are likely to result in sound measurement.

Evidence-centered design can also make explicit the constraints on the reporting of test scores. Social policy considerations, for example, can be important constraints on reporting of test results. Reporting information about accommodations might help a test score user interpret scores (Lewis, Patz, Sheinker, & Barton, 2002; Standard 10.11 of Standards, 1999). Yet laws and other rulings designed to address the possibility of discrimination against test takers with disabilities may restrict the inclusion of certain information in reports. Evidence-centered design's support for analyzing, modeling, and representing social policy, legal, and other constraints can promote the cohesion of various parts of an assessment argument.

As in any design process, the quality of the outputs from an evidence-centered assessment design process is dependent on the quality of the inputs. Some information that would be useful as inputs to the design process, such as studies of the impact of accommodations test performance and validity can be difficult to obtain due to factors such as small sample sizes, variability in examinees, and variability in accommodations (NRC, 1997; Standards, 1999; Pitoniak & Royer, 2001). While an evidence-centered approach cannot replace the generation of new knowledge through research and development, it can help leverage the value of existing knowledge through sharing and reuse between and among test developers and stakeholders and across related testing applications.

### 9.2.4 Develop a Reuse Strategy That Includes Test Content, Designs, and More

Reuse of assessments designs and of test content is a fundamental strategy for promoting efficient and valid assessments. Striving for reuse impels one to identify commonalities across assessment settings, audiences, and delivery systems.

One important kind of reuse that deserves careful attention is reuse of test content. For example, one of several goals in the design of an “accessible” test might be that the text portion of the test content could be reused across several presentation formats such as (a) visually displayed text for test takers who are deaf or non-disabled, (b) synthesized speech for test takers who are blind, and (c) Braille for individuals who are deaf-blind or blind. Ideally, there would be nothing peculiar to a specific presentation format that would require a different source text for each of these presentation formats. While complete reuse of test content may be a good design goal, one also needs to be ready for the possibility that content -- text content in this example -- might *not* be completely reusable across test formats. For example, acronyms that might make sense in print to a sighted test taker may voice improperly in synthesized speech used by a person who is blind. Ensuring proper access by the person who is blind and receiving content via audio might require, for example, that letters be separated by spaces to facilitate proper voicing. In some content areas, different formats may require supplementary media such as tactile (raised-line) drawings of math figures, and the on-screen content may need to alert the user about the availability of such. Specialized directions may be necessary for different formats. These kinds of differences between formats may reduce certain kinds of test content reuse.

It follows that a sensible strategy for reuse should include test content but should also extend further, for example, to include reuse of assessment designs. As suggested earlier, an assessment design that comprises an explicit argument can promote greater reuse. Furthermore, strategies for reuse should not overlook the reuse of personnel and other test development and delivery resources.

## 9.3 Delivery and Authoring Tool Principles for Testing and Assessment

Delivery and authoring tools need to support mechanisms for authoring, storing, and using accessible assessment content. Many of these mechanisms are identical to those described throughout this document, but some concerns are specific to the authoring and delivery of assessments.

**Note:** This is for work in a future version, but may be expected to include:

- information about purpose.
- information about accessibility properties of content.
- control information for enabling and disabling media and other alternatives.
- possibly the means for recording data about actual execution uses of alternatives (tracking).

## 9.4 Content Development Principles for Testing and Assessment

Enabling flexible, accessible assessment content will require encoding additional information in the meta-data about each assessment item and its alternatives. Other concerns are specific to the development of assessments.

### 9.4.1 Provide Information about Purpose

**Note:** This is for work in a future version, but may be expected to include purpose information which is needed at multiple levels including

- the level of an assessment.
- the level of an item so as to allow selection of appropriate items.
- at the level of a media alternative.

### 9.4.2 Provide Information about Accessibility of Content at Varying Levels

**Note:** This is for work in a future version.

**Resources:**

- AERA, APA, & NCME. [Standards] (1999). *Standards for Educational and Psychological Testing*. Washington DC: American Educational Research Association, American Psychological Association, National Council on Measurement in Education.
- BSI BS 7988 A Code of Practice for the use of information technology for the delivery of assessments. Available to order from <http://bsonline.techindex.co.uk/>
- Lewis, D. M., Patz, R. J., Sheinker, A., Barton, K. (2002). Reconciling standardization and accommodation: Inclusive norms and reporting using a taxonomy for testing accommodations. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, Louisiana.
- Mislevy, R.J., Steinberg, L.S., & Almond, R.G. (in press). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*.
- National Research Council [NRC] (1997). *Educating one and all: Students with disabilities and standards-based reform*. National Academy Press.
- Pitoniak, M. J., & Royer, J. M. Testing accommodations for examinees with disabilities: A review of psychometric, legal, and social policy issues. *Review of Educational Research*, 71 (1), 53-104.
- Willingham, W. W., & Cole, N. S. (1997). *Gender and fair assessment*. Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Willingham, W. W., Ragosta, M., Bennett, R. E., Braun, H., Rock, D. A., Powers, D. E. (1988). *Testing handicapped people*. Needham Heights, Mass.: Allyn and Bacon.

## 10. Guidelines for Developing Accessible Authoring Tools

In the effort to open the learning experience to all, it is critically important to make content creation a simple and natural part of the authoring process. One goal of these guidelines is to assist authors who may have limited knowledge of accessible authoring practices. Ideally, authors who may not have any knowledge of accessible authoring practices should still be able to create accessible material using proper authoring tools.

An *authoring tool* is any software tool that can be used to create content that can be displayed in an online learning environment. Authoring tool accessibility should be approached from two perspectives:

- The authoring tool should support and encourage the author to create accessible content.
- The authoring tool should be accessible to authors with disabilities.

An authoring tool can support and encourages the creation of accessible content or learning materials by following several guidelines delineated in the W3C WAI Authoring Tool Accessibility Guidelines at:

<http://www.w3.org/TR/WAI-AUTOOLS/>

The Guidelines are accompanied by a techniques document that suggests how the guidelines can be implemented using various tools. In summary, an authoring tool should encourage the author to adopt accessible authoring practices and assist the author in checking and repairing content as it is being authored. The tool should not only offer mechanisms for creating accessible content, but should make these mechanisms easy to use. Accessibility supports should be seamlessly integrated into the look and feel of the tool and well documented in the help function and documentation supplied with the tool. Any content automatically generated by the tool or any choices made for the author by the tool, should also support accessibility.

It is critical that the authoring tool itself must be accessible to educators and learners with disabilities. Tool developers must follow platform-specific user interface accessibility standards. They must also provide supports for accessible navigation of the document and provide flexible presentation of the content to be edited (as discussed in both the W3C WAI Authoring Tool Guidelines and Techniques documents at: <http://www.w3.org/WAI/AU/>)

In a courseware development environment, these guidelines need to be applied equally to the interfaces used by the administrator, the educator or instructor and the learner or student. All three of these user groups are potential authors. The courseware development environment should also provide checking and repair tools to help detect and fix access problems in content authored using third-party authoring tools. A powerful means of ensuring that learning content is accessible within an educational institution is to provide templates and models that are accessible.

### Resource:

- A review of common courseware authoring tool accessibility issues and solutions can be found at: <http://snow.utoronto.ca/access/courseware/index.html>



# 11. Guidelines for Topic Specific Accessibility

This section focuses on specific domains of content that depend heavily on modes of presentation that fall outside of the bounds of pure text.

It should be noted that every learner stands to benefit from content made accessible to those with disabilities, particularly with respect to the specific topic areas under consideration in this section. For example, mainstream users of the Web rarely get access to “live” mathematics, and many users of content on the Web find it difficult to comprehend mathematical content. Many of the support strategies devised to make mathematical content more accessible to those with disabilities also provide support to the larger student population.

Similarly, science students also need to decode mathematical expressions, and they also must have access to laboratory experiments. Online simulations of lab exercises or remote use of real lab equipment are helpful to all students.

Likewise, graphs and diagrams can present a barrier to learning even for students who have no visual impairment. By supporting graphical content with text descriptions and discussions, developers improve the prospects of all learners who use their products.

And finally, students who study material presented in foreign languages or with the use of non-standard codes (e.g., music notation) also stand to benefit from supporting text.

The challenge of presenting alternative “views” of symbolic and semantic content makes up the leading edge of accessibility research today and many problems have yet to be solved. This document presents an overview of approaches currently in development or in use in fields such as:

- Mathematics
- Sciences
- Simulations and immersion
- Robots and telepresence
- Charts, diagrams, and tables
- Geography and maps
- Music
- Languages

## 11.1 Mathematics

In general, people use mathematics on the Web to represent ideas and to teach mathematics itself. Mathematical expressions are also used interactively to manipulate data, for both educational and work-related pursuits.

Two main types of problems inhibit accessibility. As with other graphical content, mathematical symbols are inaccessible when presented simply as graphics without supporting text descriptions or discussions. But even when mathematical expressions are expressed as text, they sometimes cannot be interpreted by screen reading software.

This document goes on to examine accessibility problems and their solutions in some detail, but in general, we can summarize the steps that content creators and users should take to improve accessibility.

**Content creators or educators may enhance mathematical accessibility for all users by following these practices:**

- chunk equations through use of stylesheets and SVG, which can also be used to organize the display of mathematical expressions.
- encode mathematical expressions using MathML.



**Currently, following W3C recommended procedures, a user can do the following:**

- employ mathematical editors to produce screen versions of expressions using appropriate symbols and then exporting them encoded in XML. These presentations use MathML presentation tags in order to be correctly reproduced on screen or on paper.
- write semantic versions of mathematical expressions, using formalized techniques or exporting text expressions from suitable software.
- take source code from Web pages and use it in mathematical applications (that process MathML).

**11.1.1 Chunking of Mathematical Expressions****Common mathematical expressions accessibility problems:**

Mathematical writing can be complex. The ability to choose which parts of an expression to make visible or available helps both novices and experts to read a mathematical argument or text. “Chunking” provided by both stylesheets and SVG organizes the display of mathematical expressions. The following expression provides an example.

“This formula:  $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$  makes it possible to predict the result with accuracy.”

**Possible solution for accessible chunking of mathematical expressions:**

With an accessible math reader in place, the user would hear the above spoken as:

“This formula, a cubed plus b cubed equals brackets, a plus b, brackets, a squared minus ab plus b; squared, makes it possible to predict the result with accuracy.”

Learners would better comprehend the formula if they were allowed to choose a stylesheet that omits the formula from the initial reading. In this case, the learner would first hear:

“This formula makes it possible to predict the results with accuracy.”

To then hear the formula, the learner could switch to a mathematical stylesheet to read the formula in detail.

**11.1.2 Inaccessible Mathematical Notation****Common problems with inaccessible mathematical notation:**

If equations are presented as graphics, users who cannot see the graphic have no idea what is on the screen. Other problems include:

- **Inability to enlarge.** The user with a visual impairment may need to enlarge the notation, or to change its color or the background against which it is displayed. When equations are presented as graphics, enlarging them creates blocky, unreadable views, and changing colors is difficult.
- **Inability to manipulate.** If the notation is designed for a screen reader it can be understood, but it may lack additional information necessary for the learner to use the expression within a standard symbolic manipulation application.
- **Inability to use chunking.** The expression may not be properly structured to help a visually impaired user break it down and work with it in pieces (or chunks) of notation. There should be a regular, standardized way to structure expressions, or the user with a vision disability may have difficulty.

**Some solutions for accessible mathematical notation:**

- Include a text rendering of the equation using ALT text or long description. Text provides an alternate and in some ways equivalent rendering of mathematical expressions. However, a standard way of reading the mathematics, such as Mathspeak, must be selected and the learner must be informed about it. An excellent description of Mathspeak is provided at the Web address: <http://www.rit.edu/~easi/easisem/talkmath.htm>  
A short piece of text can be inserted into the ALT attribute of an equation image. If the mathematical text is long, it may be better placed in a long description or by using a “d-link” next to the image.  
Examples of this technique are available on the MIT PIVoT Website in Chapters 2 and 17 of the Pivot textbook. Request a guest account at: <http://curricula2.mit.edu/pivot/>

- Provide mathematical expressions in standard (ASCII) text with special formatting. For example:  
 $a^3+b^3=(a+b)(a^2-ab+b^2)$ .

Without formatting, this equation appears as  $a3+b3=(a+b)(a2-ab+b2)$ .

The correct meaning of the expression is communicated by both characters and formatting. However, formatting is usually not available to the user. One solution is to ensure that formatting is embedded in stylesheets that can be rendered in a different modality where necessary. An audio stylesheet might render superscript characters in a higher pitch than others, or use a tone to identify changes from normal text to super- or to sub-script text.

### 11.1.3 Encoding Mathematical Expressions Using MathML

Mathematical Mark-up language, or MathML, (from W3C at <http://www.w3.org/Math/>) is an XML mark-up language for mathematical notation written with universal accessibility as a priority. It offers two formats for representing an expression: presentational tags and semantic/content tags. Users can write expressions using an accessibility-compliant editor and then view or access symbolic representations on the Web with an accessibility and MathML standards-compliant user agent. Usually this requires a MathML-enabled Web editor such as Amaya (freely available from the W3C at <http://www.w3.org/Amaya/>), or a mathematics application (see the W3C's list of MathML compliant browsers and the list of MathML compliant mathematical software).

**Content creators and/or educators may enhance the accessibility of mathematical content for all users by following these practices:**

- use a mathematics generator that produces TeX or LaTeX and MathML. (TeX is a typesetting system used for formatting complex mathematical expressions, and LaTeX is a Generalized Mark-up Language macro package for the TeX typesetting program.) Converting one into the other may require making some small adjustments.
- embed mathematical expressions within an XHTML page with references to the different namespaces.

**To help improve the accessibility of mathematical content users should:**

- request that writers produce a MathML version of expressions and that there is an alternative TeX or LaTeX version available.
- acquire a MathML plug-in for their browser.
- use AsTeR to read the LaTeX or TeX version (see subsection 11.1.4).

If mathematics on the Web is marked up properly in MathML, it may be possible to access it directly by using a mathematical symbolic editor that understands MathML and is itself accessible. A comprehensive list of these is available from the W3C math website at: <http://www.w3.org/Math/>

### 11.1.4 Localized Applications for Mathematical Accessibility

**Available solutions for mathematical accessibility include:**

- AsTeR (Audio System for Technical Readings) is software that formats and renders mathematical notation auditorily. The AsTeR site explains the software: "AsTeR has three main components. A recognizer parses LaTeX notation and creates an internal representation that is easier for the program to manipulate. An audio formatting language, called AFL, renders the parsed text using both speech and non-speech sounds. The third component is a facility for audio browsing, or actively traversing the structure of a document."  
 The AsTeR website is: <http://www.cs.cornell.edu/Info/People/raman/aster/aster-toplevel.html>  
 A paper about AsTeR and its use is available from the American Scientist: <http://www.amsci.org/amsci/issues/Comsci96/compsci96-03.html>
- HP EzMath is an older solution that provides an easy-to-learn notation for embedding mathematical expressions in Web pages. The notation is inspired by the way that expressions are spoken aloud. The notation also uses some abbreviations for conciseness (e.g.,  $x^y$  denotes  $x$  raised to the power  $y$ ). EzMath is in the process of being upgraded and will soon be open source on Source Forge (a website repository of open source code and applications) at: <http://sourceforge.net/>
- IBM TechExplorer is a plug-in for Navigator and Internet Explorer, an IE 5.5 XML Behavior, and an ActiveX control for applications like Microsoft PowerPoint and Word. TechExplorer enables the display of TeX, LaTeX, and MathML documents and allows publishing of interactive scientific material on the Web. Version 3.1 includes

full support for MathML 2.0, augmented Latex display, new ways to enliven documents via scripting/programming and a Web equation editor. New to 3.1 is a Macintosh version, Mathematica connectivity, and IE MathML behavior. It is available at: <http://www.software.ibm.com/techexplorer/>

### **A Note for Tool Developers:**

Authoring tool developers, including people who make templates for others to use, face accessibility problems, specific to MathML. MathML is not yet widely used, is not native to commonly used browsers, and many of the older courseware packages are not accessibility-aware. The landscape *is* changing but it is still unlikely that courseware developers will be able to alleviate the problem quickly enough. However, there are plug-in solutions that might be useful to courseware developers.

In the meantime, toolmakers can help their users solve this problem by presenting best practice information and, where possible, incorporating software that supports good mathematics mark-up.

### **Resources:**

Visit the MathType website at <http://www.dessci.com/webmath/> to learn how to write mathematics notation in Word (Equation Editor), translate them into MathML using WebEQ generator (a shareware translator) and allow users to view the notation using graphical browsers augmented with the free WebEQ plug-in. The W3C Math home page also provides MathML information at: <http://www.w3.org/Math/>

### **11.1.5 Accessible Mathematical Applications and Devices**

The Audio-Accessible Graphing Calculator is a self-voicing Windows application that is under development by the Science Access Project at Oregon State University. It includes the capabilities to:

- compute and display visually either of two functions, their sum, or their difference.
- display the above as an audio tone plot.
- permit piece-by-piece audio browsing.
- print output to any Windows printer including the Tiger tactile graphics embosser.
- act as a universally usable on-screen scientific calculator.
- act as a powerful expression evaluator.
- accept input of tabulated data for display.
- compute statistical functions for tabulated data.

The Audio-Accessible Graphing Calculator is available at: <http://dots.physics.orst.edu/agc.html>

Braille ‘n’ Speak is a Braille note taker that can also serve as a calculator which can do standard math. Additional software is available for students needing the power of a financial calculator. The Braille ‘n’ Speak can also be used to generate tactile graphs using the graphing calculator software Graphit in conjunction with any Braille embosser.

Information about the Braille ‘n’ Speak is available at:

[http://www.freedomscientific.com/fs\\_products/notetakers\\_bns.asp](http://www.freedomscientific.com/fs_products/notetakers_bns.asp)

Information about Graphit is available at:

[http://www.freedomscientific.com/fs\\_products/notetakers\\_related.asp#graphit](http://www.freedomscientific.com/fs_products/notetakers_related.asp#graphit)

## **11.2 Science**

As with mathematics, it is important to develop a mark-up standard that can be used to encode and decode scientific expressions for accessibility. In addition, many branches of science have discipline-specific 2D and 3D representations that need to be rendered on the screen.

**The Problem:**

Equations are often presented as graphics to users who cannot see the graphic and have no idea what is on the screen. Also, scientific symbols are often inaccessible to people who are using screen reading software.

The following subsections present solutions to enhance the accessibility of chemistry and physics.

Other fields, such as astronomy, biology, and geology still await the development of appropriate accessibility aids.

A useful resource on accommodating students with disabilities in math and science classes from the University of Washington's DO-IT program can be found at: <http://www.washington.edu/doi/MathSci/>

**11.2.1 Chemistry**

The chemistry equivalent to MathML is ChemML. ChemML enhances accessibility by allowing the developer to use suitable stylesheets. Like MathML, ChemML must be embedded within XML (or XHTML). CML is a version of ChemML.

For more information about ChemML visit: <http://www.xml-cml.org/information/position.html>

To make chemistry accessible, ChemML interoperates with other mark-up languages and XML protocols including:

- XHTML for text and images.
- SVG for line diagrams, graphs, reaction schemes, phase diagrams, etc.
- PlotML for graphs and MathML for equations.
- XLink for hypermedia (including atom-spectralPeak assignments, reaction mapping).
- RDF and Dublin Core for meta-data.
- XML schemas for numeric and other data types.

Accessible science experiments in chemistry and physics are available from Barrier Free Education at:

<http://barrier-free.arch.gatech.edu/Lab/index.html>

The Accessible Periodic Table (ATRC) is a haptic periodic table under construction that will give students a "feel" for the periodic table and the elements. It is a worthy attempt to use haptics to increase the dimensions of data available from the table (e.g., the weight of elements). For more information visit: <http://nide.snow.utoronto.ca/hperiodic.htm>

"Teaching Chemistry to Students with Disabilities" is available from the American Chemical Society Committee on Chemists with Disabilities at: <http://membership.acs.org/C/CWD/teaching/start.htm>

"Working Chemists with Disabilities: Expanding Opportunities in Science" from the American Chemical Society Committee on Chemists with Disabilities is available at: <http://membership.acs.org/C/CWD/workchem/start.htm>

**11.2.2 Physics**

The Access to PIVoT (Physics Interactive Video Tutor) project identifies and addresses the needs of deaf and blind students in the design of user interface and navigation systems. It all supports the presentation of video, text, illustrations, graphs and tables. NCAM is working with MIT to develop and apply methods for improving website layout and navigation, access to complex graphics, illustrations and equations, as well as to create captions and audio descriptions for the multimedia.

PIVoT allows students to interact with a renowned MIT physics professor using streaming digital video over the Internet. The PIVoT website offers online multimedia content for students in MIT's introductory physics course covering classical mechanics.

For more information on PIVoT visit:

<http://ncam.wgbh.org/webaccess/pivot/>  
[http://caes.mit.edu/research/pivot/report\\_2000-10.html](http://caes.mit.edu/research/pivot/report_2000-10.html)

Request a PIVoT guest account at:

<http://curricula2.mit.edu/pivot/>

## Resources

The American Physical Society provides information on physics and persons with disabilities at: <http://www.aps.org/memb/disabilities.html>

## 11.3 Simulations and Immersion

### The Problem:

Students are not always able to participate in live experiments. Either distance or physical disability can stand in the way of a traditional “hands-on” learning experience.

### Some solutions for accessible simulations and immersion:

Simulations and artificial environments created by “immersive technologies” can provide equivalent alternatives to real life experiences. Working with one’s hands in three dimensions provides alternative opportunities for interaction with content. Access to simulations and immersive experiences should be available for all users. Several technologies currently exist, including:

- Electric Chemistry Set. This is a learning resource based around a database of chemical reactions. Up to three chemicals may be mixed and the reaction temperature varied. The application displays results as chemical equations and in quantitative tables. Visit: <http://dbweb.liv.ac.uk/itsnpssc/softfinx.asp>
- Immersion Medical Simulators. These technologies combine tactile feedback sensations with high-power graphics and high-fidelity sound. Medical professionals can thus practice difficult procedures in a risk-free virtual environment. Visit: <http://www.immersion.com/products/overview.shtml>
- Phantom Haptics 3D. The sense of touch allows users to interact directly with digital objects; exactly as they do in the real world. Modelers sculpt as naturally in digital form as they do in clay and surgeons perfect their crafts with virtual patients instead of actual people. Visit: <http://www.sensable.com/haptics/haptics.html>
- Biology Simulations. Several interactive biology simulations are currently in development, including an interactive frog dissection lab experience and an experiment in photosynthesis. For more information visit Barrier Free Education at: [http://barrier-free.arch.gatech.edu/Lab/sim\\_bio.html#interact\\_frog](http://barrier-free.arch.gatech.edu/Lab/sim_bio.html#interact_frog)
- Physics Simulations. Several interactive physics tools are in development or are available. These include: *Explore Science*, *Interactive Physics v 3.0*, *Interactive Physics with Java*. For more information, visit Barrier Free Education at: [http://barrier-free.arch.gatech.edu/Lab/sim\\_physics.html](http://barrier-free.arch.gatech.edu/Lab/sim_physics.html)

## 11.4 Robots and Telepresence

### The Problem:

For students who participate in laboratory work from a distance or who have a physical disability, the inability to physically manipulate objects can interfere with the learning process.

Several technologies that make robots available to students to allow them to manipulate objects are under development.

### Possible solution for accessible robots and telepresence:

- One example is PEARL (Practical Experimentation by Accessible Remote Learning), developed at The Open University in the UK. PEARL is designed to make science and engineering education accessible. The technology seeks to go beyond the capabilities of multimedia packages that offer “virtual science” experiences. PEARL allows students to work with actual physical objects and enhances the teaching of scientific inquiry and engineering practices. PEARL will allow students to work together over the Internet (or campus Intranet). Their experience would mimic the kind of cooperation found in a traditional laboratory environment. They will be able to interact with the remote experiment, change parameters and in some cases modify and design their own experiments. They will

discuss their actions and what they anticipate will happen, while observing and analyzing their results. For more information see “Practical Experimentation by Accessible Remote Learning (PEARL)” at the Web address: <http://kmi.open.ac.uk/projects/pearl/summary/index.htm>

## 11.5 Charts, Diagrams, and Tables

### The Problem:

Because of their visual nature charts, diagrams, and tables present a real problem for learners with visual impairments. But other students also have difficulty interpreting graphical information, including those with learning disabilities. To be accessible charts, diagrams, and tables cannot be allowed to stand on their own. They must be paired by other kinds of information and conform to the needs of interpretive technologies.

### Some solutions for accessible charts, diagrams, and tables:

Making charts, diagrams, and tables more accessible means presenting the same information via a non-visual sensory channel. Interactive audio descriptions elaborating the graphic elements and guided by the user are a standard tool.

However, a number of technologies that directly translate the graphic into an object that can be touched are in development and they are proving to be powerful learning tools. These include both technologies that create a permanent physical object through a version of printing, as well as technologies that use a refreshable touch display analogous to a computer screen.

Using the sense of touch to comprehend graphical information depends on the user’s ability to assimilate a mental spatial image of the material. A tactile image reduces the mental effort, and a combination of touch and audio (or braille) feedback has been found to be very successful in making maps and other graphics accessible to blind users.

Unfortunately, there are no technologies for displaying refreshable tactile images online. However, it is possible to make a tactile *copy* of a computer picture. With this figure installed on a touch screen or other digitizing tablet a blind user can feel the tactile images, and receive audio feedback from the computer about those images. Tactile copies can be made using swell paper or tactile printers. Swell paper is a special paper that swells when radiant heat is applied to black areas, enabling users to feel images. One source of swell paper and the equipment needed to use it is available from Repro-tronics Inc.: <http://www.repro-tronics.com>

A commercially available tactile printer is the TIGER printer (TactIle Graphics EmbosseR). It is available from ViewPlus Technologies, Inc. More information is available at: <http://www.viewplustech.com>

A touch screen that can be used with tactile graphics is the NOMAD Pad. This touch-sensitive pad connects to the user’s personal computer and allows the user to place raised-line graphics as an overlay on the pad. Once an appropriate file has been selected, the user can touch various points on the graphic and NOMAD will describe them with synthetic speech. TouchBlaster Software is included with NOMAD pads. It provides digitized sound, a simplified menu option and a stand-alone mode that incorporates a built-in Braille alphabet. More information about NOMAD is at: <http://www.rit.edu/~easi/easisem/nomadaph.html>

Off-line printing makes it difficult for blind users to take advantage of computer-supported features like zoom views. A number of online haptic technologies are in development and will find use with virtual reality applications. One such technology, the haptic mouse, has already been introduced by several companies including Immersion Corporation (San Jose, CA, USA) and Control Advancements (Kitchener, Ontario, Canada).

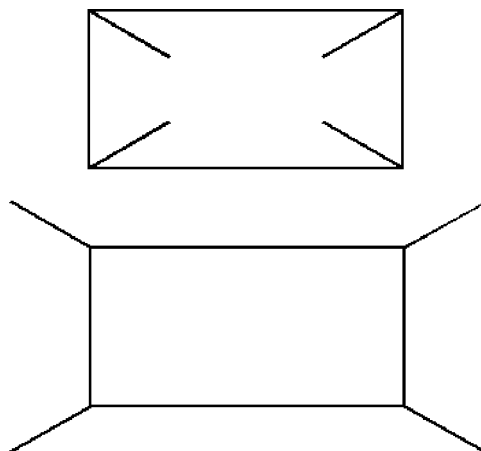
### 11.5.1 Haptic Perception

Haptic perception allows those with visual impairments to comprehend the representational content of an image in the same way that visual perception allows those who can see to infer three-dimensional images from two-dimensional images.

The exact way in which a person learns to “read” a two-dimensional representation of a three-dimensional object depends on environment. Different cultures represent the third dimension differently. In the western tradition, “perspective points” determine the form of representation.



For people who *feel*, rather than *see* 3D representations, the objects become folded out (e.g., a Coke can might be represented as a flat side and two round ends) and perspective is portrayed differently. In this tactile culture, 3Dness is not represented via perspective, as it is for sighted people, but rather via relativity (see the haptic perception image below).



**Figure 1: Two images of rectangles, one with short lines extending inwards from corners of rectangle and the other with short lines extending out from corners.**

These two haptic versions of a four-legged table are two equivalent solutions for flattening a long table. The upper representation is chosen if the table is surrounded by other objects, the lower one if objects stand on the tabletop.

For an interesting paper that discusses this issue visit:

[http://www.inf.fu-berlin.de/~kurze/publications/chi\\_97/mk-hapt.htm](http://www.inf.fu-berlin.de/~kurze/publications/chi_97/mk-hapt.htm)

### 11.5.2 Haptic Image Sources

The National Centre for Tactile Diagrams at the University of Hertfordshire, UK, offers guidelines for making haptic maps and images. Those guidelines and a collection of haptic images can be found at: <http://www.nctd.org.uk/#diagpro>

Another source, the American Printing House for the Blind offers both ready-made graphics covering a variety of subjects and custom-made graphics following user specifications. Visit: <http://www.aph.org>

Finally, anyone can make tactile graphics using a number of common objects such as puff paint, glue, a hot glue gun, drafting tape, yarn, or similar tools.

### 11.5.3 Text Representations of Charts, Diagrams, and Tables

Charts, diagrams, and tables can also be represented in text, using similar techniques to those used for equations (ALT attribute and long description, for example). Text can be generated by content authors who summarize the information provided by the graphic and give a detailed listing of any data.

A tool called PopChart Xpress from Corda Technologies can be used to create accessible Web-ready charts from data copied from a spreadsheet. It is best used to represent charts with static data. Corda also offers PopChart, which is a server application for websites that need to make a large number of charts accessible, or to adapt charts that access data in a database. More information is available from Corda at: <http://www.corda.com>. Examples of interactive, data-driven graphics are available from Corda at: <http://www.corda.com/examples/>

## 11.6 Geography and Maps

### The Problem:

Spatial information, particularly represented as maps, creates a number of obvious problems for users who do not have access to the full range of multimedia including graphics, color differentiation, sound, and so on. There are two kinds of maps available online – those stored as data and those stored as graphical images. Maps comprised as data can be more easily made accessible than those stored graphically.

### Possible solutions for accessible geography and maps:

Most maps are produced from textual information that is held in databases. Currently the best means of representing maps for users with impairments is to print them on a haptic printer that can create different tactile representations according to graphic detail. What a visual learner might see as a change color might be a change in texture on a haptic map.

Data-driven graphical representations of all kinds, including maps, should be rendered as SVG, a W3C recommended language. SVG objects are, as the name implies, scalable; they can be enlarged without deterioration. SVG contains meta-data associated with individual objects and in fact be built out of a set of objects. This means that a composed image, such as a map of a school playground, can describe both its composition as well as the objects used in that composition.

Similarly, developers should attach stylesheets to the objects and even to the relationships between them so that audio, haptic, or other representations can be generated from that data. Using SMIL, voice, captions, descriptions, and other representations can thus be attached to the objects.

Work is being done on making images audible. For more information, see *The Voice - Vision Technology for the Totally Blind* at: <http://www.seeingwithsound.com/voice.htm>

It should be noted that not all blind people can read Braille and Brailles differ. So sound maps are a good alternative. Sound cues are also useful for dyslexic, illiterate, and foreign language learners. For more information, visit: <http://garnet.acns.fsu.edu/~djacobso/haptic/press/press93106.html>

For other resources that outline the benefits of using tactile maps when trying to learn spatial concepts and while learning to move about in space visit: <http://garnet.acns.fsu.edu/~djacobso/haptic/pub/hapticabstracts.html>

And for more information regarding haptic maps and the haptic mouse visit: <http://garnet.acns.fsu.edu/~djacobso/haptic/press/pressksby.html>

## 11.7 Music

### The Problem:

Music notation is a code unto its own and most screen reading software can't translate it to speech.

In the past, only a small number of specialists had the skill to produce Braille music. For example, a blind flutist wishing to participate in the school band would need to send copies of the musical score to a transcriber. Because of the busy schedules of these specialists, it could take weeks or months to translate the score into Braille.

### Some solutions for music accessibility:

Today, using a computerized translator, transcriptions of sheet music can be produced locally in a few hours by sighted copyists with no special training in music Braille. A number of products are making music notation more accessible to those with visual impairments.

- **Dancing Dots' GOODFEEL Braille Music Translator.** With this software package, and the help of several mainstream products, a sighted musician can prepare a Braille score. Blind users can make sound recordings and print Braille editions of their compositions and arrangements. Music scanning software can speed data entry.

Translator software automates transcription of Braille music. Braille scores are produced from the same computer



music files used to print staff notation. GOODFEEL converts MIDI, Lime, and other file formats into music Braille.

For more information about GOODFEEL, visit their website at: [http://www.dancingdots.com/dd\\_desc.htm](http://www.dancingdots.com/dd_desc.htm)

- **The Lime Music Editor.** This software package converts a PC into a blank piece of music staff paper. Sighted customers use Lime to prepare staff notation for GOODFEEL to transcribe. GOODFEEL applies the rules for Braille transcription and automatically embosses the equivalent music in Braille on a Braille printer or a Braille display.
- **Cakewalk.** This software package converts a PC into a multi-track recording studio. With Cakewalk a blind musician can independently prepare MIDI files to be transcribed by GOODFEEL. Cakewalk offers the option to make sound recordings or to print the compositions for sighted players.
- **Musitek's SmartScore.** This music scanning software speeds data entry. Any TWAIN-compatible, desktop scanner can be used to scan sheet music. SmartScore analyzes logical score structure and passes it to Lime or to Cakewalk for further editing.
- **Scoresynth.** This flexible program uses Petri nets (PNs) and algorithms for describing, processing and synthesizing music scores in a manner closer to "music thinking and perception" than can be achieved with conventional music notation. The system employs music objects, which are any simple, complex, or abstract musical entity, at PN nodes to describe information, with transformation rules and causal relationships between entities assigned to PN parameters and structures. Scoresynth algorithms enable the transformation of various parameters of sound and the firing of notes during transitions of the PN. For more information visit: [http://www.informatik.uni-hamburg.de/TGI/pnbib/h/haus\\_g2.html](http://www.informatik.uni-hamburg.de/TGI/pnbib/h/haus_g2.html)
- **Toccata.** This is a Windows program that uses the mouse or keyboard to create music of any complexity. As notes are placed the program's music editor, they are played through the computer's speakers, maintaining the same pitch and duration as the note selected. Toccata can play music back in real-time. A separate window shows the translated and formatted Braille, ready to be sent to a suitable embosser.

Alternatively, using the same software a musician can load an entire music score onto the computer as a MIDI file or as scanned sheet music. Displayed on-screen, the music can then be edited using the mouse. The SharpEye Music Reader program, which is included with Toccata, scans the music, but other programs are also available, such as SmartScore from Musitek.

The Braille output produced by this method can be instantly reformatted in common formats (e.g., Bar-over-Bar, Section-by-Section) and a wide range of optional styles can be incorporated (e.g., Show Hand Signs, Lyrics in Grade II contracted Braille, and so on). It is also possible to edit the Braille directly with six key input.

A video demo of Toccata is available at: <http://members.optusnet.com.au/~terryk/topicframe.htm>

Additional information regarding Toccata is available at: <http://members.optusnet.com.au/~terryk/toccata.htm>

## 11.8 Languages

### The Problem:

The Web is becoming increasingly internationalized. As it does so, language issues present a considerable barrier to access. Switching alphabets often involves much more than a simple change in font. It is not easy to allow for the wide variety of character representations that make up written languages, let alone integrate "texts" that run from left to right, with those that run from right to left, or top to bottom. In addition, many languages are not written using phonetic alphabets at all.

### Possible solutions for language accessibility:

The most effective answer to these problems so far is a work-around that relies on Unicode, coupled with SVG, SMIL, and stylesheets. Unicode is a more complex matrix set than the original ASCII character set developed in the early days of computing. With the involvement of so many languages on the Web today, Unicode provides a way to develop new character sets for languages that have not traditionally been used in computer interfaces. Using Unicode comes at

a price, however. Most legacy software does not understand Unicode, and only a few browsers can cope with it. So there will be a transition period during which Unicode will need specific support and promotion, as well as the development and use of plug-ins.

Layout options are no longer a problem, and the use of Unicode is becoming far more widespread in authoring and access packages.

Basic accessibility requirements expect that a change in the language within a document will be flagged with tags that indicate the name of the language that follows. For more information visit the W3C Internationalization site at: <http://www.w3.org/International/>

There are a number of new standards and practices that, in combination, make it possible to create interesting multi-lingual pages. Language tagging (i.e., indicating a change of language) is an accessibility requirement. Because some of the challenges that arise when working with foreign languages, (e.g., combining several languages within a single Web resource or working with non-English languages) developers should consult the W3C Internationalization help pages at: <http://www.w3.org/International/O-help.html>

### 11.8.1 Ruby Annotation as a Solution for Language Accessibility

Ruby annotation, a W3C recommendation, is a run of text that is associated with another run of text, referred to as the base text. Ruby text is used to provide a short annotation of the associated base text. It is most often used to provide a reading (pronunciation guide). Ruby annotations are used frequently in Japan in many kinds of publications, including books and magazines.

In those cases, ruby texts might appear on either side of the base text. The ruby text that appears before the base text is often meant to indicate reading while the lower text is meant to indicate meaning.

The following is such an example: center base text with two ruby texts above and below. The base text is Japanese Kanji pictograms. The upper ruby text offers a phonetic guide in Japanese Hiragana and the lower text offers a phonetic guide in Latin letters.

しんかんせん ← *ruby text*  
 新幹線 ← *ruby base*  
 shinkansen ← *ruby text 2*

The W3C Ruby specification defines Ruby mark-up to be usable with XHTML, so that ruby text is available on the Web without using special workarounds or graphics. For more information visit: <http://www.w3.org/TR/ruby/>

Ruby accessibility techniques are available at: <http://www.w3.org/TR/ruby/#non-visual>

Documents that contain Ruby mark-up may need to be rendered by non-visual user agents such as voice browsers and braille user agents. For such rendering scenarios, it is important to understand that:

- different ways of rendering may be appropriate, depending on the user and the situation.
- Ruby text that guides pronunciation may need different treatment than Ruby text that contains other information.
- appropriate non-visual rendering requires that the developer indicate the function of each Ruby text item.
- discrepancies can exist between the pronunciation guidelines indicated by the Ruby text and the actual pronunciation.
- users may want more information about the ideographic base text.

## Appendix A – Legal Issues for Accessible Distance Learning

This summary of legal issues in various countries is not intended to be exhaustive or to provide legal guidance. It is instead a resource for finding information about relevant legal requirements in countries where that information is available.

The United Nations has an Overview of International Legal Frameworks for Disability Legislation available at: <http://www.un.org/esa/socdev/enable/disother.htm>

A number of countries have established policies on accessibility of government websites. These policies may or may not include requirements for distance learning services delivered by those governments. A list is maintained by the Web Accessibility Initiative at: <http://www.w3.org/WAI/Policy/>

Several countries have more detailed policies that specifically apply to education as described here:

### Australia

Australia has a Disability Discrimination Act (1992) which applies to education. Draft disability standards have been developed. For information, see: [http://www.hreoc.gov.au/disability\\_rights/education/education.html](http://www.hreoc.gov.au/disability_rights/education/education.html)

In addition, state-by-state regulations for accessibility of government websites are summarized in: <http://www.ozewai.org/presentations/webb-legal.html>

### Canada

Canada has a Charter of Rights and Freedoms that guarantee the basic rights and freedoms important to Canada as a free and democratic society. For more information, see the Guide to the Canadian Charter of Rights and Freedoms: [http://www.pch.gc.ca/ddp-hrd/canada/guide/index\\_e.shtml](http://www.pch.gc.ca/ddp-hrd/canada/guide/index_e.shtml)

The Canadian Government has also established a Common Look and Feel for government websites, which includes accessibility provisions. Details are available at: [http://www.cio-dpi.gc.ca/clf-upe/index\\_e.asp](http://www.cio-dpi.gc.ca/clf-upe/index_e.asp)

In addition, some Canadian provinces have separate disabilities legislation. See especially the Ontarians with Disabilities Act: <http://www.gov.on.ca/MCZCR/english/about/odaintroduction.htm>

### United Kingdom

The Special Educational Needs and Disability Act takes effect on 1 September 2002. The Act removes the previous exemption of education from the Disability Discrimination Act (1995), ensuring that discrimination against disabled students will be unlawful. Institutions will incur additional responsibilities in 2003, with the final sections of legislation coming into effect in 2005.

A summary article (from which this brief introduction was taken) is available at: <http://www.techdis.ac.uk/resources/skill01.html>

Codes of Practice developed by the Disabilities Rights Commission are available from: <http://www.drc-gb.org/drc/InformationAndLegislation/Page34A.asp>

### United States

Policies are now in place or under consideration in several markets and jurisdictions in the United States that make accessibility a requirement for distance learning.

#### U.S. Federal Government Requirements

##### ADA and Section 504

Two Federal laws govern accessibility of education: Title II of the Americans with Disabilities Act (ADA) and Section 504 of the Rehabilitation Act of 1973 (as amended in 1998). All elementary, secondary, and post-secondary educational institutions are regulated under these laws. A useful legal analysis of these requirements is provided in the California Community Colleges' Distance Education Access Guidelines, available at: <http://www.htctu.fhda.edu>

The U.S. Department of Education Office for Civil Rights enforces these laws. Visit:  
<http://www.ed.gov/ocr/disability.html>

### Section 508

In order to ensure that its technology is accessible to its own employees and to the public, the Federal government has created regulations based on Section 508 of the Rehabilitation Act which require that electronic and information technology developed, procured, maintained, or used by the Federal government be accessible to people with disabilities. These regulations apply to all federal purchases of technology. Requirements in Section 508 may also impact state colleges and universities, pending policy decisions from the Department of Education Office of Civil Rights. For official information on Section 508 see:

- The Access Board: <http://www.access-board.gov>
- Federal IT Accessibility Initiative: <http://www.section508.gov>

### **U.S. State Policies**

A partial list of U.S. states with accessibility requirements for government Web pages is at:  
<http://www.resna.org/taproject/policy/initiatives/508/atprojects.html>

Some states have instituted additional requirements, including the following:

#### California Higher Education Requirements

The California Community College system has released Distance Education Access Guidelines and Alternate Media Access Guidelines. The Alternate Media Access guidelines serve as a guide for the implementation of California law AB422 requiring publishers to provide textbooks in electronic format to the three systems of higher education in California (the University of California, the California State University, and the California Community Colleges). The Distance Education Access Guidelines include a summary of legal requirements as well as access guidelines for specific modes of distance education instructional delivery. These documents and other resources are available from:  
<http://www.htctu.fhda.edu>

#### Texas K-12 Textbook Adoptions

Texas has for several years been studying the issue of access to electronic books and educational software for students with disabilities. Two reports, one issued in 1997 and one in 1999, provide information on how educational materials can be made accessible. Texas requires publishers to provide electronic files for adopted print materials and is in the process of incorporating the U.S. Federal Section 508 requirements as an optional part of their adoption process for interactive educational software and electronic textbooks. Texas reports on textbook accessibility is available at:  
<http://www.tsbvi.edu/Education/index.htm#Textbooks>

## Appendix B – Resources

The following is compilation of the websites referenced throughout the document, sorted in order by the section in which they appeared.

### 1. Introduction

Learning Anytime Anywhere Partnerships (LAAP)

<http://www.ed.gov/offices/OPE/FIPSE/LAAP/>

Blackboard

<http://www.blackboard.com/>

CETIS (Centre for Educational Technology Interoperability Standards)

<http://www.cetis.ac.uk/>

DEST (Department of Education, Science, and Training - Australia)

<http://www.dest.gov.au/>

ETS (Educational Testing Service)

<http://www.ets.org/>

IMS Global Learning Consortium

<http://www.imsglobal.org/>

Industry Canada

<http://www.ic.gc.ca/>

NCAM (National Center for Accessible Media)

<http://ncam.wgbh.org/>

The Open University (UK)

<http://www.open.ac.uk/>

Sheffield Hallam University (UK)

<http://www.shu.ac.uk>

University of Toronto ATRC (Adaptive Technology Research Centre)

<http://www.utoronto.ca/atrc/>

WebCT

<http://www.webct.com/>

### 2. Primer on Accessibility

*Accessible Software Guidelines* from the Trace R & D Center

[http://www.trace.wisc.edu/world/computer\\_access/software/](http://www.trace.wisc.edu/world/computer_access/software/)

*Application Software Design Guidelines* from the Trace R&D Center

[http://trace.wisc.edu/docs/software\\_guidelines/toc.htm](http://trace.wisc.edu/docs/software_guidelines/toc.htm)

U.S National Institute of Neurological Disorders and Stroke Dyslexia Information page

[http://www.ninds.nih.gov/health\\_and\\_medical/disorders/dyslexia\\_doc.htm](http://www.ninds.nih.gov/health_and_medical/disorders/dyslexia_doc.htm)

Learning Disabilities Resource Community

<http://www.ldrc.ca/>

Resources on Computers and Learning Disabilities from the Trace R & D Center

[http://trace.wisc.edu/docs/quick\\_sheets/qs17.html](http://trace.wisc.edu/docs/quick_sheets/qs17.html)

Technical Glossary of Adaptive Technologies from the Adaptive Technology Resource Centre at the University of Toronto

<http://www.utoronto.ca/atrc/reference/tech/techgloss.html>

### 3. Principles for Accessibility in Online Distributed Learning

IMS specifications (openly available to the public and available free of charge)

<http://www.imsglobal.org/specifications.html>

*XML Accessibility Guidelines* from the W3C

<http://www.w3.org/TR/xmlgl>

### 4. Using XML for Accessibility

*XML Accessibility Guidelines* from the W3C

<http://www.w3.org/TR/xmlgl>

*Accessibility Features of SVG* from the W3C

<http://www.w3.org/TR/SVG-access/>

SVG Document Structure

<http://www.w3.org/TR/SVG/struct.html>

*Accessibility Features of SMIL* from the W3C

<http://www.w3.org/TR/SMIL-access/>

Media Access Generator (MAGpie) from NCAM

<http://ncam.wgbh.org/webaccess/magpie/>

DAISY (Digital Audio-based Information System)

<http://www.daisy.org>

Open eBook Forum

<http://www.openebook.org/>

### 5. Guidelines for Flexible Media Delivery of Text, Audio, Images, and Multimedia

*User-Friendly Materials and Alternate Formats* from The National Center for the Dissemination of Disability Research

<http://www.ncddr.org/du/products/ufm/ufm.html>

*Guidelines for Producing Instructional and Other Printed Materials in Alternate Media for Persons with Disabilities* from the California Community Colleges

<http://www.htctu.fhda.edu/amguidelines/am33000.htm>

Media Access Generator (MAGpie) from NCAM

<http://ncam.wgbh.org/webaccess/magpie/>

*Tips on Presenting Alternatives to Sound* from CAST (Center for Applied Special Technology)

<http://www.cast.org/udl/index.cfm?i=352#tips>

“SigningAvatar”™ from Vcom3D

<http://www.signingavatar.com>

Making Images Accessible from SNOW (Special Needs Opportunity Windows)

<http://snow.utoronto.ca/access/higher/images.html>

Techniques for Web Content Accessibility Guidelines 1.0 from the W3C (section 2.1.1 - techniques for providing alternatives to auditory and visual content)

<http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#images>

*Use of ALT texts in IMGs* from the Web Design Group

<http://www.htmlhelp.com/feature/art3.htm>

A tutorial from WebAIM, including code examples, illustrating the W3C Web Content Accessibility Guidelines 1.0 recommendation checkpoint for providing “equivalent alternatives to auditory and visual content”

<http://www.webaim.org/tutorials/alt>

*Scalable Vector Graphics (SVG) Specification* from the W3C

<http://www.w3.org/TR/SVG/>

*SVG Accessibility Features* from the W3C

<http://www.w3.org/TR/SVG-access/>

Rich Media Accessibility Resource Center from NCAM

<http://ncam.wgbh.org/richmedia/>

A tutorial on Creating Audio Descriptions for Rich Media from NCAM

<http://ncam.wgbh.org/richmedia/audiodesc.html>

A tutorial on Creating Captions for Rich Media from NCAM

<http://ncam.wgbh.org/richmedia/captioning.html>

## **6. Guidelines for Developing Asynchronous Communication and Collaboration Tools**

*Web Content Accessibility Guidelines* from the W3C

<http://www.w3.org/TR/WCAG10/>

Examples of accessible threaded message boards from WorldEnable Internet Accessibility Discussion

<http://www.worldenable.net/iadiscuss/>

*Authoring Tool Accessibility Guidelines* from the W3C

<http://www.w3.org/TR/WAI-AUTOOLS/>

Examples of accessible document repositories from WAI Mail Archives

<http://lists.w3.org/Archives/Public/w3c-wai-ig/>

WAVE accessibility evaluation tool

[http://www.temple.edu/inst\\_disabilities/piat/wave/](http://www.temple.edu/inst_disabilities/piat/wave/)

The University of Illinois' Microsoft Power Point WWW Publishing Accessibility Wizard

<http://www.rehab.uiuc.edu/ppt/>

W3C Slidemaker Tool

<http://dev.w3.org/cvsweb/slidemaker/>

ArsDigita WimpyPoint

[http://www.arsdigita.com/acs-repository/one-version?version\\_id=1114](http://www.arsdigita.com/acs-repository/one-version?version_id=1114)

## **7. Guidelines for Developing Synchronous Communication and Collaboration Tools**

CAST (Center for Applied Special Technologies)

<http://www.cast.org/>

EASI (Equal Access to Software and Information)

<http://www.rit.edu/~easi/>

SNOW (Special Needs Opportunity Windows)

<http://snow.utoronto.ca>

University of Washington DO-IT Program (Disabilities, Opportunities, Internetworking, and Technology)

<http://www.washington.edu/doiit/>

NCAM (The CPB/WGBH National Center for Accessible Media)

<http://ncam.wgbh.org/>

## 8. Guidelines for Developing Accessible Interfaces and Interactive Environments

*Application Software Design* Guidelines from the Trace R&D Center

[http://trace.wisc.edu/docs/software\\_guidelines/toc.htm](http://trace.wisc.edu/docs/software_guidelines/toc.htm)

*Making Educational Software Accessible: Design Guidelines Including Math and Science Solutions* from NCAM

<http://ncam.wgbh.org/cdrom/guideline/>

Prototypes of accessible educational software are from NCAM (a talking math game and a talking science simulation)

<http://ncam.wgbh.org/cdrom/guideline/download.html>

Digital Field Trip to the Rainforest AT Version from Digital Frog International

<http://www.digitalfrog.com/products/rainforest.html>

Technical details on how to create talking DVD menus from NCAM

<http://ncam.wgbh.org/convergence/>

*Abraham and Mary Lincoln: A House Divided* an accessible DVD, includes talking menus for access to navigation by blind viewers and access features for the program for both blind and deaf viewers

<http://www.shop.pbs.org>

Set-Top Box Accessibility information and a prototype Electronic Program Guide from NCAM

<http://ncam.wgbh.org/convergence/>

Information about accessible kiosks and touch screen interfaces from EZ™ Access at the Trace R & D Center

<http://trace.wisc.edu/world/kiosks/ez/>

V2 Support Project at the Trace R & D Center

<http://trace.wisc.edu/world/v2/index.htm>

Microsoft Accessibility Home Page

<http://www.microsoft.com/enable/>

Microsoft Accessibility Information for Developers

<http://www.msdn.microsoft.com/at/>

Apple Education Disability Resources

<http://www.apple.com/education/k12/disability/>

*Macintosh Human Interface Guidelines* Book

<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-2.html#avail1-0>

Universal Access section of *Macintosh Human Interface Guidelines* Book

<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-40.html#MARKER-9-55>

Human Interface Design Principles section of *Macintosh Human Interface Guidelines* Book

<http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-16.html#HEADING16-0>

Alva Access Group (developers of the outSPOKEN screen reader and inLARGE screen magnifier, Alva's time-based demos of their products can be downloaded for testing purposes)

<http://www.aagi.com>

W3C Web Access Initiative (WAI)

<http://www.w3.org/wai/>

Sun Microsystems' Accessibility Program - Developer Information

<http://www.sun.com/access/developers/>

The Java™ Access Bridge

<http://www.java.sun.com/products/accessbridge/>



Java™ Accessibility Helper (follow the link from the Sun Microsystems Accessibility Program website)

<http://www.sun.com/access/>

The Java™ Tutorial - Accessibility Section

<http://java.sun.com/docs/books/tutorial/uiswing/misc/>

*IBM Guidelines for Writing Accessible Applications using 100% Pure Java™*

<http://www-3.ibm.com/able/snsjavag.html>

List of accessibility techniques for specific development environments from the W3C

<http://www.w3.org/TR/ATAG10-TECHS/#check-use-system-conventions>

## 9. Guidelines for Testing and Assessment

British Standards Institution (BSI) BS 7988 A Code of Practice for the use of information technology for the delivery of assessments

<http://edd.bsi.org.uk/link.php3/ist/43>

## 10. Guidelines for Developing Accessible Authoring Tools

*Authoring Tool Accessibility Guidelines* from the W3C

<http://www.w3.org/TR/WAI-AUTOOLS/>

Authoring Tool Accessibility Guidelines Working Group of the W3C

<http://www.w3.org/WAI/AU/>

A review of common courseware authoring tool accessibility issues and solutions from SNOW

<http://snow.utoronto.ca/access/courseware/index.html>

## 11. Guidelines for Topic Specific Access

Description of Mathspeak

<http://www.rit.edu/~easi/easisem/talkmath.htm>

PIVoT Web site

<http://curricula2.mit.edu/pivot/>

W3C's Math Home Page (provides MathML information)

<http://www.w3.org/Math/>

Amaya Web editor

<http://www.w3.org/Amaya/>

AsTeR website

<http://www.cs.cornell.edu/Info/People/raman/aster/aster-toplevel.html>

American Scientist article *Speaking of Mathematics*

<http://www.amsci.org/amsci/issues/Comsci96/compsci96-03.html>

EzMath is being upgraded and will soon be open source on SourceForge (a website repository of open source code and applications)

<http://sourceforge.net/>

IBM TechExplorer Hypermedia Browser

<http://www.software.ibm.com/techexplorer/>

MathType

<http://www.dessci.com/webmath/>

The Accessible Graphing Calculator from Oregon State University's Science Access Project  
<http://dots.physics.orst.edu/agc.html>

Braille 'n' Speak notetaker with calculator  
[http://www.freedomscientific.com/fs\\_products/notetakers\\_bns.asp](http://www.freedomscientific.com/fs_products/notetakers_bns.asp)

Graphit graphing calculator software for the Braille 'n' Speak  
[http://www.freedomscientific.com/fs\\_products/notetakers\\_related.asp#graphit](http://www.freedomscientific.com/fs_products/notetakers_related.asp#graphit)

Accommodating Students with Disabilities in Math and Science Classes from the University of Washington's DO-IT program  
<http://www.washington.edu/doi/MathSci/>

*Chemical Markup Language. A Position Paper*  
<http://www.xml-cml.org/information/position.html>

Barrier Free Education's Accessible Science Experiments in Chemistry and Physics  
<http://barrier-free.arch.gatech.edu/Lab/index.html>

Haptics Periodic Table from NIDE (Network for Inclusive Distance Education)  
<http://nide.snow.utoronto.ca/hperiodic.htm>

The American Chemical Society (ACS) Guidelines for "Teaching Chemistry to Students with Disabilities"  
<http://membership.acs.org/C/CWD/teaching/start.htm>

The American Chemical Society (ACS) Committee on Chemists with Disabilities "Working Chemists with Disabilities: Expanding Opportunities in Science"  
<http://membership.acs.org/C/CWD/workchem/start.htm>

PIVoT (Physics Interactive Video Tutor)  
<http://ncam.wgbh.org/webaccess/pivot/>  
[http://caes.mit.edu/research/pivot/report\\_2000-10.html](http://caes.mit.edu/research/pivot/report_2000-10.html)  
<http://curricula2.mit.edu/pivot>

American Physical Society: Physics and Persons with Disabilities  
<http://www.aps.org/memb/disabilities.html>

Electric Chemistry Set  
<http://dbweb.liv.ac.uk/ltsnpsc/softfinx.asp>

Immersion Medical simulators  
<http://www.immersion.com/products/overview.shtml>

Phantom Haptics device  
<http://www.sensable.com/haptics/haptics.html>

Barrier Free Education's Biology Simulations  
[http://barrier-free.arch.gatech.edu/Lab/sim\\_bio.html#interact\\_frog](http://barrier-free.arch.gatech.edu/Lab/sim_bio.html#interact_frog)

Barrier Free Education's Physics Simulations  
[http://barrier-free.arch.gatech.edu/Lab/sim\\_physics.html](http://barrier-free.arch.gatech.edu/Lab/sim_physics.html)

Practical Experimentation by Accessible Remote Learning (PEARL)  
<http://kmi.open.ac.uk/projects/pearl/summary/index.htm>

Repro-Tronics (tactile imaging paper)  
<http://www.repro-tronics.com>

ViewPlus Technologies  
<http://www.viewplustech.com>

NOMAD Talking Touch Pad

<http://www.rit.edu/~easi/easisem/nomadaph.html>

Rendering Drawings for Interactive Haptic Perception

[http://www.inf.fu-berlin.de/~kurze/publications/chi\\_97/mk-hapt.htm](http://www.inf.fu-berlin.de/~kurze/publications/chi_97/mk-hapt.htm)

National Centre for Tactile Diagrams (information about making haptic maps and images)

<http://www.nctd.org.uk/#diagpro>

American Printing House for the Blind

<http://www.aph.org>

Corda Technologies

<http://www.corda.com/>

<http://www.corda.com/examples/>

The Voice - Vision Technology for the Totally Blind

<http://www.seeingwithsound.com/voice.htm>

Article from Haptic Soundscapes about auditory maps

<http://garnet.acns.fsu.edu/~djacobso/haptic/press/press93106.html>

A series of papers covering the benefits of using tactile maps when trying to learn spatial concepts and to move about in space

<http://garnet.acns.fsu.edu/~djacobso/haptic/pub/hapticabstracts.html>

Article: "Finding direction by sound"

<http://garnet.acns.fsu.edu/~djacobso/haptic/press/pressksby.html>

Dancing Dots' GOODFEEL Braille Music Translator

[http://www.dancingdots.com/dd\\_desc.htm](http://www.dancingdots.com/dd_desc.htm)

Scoresynth: A System for the Synthesis of Music Scores Based on Petri Nets and a Music Algebra

[http://www.informatik.uni-hamburg.de/TGI/pnbib/h/haus\\_g2.html](http://www.informatik.uni-hamburg.de/TGI/pnbib/h/haus_g2.html)

Video demo of Toccata Braille Music Software

<http://members.optusnet.com.au/~terryk/topicframe.htm>

Toccata Braille Music Software

<http://members.optusnet.com.au/~terryk/toccata.htm>

W3C Internationalization / Localization website

<http://www.w3.org/International/>

Help with creating a multilingual site from the W3C Internationalization / Localization website

<http://www.w3.org/International/O-help.html>

Ruby Annotation from the W3C

<http://www.w3.org/TR/ruby/>

Ruby techniques for non-visual rendering from the W3C

<http://www.w3.org/TR/ruby/#non-visual>

## About This Document

Title	IMS Guidelines for Developing Accessible Learning Applications
Editors	Cathleen Barstow, Mark McKell, Madeleine Rothberg, Chris Schmidt
Version	1.0
Version Date	27 June 2002
Status	<b>White Paper</b>
Summary	This document presents the IMS Guidelines for Developing Accessible Learning Applications.
Revision Information	27 June 2002
Purpose	Describes the guidelines for creating or retrofitting learning applications to meet the demands of accessibility.
Document Location	<a href="http://www.msglobal.org/accessibility/accv1p0/imsacc_guidev1p0.html">http://www.msglobal.org/accessibility/accv1p0/imsacc_guidev1p0.html</a>

## List of Contributors

The following individuals contributed to the development of this document:

<b>Name</b>	<b>Organization</b>	<b>Country</b>
Cathleen Barstow	NCAM	USA
Reidy Brown	Blackboard, Inc.	USA
Chetz Colwell	Open University	UK
Martyn Cooper	Open University	UK
Eric Hansen	Educational Testing Service	USA
Andy Heath	UK eUniversities Worldwide and Sheffield Hallam University	UK
Hazel Kennedy	Open University	UK
Liddy Nevile	DEST	Australia
Mark Norton	IMS Global Learning Consortium, Inc.	USA
Jan Richards	Industry Canada and University of Toronto ATRC	Canada
Madeleine Rothberg	NCAM	USA
Jutta Treviranus	Industry Canada and University of Toronto ATRC	Canada

## Revision History

Version No.	Release Date	Comments
0.5	12 October 2001	The first released version of the accessibility guidelines.
0.6	19 October 2001	a) Added a section about synchronous communication and collaboration tools. b) Made many minor edits.
0.9	12 February 2002	Internal working group draft. a) Reorganized and expanded some material and made extensive edits. b) Added new sections about using XML for accessibility and accessibility information for operating systems and platforms.
1.0	27 June 2002	Edited for consistency.

# Index

## A

Alternative access 13  
 ASL 24, 25  
 Audio 6, 9, 13, 23, 24, 25, 26, 29, 37, 50, 51, 56

## B

Blind 6, 8, 9, 10, 11, 13, 16, 25, 37, 38, 51, 54, 57  
 Braille 8, 12, 13, 14, 23, 24, 51, 54, 56, 57, 58

## C

ChemML 52  
 Cognitive Disabilities 10, 15

## D

Deaf 6, 9, 15, 16, 37, 38  
 Deaf-blind 12, 14, 16, 37

## E

Email 27, 28  
 Equivalent access 13, 15

## F

FIPSE 7

## H

Haptic 53, 54, 55  
 haptic 21, 52, 54, 55, 56

## I

IMS 5, 15, 16, 17  
 Internet 28, 53

## K

Keyboards 12, 13, 14, 16

## L

LAAP 7  
 Language 10, 57, 58  
 Latex 50  
 Learning Disabilities 10, 11  
 Low-vision 6, 9, 12, 13, 15, 25, 28

## M

Magnifiers 12, 16  
 Mathematics 48, 49, 50, 51  
 MathML 49, 50, 51, 52  
 Message Boards 27  
 Music 56, 57

## N

NCAM 25, 37, 38

## P

Physical Disabilities 10, 12, 13, 38

## S

Science 37, 51  
 Screen readers 6, 8, 11, 14, 16, 24  
 Single switches 13, 16  
 SMIL 21, 56, 57  
 SVG 20, 25, 48, 49, 52, 56, 57

## T

Tactile 5, 23, 24, 51, 53, 54, 55, 56  
 Trace Center 8, 38

## U

Universal Design 5, 50

## V

Visual 5, 9, 15, 16, 24, 26, 38, 49, 54  
 Voice recognition 12, 14, 16

## W

W3C 23, 25, 47, 49, 50, 58  
 WAI 27, 28

## X

XHTML 50, 52, 58  
 XML 25, 49, 50, 52

*IMS Global Learning Consortium, Inc. (“IMS”) is publishing the information contained in this IMS Guidelines for Developing Accessible Learning Applications (“White Paper”) for purposes of scientific, experimental, and scholarly collaboration only.*

*IMS makes no warranty or representation regarding the accuracy or completeness of the White Paper.*

*This material is provided on an “As Is” and “As Available” basis.*

*The White Paper is at all times subject to change and revision without notice.*

*It is your sole responsibility to evaluate the usefulness, accuracy, and completeness of the White Paper as it relates to you.*

*IMS would appreciate receiving your comments and suggestions.*

*Please contact IMS through our website at <http://www.imsglobal.org>*

*Please refer to Document Name: IMS Guidelines for Developing Accessible Learning Applications*

*Date: 27 June 2002*