Accessible and Adaptive e-Learning

Greg Gay

Adaptive Technology Resource Centre – University of Toronto

g.gay@utoronto.ca

Abstract

The University of Toronto's Adaptive Technology Resource Centre (ATRC) conducted a technical audit and user study that examined the support being provided by popular online learning management systems for students with disabilities. Both studies revealed that none of the current systems was inclusive to all potential learners. What evolved out of the results of these studies was the ATutor Learning Content Management System, which was developed with the guiding philosophy of «accessibility and adaptability» for all. The development team's initial focus was to develop a system that was accessible to all potential users regardless of the technologies they might be using to browse the Web. We were also intent on creating a system that would adapt to the learning needs or styles of individual learners, and providing a teaching environment that could be adapted to the pedagogical preferences of the instructors using it. Lastly, the team focused on making the system highly customizable, so groups using ATutor could easily modify the look and feel of the learning environment. ATutor continues to evolve and expand. Many related secondary projects are underway to extend its functionality and its community of users is rapidly growing.

Edizioni Erickson – Trento
www.erickson.it

Je-LKS Journal of e-Learning and Knowledge Society — *Vol. 2, n. 1, marzo 2006 (pp. 35-46)*

1. Introduction

The ATRC completed two studies that evaluated the accessibility of online learning environments for people with disabilities, and, more specifically, those using adaptive technologies to access the Web.

First, the «Courseware Accessibility Study» (Gay, Harrison, Richards, and Treviranus, 1999) was a technical audit that assessed the compliance of seven popular commercial *Learning Management Systems* (LMS) with the W3C Web Content Accessibility Guidelines (WCAG 1.0, Chisholm, Vanderheiden, & Jacobs 1999).

Second, the «Inclusion in an Electronic Classroom» (Gay, Landon, and Treviranus, 2000) study recruited learners with various disabilities to participate in an online disability advocacy course of six weekly lessons each presented in a different LMS. An observer accompanied each student during the lessons and recorded any difficulties they had with standard sets of activities (e.g. reading content, posting forum message, adding resource links, etc.).

In both studies it was found that existing learning management systems did not provide full support for students with disabilities. While some systems fared better than others, none allowed students across all disability groups to participate in every learning activity offered by a system.

At the time of these studies public awareness regarding Web accessibility issues was minimal. The typical LMS developer had no concept of accessibility, or the inclusion of students with disabilities in online courses. Since then, public awareness has improved considerably, and most LMS developers are now addressing issues of inclusion. In most cases, however, developers are retrofitting existing systems to make them more accessible. While this may be the only option on mature systems, it can be costly and time consuming. Addressing accessibility from the outset is a more efficient and cost-effective approach.

As a result of the LMS studies, the ATRC embarked upon a project to develop a fully accessible LMS, with accessibility built into the system. The ATutor *Learning Content Management System* (LCMS) is the outcome of that project.

Most readers will be familiar with the LMS classification, while LCMS is a relative newcomer to the realm of e-Learning. The primary difference between the two is in the handling of content: an LCMS is more of a content authoring/management tool, while the LMS is primarily a course, enrollment, results management tool. In reality, however, the boundary between the two has become blurred, with most systems now adopting both course and content management tools. ATutor includes most of the tools found in an LMS, in addition to its collection of content management tools.

The description most often associated with ATutor is «an open source LCMS designed with accessibility and adaptability in mind». *Accessibility* and *Adaptability* were the primary goals in the development of ATutor.

2. LMS Accessibility

The first major focus in the creation of ATutor was to ensure the system was accessible to people with disabilities. While discussing the technical aspects of authoring accessible HTML might be useful, such tutorials are plentiful on the Web, so the focus of this section will be on usability and making a learning environment fully accessible. While it is possible to achieve technical compliance with accessibility guidelines, it is also possible to be technically compliant but have a system that is not usable.

By far, blind users face the most barriers when learning online, and much of the following discussion will make reference to them. A blind person will likely use a screen reader to access the Web. A screen reader essentially extracts the text and surrounding structure from a Web page or an application, and reads it aloud. Screen readers usually include a variety of utilities for such things as extracting a list of links or the headings on a page, or listing the frames present, making it easier to navigate within Web content by utilizing the structural elements within the HTML markup.

The strategies that follow have all been implemented in ATutor, and should be a part of any such system to ensure full accessibility to all users.

2.1 Visual Content

Perhaps the most obvious barrier for blind users is graphical content, such as images, videos, and other types of multimedia that do not include *alternative formats*. In such cases a text description, transcript, or descriptive audio may be the only means through which blind students, or others with disabilities, are able to access learning content. That said, the LMS itself is likely only to require text descriptions via the HTML Alt attribute for images used within the interface. Other alternative formats are more applicable to the content that might be presented in the LMS. Where a system does allow the creation of multimedia content, a means must be provided to allow the author to add text alternatives for what would otherwise be inaccessible content to blind users.

While it may be the responsibility of the external authoring tools to provide a means of including alternative formats, tools such as the AChecker (Ridpath, 2004) Web service can evaluate content accessibility as it is being assembled in an LMS.

2.2 Complexity of the Interface

Complexity is another barrier, not only faced by blind users, but also by those with cognitive disabilities, or even novice users who may be unfamiliar with the Web or online learning systems. As complexity increases a learning environment

becomes more difficult to use, taxing cognitive capacities that should be devoted to content learning. The ability to simplify the environment will improve usability for all users, not just those with disabilities. To simplify the learning environment, an LMS must allow users and instructors to *minimize information* presented on the screen at any time. In addition to minimizing information, an *intuitive navigation* system must also exist allowing users to move between screens or features without requiring too much thought. All significant features should be accessible within one or two mouse clicks.

2.3 Within Screen Navigation

The ability to navigate within screens of an LMS can greatly improve usability for screen reader users. The most common navigation element of this type is frequently referred to as the *navigation bypass* link. It is often an invisible 1 pixel by 1 pixel image located in the top left corner of the screen, linked to an anchor at the start of the primary content appearing further down the page. The Alt text associated with the image might read «Bypass navigation and jump to content» or something to that effect. When blind users navigate into a page they can select the bypass link and reposition the cursor at the beginning of the content area. Without such a link, screen reader users may be forced to listen to the navigation links being repeated over and over for each page before reaching the relevant instructional content below. In many cases there are dozens of links at the top of each screen, which can add considerably to the amount of time it takes blind students to navigate through course content.

Bypass links can also be used to jump to a navigation bar or to a table of contents, but such links should be restricted to key elements within the learning environment. Bypass links might also be used to skip over data tables, embedded multimedia objects, an alphabetic list, or any other repetitive or complex content.

2.4 Data Tables

While data tables might include a bypass link enabling screen reader users to skip past the table, in many cases these learners will want to listen to the content of the table, as well. As data tables become larger they become increasingly more difficult to comprehend, requiring blind users to remember where they are within the cells of the table. It also becomes more difficult to make associations between the content in a data cell and the heading for the column or row in which the data appears. Without some form of explicit association between header titles and data cells, most screen reader users will become disoriented. The solution is generally quite simple; including the «scope» attribute (i.e. scope=«col») in the cells of the header column or row causes the screen reader to announce the header titles when-

ever the data cells are read, thus easing the strain on a user's memory. Multi-level header rows in data tables can also be marked up to associate the header labels with specific data cells using the «headers» and «id» attributes within every data cell. This technique, however, is more complex than the one previously described.

2.5 Web Forms

Web forms can also present barriers for some users. The easiest way to ensure that forms are accessible to all users is to explicitly label form fields using the HTML Label element. This ensures that form fields do not become separated from their text labels if the screen is resized, for example. This technique also enables users to activate the respective field by clicking on the label. Being able to click on a label can be important for those with mobility impairments who might have difficulty positioning a mouse pointer over a tiny form element such as a radio button or checkbox. Another way of making Web forms more accessible is to group related content (i.e. all personal information is grouped together, all address information, all student information, and so on). Grouping related fields also helps to avoid overtaxing a user's memory. Required fields within a form should also be grouped together and labeled as such so that users do not have to guess whether the information is mandatory or not. The required field label should be defined before the beginning of a form rather than after it.

2.6 Functional Feedback

Imagine for a moment that a blind user has just submitted the registration form described above, but omitted a required field within the form. The page reloads after submitting the form, and the screen reader is repositioned at the default top left corner of the page and begins reading. The first question that comes to mind for this users is «Did the form submit successfully?». This person could navigate through the page to see if there were any error messages generated, and if not make the assumption that the form submission was successful. Finding error messages — if present — requires both time and effort. If a navigation bypass link is available, and the error messages consistently appear immediately after the anchor, discovering such messages can be relatively easy. If, on the other hand, error messages appear beside the missing fields, it will take users longer to find them, if they are found at all. If no error messages are found, a user can sometimes assume the form was successfully submitted, but a sense of doubt will remain. Consistent use of both positive (success) and negative (error) feedback is very important for improving usability, particularly for blind users, but also for those with cognitive disabilities. It can also be useful to suggest a course of action in feedback messages. For example, feedback provided after a registration form has been successfully submitted might suggest logging into the system, and provide a link to the login

screen. If an error occurred, a direct link from the message to the location of the error can greatly speed up navigation for users with disabilities.

2.7 Consistent Presentation

Consistent presentation of features within an LMS is also important in order to improve usability and accessibility for screen reader users. When a blind user initially enters the LMS he or she will generally attempt to imagine the layout of the environment, and develop a mental picture of where features are located on the screen. Once a mental picture has been memorized, blind users can usually find their way around the environment much like anyone else, assuming the environment remains consistent. When features of the environment change from page to page, a blind user will often become disoriented and lose focus on the learning that should be taking place within the context of the course content.

There are times, however, when inconsistencies can be useful — to highlight changes such as the availability of a test, for example. If a «Tests» tool suddenly appears in the interface where one did not exist before, the user is likely to stop and consider this new information before proceeding.

With regard to feedback described above, it is important to position the feedback messages in the same place on the screen for each page, preferably at the top of the content being presented, and consistently after specific types of actions, such as submitting a form.

2.8 Structured Content

While less associated with the LMS itself, well-structured content can greatly improve usability for screen reader users. In most cases this means using properly nested heading markup to section content within a larger document. As mentioned earlier, screen readers will include a utility to extract the headings on a page (assuming they are properly marked up using the HTML elements H1, H2, H3 and so on). This feature presents a quick summary of the content on the page, and also provides quick navigation through the various sections within a page. If the section of interest is at the bottom of the screen, choosing the last heading listed will take the user directly to that section. If headings are not marked up with heading elements, the page will appear as a single block of text. To reach the last section within a page, blind users would be required to navigate through all text that appeared before it.

3. LMS Adaptability

Another major focus in the creation of ATutor was the system's adaptability for learners, instructors, and administrators. This section describes the adaptability

of ATutor. The adaptive strategies described below have all been implemented in ATutor, and should be a part of any such system to ensure that full participation is possible for all users.

3.1 Student Adaptability

In addition to being able to adapt an LMS to function effectively with assistive technologies, from the perspective of cognitive science, a system should also adapt to the learning tendencies of individual learners. While human learning is highly complex and much about it is still hypothetical, there are a few so-called «givens» in our understanding of how humans learn. In an attempt to create a system that adapts to learning tendencies, a simplified six-point model was used to provide a structure for creating the system. Three perceptual tendencies, and three structural tendencies make up the model. These six points do not represent all possible styles, abilities, or representations of knowledge, but do provide a good starting point from which to design a system based on a theoretical understanding of human cognition. In most cases learners will exhibit all of the perceptual and structural styles mentioned, though often they will prefer some combinations to others.

A six-point model for designing an adaptive learning system:

- Perceptual learning and processing styles:
  1. Verbal
  2. Visual
  3. Kinesthetic
- Structural representations of knowledge:
  1. Sequential
  2. Hierarchical
  3. Global

3.1.1 Perceptual Styles

Perceptual styles are accommodated in ATutor by providing learners with multiple means of perceiving the learning environment. By default, all features accommodate *verbal preferences* (i.e. reading text). Most learning management systems are verbally oriented, accommodating those whose preference is written content. With traditional pedagogical approaches, most classroom learning is also geared toward verbal learners. In online learning environments, the opportunities for providing alternatives to verbal information is much greater for those who prefer other means of learning (i.e. seeing, hearing, doing) as their primary means of absorbing new information.

During the writing of this article, the ATalker add-on for ATutor was being developed to provide an audio enhanced learning experience for those who learn

better by hearing content, perhaps in addition to reading it. Students themselves are able to use ATalker to convert text content into either a WAV, OGG or MP3 audio file that they can listen to in a typical multimedia player. Instructors can also use ATalker to generate audio-based content and, using the SABLE (Bell Labs, 1999), or SSML (Synthesized Speech Markup Language) (W3C, 2004) markup languages, create high quality synthetic audio dialogs to accompany their text-based content. Administrators can generate a wide range of voices for the ATutor system that play when a mouse pointer hovers over an interface element or when a feedback message is generated. To accommodate *visual learning preferences*, student tools generally include a visual representation (i.e. an icon) so those who would rather navigate via graphical links can do so, minimizing the need to read text. Visual representations are helpful for those with learning disabilities or other cognitive disabilities, but they are also useful to those whose primary language differs from the learning environment in which the content is being presented. The interface is also presented in a consistent manner so that visual learners can create a mental image or map of the interface and locate tools through visualization. To accommodate *kinesthetic preferences,* the environment is fully accessible via keyboard. The system will function effectively without a mouse, though a combination of keyboard and mouse is generally the most efficient way to navigate through ATutor for those with kinesthetic talents. The HTML Accesskey attribute is included with many features making it possible to learn a set of standard keystrokes to control the ATutor environment.

3.1.2 Structural Styles

Structural styles are accommodated in ATutor by providing learners with multiple means of navigating through content.

Sequential links allow learners to navigate through content in order or, if they prefer, enable them to include additional structure where content becomes more difficult.

Hierarchical content structures are provided for those who prefer to move up and down through topics and sub-topics. Breadcrumb links (i.e. a string of links that enable users to retrace their steps through visited pages), a table of contents for each content section, and hierarchical visual presentations of content structure are displayed in the Sitemap and in the Content Navigation menu. Hierarchical representations can also be enhanced with optional topic numbering (e.g. 1.2, 1.2.1, 1.2.1.1.) which also helps represent the hierarchical structure between topics and sub-topics for those with numeric talents.

Global content structures are also provided for those who prefer to move around within a content area through related topics, and for those who prefer to develop a «big picture» of a content area before they tackle the finer details. The Sitemap and the Content Navigation menu allow learners to view all content topics at once,

and move through topics in any order they choose. The Content Navigation menu also allows learners to limit the size of the global menu display, focusing on smaller global units (such as the topics within a single learning module, for example).

3.2 Instructor Adaptability

Instructors have the same perceptual and structural elements available to them as learners do. The focus on adaptability for instructors, however, is more toward providing a broad collection of tools to allow them to apply any combination of pedagogical approaches.

The following describes only four of the myriad of pedagogical approaches available to instructors, and how they might be accommodated in an online learning environment. Traditional and behaviourist approaches are discussed briefly, with additional details and examples provided for more current cognitivist and constructivist approaches.

3.2.1 Traditional Transmission

The traditional transmission model of teaching and learning has fallen out of favour in recent years, as science has come to understand that such educational instruction is generally less effective than most other teaching methodologies. Nonetheless, ATutor can be used to transmit knowledge (i.e. post content) and test learners' assimilation of that knowledge through quizzes or tests.

3.2.2 Behaviourist Approach

The behaviourist pedagogical model (which is similar to the transmission model) has also lost some of its appeal as cognitive theories began to take hold in the 1980s. Behaviorist approaches focus on delivering subject matter through sequencing, reinforcement, and learner control. The SCORM (Sharable Content Object Reference Model) utilities (ADL, 2001) found in ATutor support such a behaviourist model, providing sequences of learning content controlled by the learner, based on behavioural outcomes (i.e. answers to test questions). Learners review or experience subject matter, test their knowledge of that subject matter, receive feedback about their progress, and are directed to the next topic in the sequences of topics based on their test outcomes. For example, a subsequent or advanced topic may be available if a test is passed, while a remedial module or review of the topic might appear if a test is failed.

3.2.3 Cognitivist Learning

As a Web-based learning environment, ATutor is built primarily on a cognitivist model, with an interface based on an understanding of how humans process

information, and how that information is stored in memory. By default, ATutor is presented in visual, verbal, and kinesthetic forms, while learning content within it is presented in sequential, hierarchical, and global forms.

Cognitivist approaches encompass many concepts including discovery learning, scaffolding, instructor as coach, problem-based instruction, learner control, cooperative learning, metacognitive devices, and active engagement, among others (Deubel, 2003). All of these concepts are based in cognitive theory and with an understanding of the processes that are occurring in the mind of a learner.

3.2.4 Constructivist Pedagogy

Constructivist approaches have become popular in the past decade. Constructivist pedagogy essentially includes teaching and learning strategies in which learners construct knowledge, generally as a group or social activity. Much of constructivist theory is based on cognitivist approaches with added emphasis on the construction of knowledge and the social aspects of learning. Essentially, learners are given a task and asked to work together to fully define the problem. They gather resources, and assign duties to group members who proceed with their individual tasks constructing their piece of knowledge. The group then comes together and assembles the various pieces of knowledge into a complete unit that thoroughly addresses the assigned problem.

In ATutor there are many ways in which to support a constructivist approach to teaching and learning. Tools such as the Link Database, TILE repository, and course glossary can be used to gather resources. The tools in the ACollab group collaboration add-on can also be used to gather resources, and for cooperative document authoring, allowing group members to take on sub-tasks in creating a group paper or project.

The EWiki add-on for ATutor can be used for entire class knowledge-building activities. Instructors can begin the activity by defining a problem. Students then branch from the main problem space to create their own sub-topics, gather resources and documenting their findings to share with classmates. As resources and documentation materials are gathered, topics are linked together to form a web of knowledge all relating back to the initial problem.

A variety of communication tools are also available within ATutor to accommodate the social aspects of learning. Both synchronous and asynchronous tools allow students and teachers to communicate in real time, or at any time it is convenient for the user. Discussion forums and chats can be limited to a group of students, an entire course, extended across courses, or even across all the courses on an ATutor system to create a community discussion place. Tools like AComm, a Java-based white board and chat application, can also be used for real-time communication activities, while the Users Online tool allows course members to communicate directly with others currently logged into the system.

3.3 System Adaptability

At the system level, administrators also have a variety of customizations available. ATutor uses a cascading template system that separates ATutor's appearance (themes) from its functionality (programming). It is relatively easy to create a new look for an ATutor installation with just a basic understanding of HTML. A new theme can often be created in a matter of minutes by copying and adjusting the style sheet from the default theme, and perhaps modifying the header and footer files. Characteristics of the new theme cascade over top of the default theme, leaving intact the features of the original look-and-feel that were not modified.

ATutor is an international system that can accommodate any number of languages. Once a language is installed, the administrator can modify it through the system's Language Manager, making it relatively easy to customize terminology. Where a language is not yet available, administrators can translate the new language from within ATutor, and export the language pack to share with others.

Created using the interpreted programming language PHP, the ATutor system is also platform independent, allowing it to run on any Unix/Linux-, Windows-, or Macintosh-based server that has the PHP interpreter installed.

4. Conclusion

ATutor is still in its infancy, with an initial release in December of 2002. It has, however, gained wide recognition as a viable open source alternative to commercial learning management systems. The «accessible and adaptive» philosophy has raised the bar throughout the industry in terms of what an LMS should provide.

Much of the future is already mapped out for ATutor. In the short term, an automated module management system will be introduced so that ATutor can be extended with a variety of add-on tools that support many different online learning activities. This will encourage broader participation from those in the ATutor community who have developed their own tools and wish to make them available to others either as an add-on or as part of ATutor's core source code thus ensuring that their tool will continue to develop as ATutor evolves.

Development of the ATalker add-on will continue with an audio interface being added for ATutor, as well as text reading and audio rendering capabilities for learning content.

The TILE learning objects repository is another parallel project being undertaken at the ATRC. The TILE project is responsible for developing a learning content repository and content transformation tools. Like ATutor, the TILE repository adheres to the accessibility and adaptability philosophy, though from a content perspective. Content created in the TILE repository can be enhanced with alternative formats to accommodate special needs, and that content can be

imported into ATutor, and vice versa (i.e. content from ATutor can be exported to the TILE repository).

Development on such projects as AComm and AChat will also continue providing accessible synchronous communication tools for ATutor learning activities. Other projects like the BarrierFree Broadband Authoring Tool and Player will be introduced into the ATutor tool set enabling content developers to create accessible video-based content that can be presented from within the ATutor environment.

Information about these and other ATRC projects can be found through the main ATRC Web site (http://atrc.utoronto.ca), and through the ATutor Web site (http://www.atutor.ca).

BIBLIOGRAPHY

ADL (2001) Sharable Content Object Reference Model Version 1.2, URL: < http://www.adlnet.org/scorm/index.cfm > accessed on 27th July, 2005.

Bell Labs (1999) SABLE: A Synthesis Markup Language (version 1.0),URL: < http://www.bell-labs.com/project/tts/sable.html > accessed on the 9th of August 2005 .

Burnett, D., Walker, M. & Hunty, A (2004) Synthesized Speech Markup Language Version 1.0, W3C Recommendation. URL:< http://www.w3.org/TR/speech-synthesis/ > accessed on the 9th of August 2005.

Chisholm, W., Vanderheiden, G., & Jacobs, I. (eds) (1999) Web Content Accessibility Guideline 1.0, URL: < http://snow.utoronto.ca/initiatives/crseval/crseval.html > accessed on 19th July 2005

Deubel, P. (2003). An Investigation of Behaviorist and Cognitive Approaches to Multimedia Design. Journal of Educational Multimedia and Hypermedia, 12(1), 63-90.

Gay, G., Harrison, L., Richards, J, & Treviranus, J. (1999) Courseware Accessibility Study 1999, URL: < http://snow.utoronto.ca/initiatives/crseval/crseval.html > accessed on 19th July 2005.

Gay, G. Landon, B. & Treviranus, J. (2000) Inclusion in an Electronic Classroom, URL: < http://snow.utoronto.ca/initiatives/access_study/inclusion.html > accessed on 19th July 2005.

Ridpath, C. (2004) Web Accessibility Checker URL: < http://checker.atrc.utoronto.ca/ > accessed on August 9th 2005.