**A Comparison of Learning Management System Accessibility**

**Introduction**

If not designed with accessibility in mind, Learning Management Systems (LMS) can pose accessibility problems for students and instructors with disabilities. Depending on the features enabled for a given course, students with disabilities could find that participating independently and effectively is nearly impossible. Some LMS tools—Discussions, Quizzes, Chat, or Wiki tools, for instance—can be more problematic than others. Learning Management Systems are becoming richer and more complex applications, and if they are not designed with accessibility in mind, it can be next to impossible to make them accessible and usable to users with various needs.

**Increasing Awareness**

Thanks to the hard work of accessibility advocates, in particular the LMS collaboration/working groups across the country, accessibility has become part of the design criteria for most commercial and open-source LMS projects. This achievement was hard fought and was not obtained overnight. Accessibility advocates have worked on at least three fronts:

1. ***Convincing IT administration to include accessibility in the purchase process.***IT administrators needed to learn that accessibility, as with privacy and security, must be supported by the LMS by default, and that they must require accessibility in the same way they require any other feature, adding accessibility to the purchasing criteria in the RFP processes.
2. ***Working with vendors to incorporate accessibility as part of the design phase.*** Vendors needed to learn that accessibility is not a "touch-up” after the completion of development but must instead be an integral part of the design process.
3. ***Pressuring for standards.*** Work with legislative and judicial entities helped define and pave the way for accessibility standards for applications used by the public, including those used in educational systems.

The good news is that we have achieved a respectable degree of public awareness for accessibility on all levels. Many IT administrators are now considering accessibility features in their product selection or at least ask about accessibility in the RFP processes. LMS vendors and open-source developers have realized that by considering accessibility in their design and implementation processes, they can make their products more accessible and usable for everyone, while increasing the likelihood that their products will be more readily adopted. We also now have several important state accessibility purchasing regulations like the [Illinois Information Technology Accessibility Act (IITAA)](http://www.dhs.state.il.us/page.aspx?item=32765), as well as other [Federal regulations](http://www.ada.gov/anprm2010.htm), in addition to [Section 508](http://www.section508.gov/), that are currently pending. Over the past few years, commercial LMS vendors, like Blackboard and Desire2Learn, and open-source communities, like Moodle and SAKAI, have invested significant resources into making their products more accessible. Indeed, these products are more accessible now than they have ever been.

**The Need to Consider Accessibility as Universal Usability**

At the same time, however, and in spite of significant strides that have made many Learning Management Systems more accessible, using these applications can still be a challenge for users with disabilities. A key reason for this is that significant usability factors that would have directly benefited all users' interaction and experience have been overshadowed by the more immediate, "pure” accessibility problems, such as ensuring alternative text for images. These essential fixes can sometimes obscure broader usability concerns and can even distract vendors from core problems that affect usability universally.

As a case in point, let's consider notification mechanisms. Most of us have experienced the scenario where we have submitted a form and don't receive any obvious notification whether the submission was successful. While this usability issue can be manageable for typical users, it can be extremely difficult for users with disabilities because they often end up reading the entire page from top to bottom just to find out if there is any acknowledgement of their form submission. Another example is when users are timed out of the LMS session but receive no notification of it—they simply find themselves at a login prompt, often having lost what they were working on. Alerting users that an active session is about to time out and providing a mechanism to renew the session is another usability feature everyone benefits from, particularly users with disabilities.

There are numerous examples such as these that are not "pure” accessibility issues and that are in fact general usability issues, but that have an overwhelmingly negative impact on the accessibility of the LMS. Ultimately, although supporting pure accessibility features in an application is essential, it does not guarantee an effective user experience. To accomplish this, we must consider basic usability factors that expedite the user experience and improve productivity, all the while guaranteeing that these are implemented with accessibility in mind.

Based on these considerations, we compiled criteria that affect the functional accessibility of four learning management systems—two commercial systems, Blackboard and Desire2learn; and two open source systems, Moodle and SAKAI.

**Author Credits**

This project was initiated and spearheaded by Hadi Rangin, Information Technology & Collaboration Coordinator from the University of Illinois, who has established and run collaboration groups with many vendors, including the [Blackboard Accessibility Interest Group](http://collaborate.athenpro.org/group/bb/), that led to improving accessibility of the Blackboard LMS over the past years.

Other members of this project include Ken Petri, Director of the Web Accessibility Center at The Ohio State University, Marc Thompson, an instructional designer with Online & Continuing Education at the University of Illinois, who is actively involved in accessibility, and has been designing and developing courses for the Moodle LMS for a number of years, Joe Humbert, Adaptive Technology and Accessibility Specialist from Indiana University, and Dan Hahn, eLearning Professional at University of Illinois who helped me with re-testing and verifying the Blackboard accessibility evaluation. Ken and Joe are experts in accessibility and are currently involved in the [Desire2Learn Accessibility Interest Group](http://collaborate.athenpro.org/group/d2l/) and the [SAKAI Accessibility Working Group](https://confluence.sakaiproject.org/display/2ACC/Accessibility+Working+Group), respectively.

**About the Evaluation and Results**

Each LMS presents a different learning environment with unique features and implementation techniques. While all tested LMS offer all common and basic tools/modules, the level of functionality and their implementations of the same tool can be significantly different. For example, Moodle offers a basic discussion tool with standard functionality, whereas Blackboard offers a more complex tool that is also potentially more difficult to make accessible.

In this round of LMS accessibility evaluation, we focused on functional tasks instead of testing specific pages of tools. For example, in the Login, Configuration, and Compatibility Testing category, we ask, "Can users detect error messages and recover from errors with certainty?" instead of checking if the login page is accessible. By focusing on functional tasks like these, we examine the accessibility of a tool with all related interaction and processes as a whole, regardless what implementation technique has been used in that tool.

We would like to underscore that we have tested only the core and common features of selected tools for accessibility and that our accessibility rating does not convey any information about the overall functionality of the selected tools. We are hoping to educate the public and product development teams about how the presence or absence of certain key usability/accessibility features can significantly impact users' experience. Toward this end, we provide a rationale for each evaluation category before defining the testing criteria. This testing protocol and rationale are then followed by specific testing results for each LMS, after which we draw our conclusions and present a side-by-side comparison.

**The Learning Management Systems We Compared**

We tested the following Learning Management Systems:

* Blackboard 9.1 Service Packs 6 and 8
* Desire2Learn 10
* Moodle 2.3
* SAKAI 2.8.0

We would like to thank the University of Illinois, The Ohio State University, and Indiana University and their respective LMS administrators for facilitating the necessary testing environments and helping us to understand the administrative view of these LMS. We would also like to thank Blackboard and Desire2learn for working with us to find answers to a number of questions we were not sure about.

**Disclaimers**

All statements and opinions of this report are entirely those of the authors. Our testing does not reflect a full or complete accessibility evaluation. This report does not represent the view or opinion of the presenters' respective institutions. Our results are preliminary, and we focus only on selected modules/tools. It is possible that we did not test for some relevant functional accessibility features. If you have suggestions for improving the testing protocol, please send your feedback to [hadi@illinois.edu](mailto:hadi@illinois.edu).

Accessibility/usability testing and evaluation were performed at different times in 2012, and our comparisons are thereby uneven in this regard. Testing that was more recent, as with the latest version of Desire2Learn version 10 (released December 2012), is thus more likely to reflect recent accessibility improvements. By contrast, testing that was less recent does not always reflect the newest enhancements introduced in the most recent LMS versions.

Blackboard revised its discussion board tool since our evaluation, for example, and is thus not included in the version we tested. Similarly, since the public release date of Moodle 2.4 (December 2012) after our evaluation, a number of accessibility enhancements have been added—most notably the incorporation of ARIA landmarks and roles—that were not included in our evaluation results.

Our evaluation of SAKAI was focused on version 2.8.0, and did not include the following accessibility improvements introduced in SAKAI 2.9 (released December 2012):

* The CKEditor is now the default editor throughout the SAKAI platform.
* Numerous unlabeled form controls across multiple tools are now explicitly labeled, particularly in Test & Quizzes tool
* Resources tool JavaScript menus are now accessible using keyboard and screen reading software.
* General improvements to the discoverability of required form controls.
* Buttons with ASCII characters now contain plain text.
* Primary language is now set for every page and is based on user preferred language.
* Improved Keyboard-only and screen reading software navigation of newly designed portal menus and pages
* Keyboard accessible skip navigation
* Added more ARIA landmarks

**Categories of Testing/Evaluation**

No doubt our testing categories are incomplete, owing to limited time and resources. We were determined to focus on commonly used (and thus high priority) aspects of the various LMS—primary modes of interacting with the systems and typically used "tools." If you have suggestions on how to expand our testing categories and related criteria, please contact Hadi Rangin at [hadi@illinois.edu](mailto:hadi@illinois.edu).

Considering the primary interactions in a typical Learning Management System environment—between end-users and the LMS, between students, and between instructors and students—we identified the following major categories for testing. then we divided each category in logical task groups.

1. **Login, Configuration, and Compatibility Testing**
   * Login and Error Recovery
   * System Compatibility Testing
2. **Personalization and Customization**
   * Layout Customization
   * Styling
   * Session Time Out
   * Saving Current State
   * Default Page
   * Selecting Editor
   * Alert Type
3. **Navigation**
   * Page Title
   * Breadcrumbs
   * Navigation Bars and Menus
   * Navigation Technique
   * Linearization and Tabbing Order
   * Link Type and Link Text
   * Visible Indication of Cursor Focus
   * Tooltips Technique
4. **Forms**
   * Form Control Labels
   * Form Submission
   * Notification/Verification
5. **Help and Documentation**
   * Inline Help
   * Tutorials and Guides
6. **Common Student Facing Modules/Tools**
   * Announcements
     + Navigation Features
     + Operation Features
   * Discussion
     + Navigation Features
     + Reading Messages
     + Replying to Messages
     + Message Discovery
   * E-mail
     + Navigation Features
     + Composing
   * Chat
     + Navigation Features
     + Text-to-Speech (TTS)
     + Chat Log
     + Chat Room Participation
     + Posting a Chat Message
   * Assignments, Activities, Course Content, Learning Modules
     + Navigation Features
     + Uploads/Downloads
   * Grade Book
     + Navigation Features
   * Quizzing/Testing Components
     + Navigation Features
     + Question Types (only minimum/common types listed)
     + Saving Quiz Progress/Draft
     + Notification and Verification
7. **Authoring Tools and Content Creation**
   * File Uploading
   * Authoring Tool
   * Grade Book (Instructor Level)
   * Multimedia Content Handling

As mentioned earlier, each LMS offers unique functionality and uses different technology to implement that functionality.

**1. Login and Configuration/Compatibility Testing**

**Rationale**

The login page is usually the first point of contact for a user interacting with an LMS, often through a localized/customized page set up by the hosting institution. In addition to the accessibility of the login page itself, login pages often have some form of application and/or browser test to check for specific tools/applications required by the LMS, such as Java, or for browser settings that enable JavaScript or pop ups. When third party components or plug ins are required, the accessibility of the LMS testing procedure and test results page are essential for providing users with vital information about any necessary configuration and software components required by the LMS.

**Criteria**

* **Login and error recovery:**
* Can users interact with the login page and submit necessary credentials with certainty?
* Can users detect error messages and recover from errors with certainty?
* **System Compatibility Testing:**
* Can users perform the necessary tests to ensure compatibility with their computer?
* Does the system have features to alert users about required tools like Java and enabling JavaScript?

**Results Summary and Comparison**

* **Blackboard:** Blackboard is doing good job in this category; they offer an accessible environment for login and compatibility testing but the error recovery needs improvement. There is a browser test module that tests for Java Script, Cookies, and Java Runtime Environment (JRE)
* **D2L:** D2L does well in this area and has in past versions. The out of the box login page has a simple and easy to navigate layout and uses accessible JavaScript alert dialogs to indicate errors. The system detects browser capabilities and will report if basic functionality, such as JavaScript, is not enabled.
* **Moodle:**Good job overall in providing accessible login and compatibility testing. Login error recovery needs work, as invalid login still returns user focus to the top of the page, and screen reader users must listen from the top of the page down until they discover the error message deeper in the tabbing order above the username and password field.
* **SAKAI:**SAKAI does a decent job in this category; they offer an accessible environment, but the error recover needs improvements. The login error should be more obvious to users. SAKAI does not provide any compatibility tests in this version.

**2. Personalization and Customization**

**Rationale**

Users have different needs and ways of viewing and interacting with the LMS. As such, it is vital that users be able to easily configure the LMS environment to their individual needs, instead of adapting to the application. Although they allow users to customize their experience to one degree or another, there are many global settings and personalization/customization features that can significantly augment or limit the usability/accessibility of complex applications and the LMS.

**Criteria**

* **Layout customization:**
* Is the default layout the most accessible?
* Can the user configure and select desired layout independently and save the setting globally?
* **Styling:**
* Can the user set globally the desired contrast, bg/fg colors, font-size, and font-type?
* **Session time out:**
* Can the user set globally the session timeout to the desired length?
* Is the user alerted before the session expires and can it be extended if desired?
* **Saving current state:**
* Can the user save/bookmark the current state and return to it on request? (side menu collapsed, hide unwanted menus, bookmark the state of a module e.g. page 6 of lesson 5)
* **Default page:**
* Can the user set the default page globally?
* **Selecting Editor:**
* Can the user set a desired editor globally?
* Can the editor be changed on-the-fly?
* **Alert Type:**
* Is the default alert type accessible?
* Can the user select desired alert type?

**Results Summary and Comparison**

* **Blackboard:** Blackboard does not allow user set the global timeout session nor does it provide an alert before a session expires. Blackboard uses two alert types one that works best with Java Access Bridge but the embedded page alerts do not work consistently. Bookmarking can be improved by allowing users to bookmark areas in a course, but instead it is done through the Content Collection, which complicates the user experience because only the instructor can set the default Course Entry Page. User can set the editor globally on the fly, but can also be set in the Personal Information page which is difficult to navigate to.
* **D2L:** D2L allows users a significant amount of customization. Especially welcome is the being warned about session time out (through accessible JavaScript alert) and being allowed to extend the session. Though the default page cannot be set, the state of the page (sections displayed or set expanded or collapsed) is remembered between sessions. Choice of editor (WYSIWYG or raw text) and font size and family are configurable globally, as is the type of alert (modal or potentially more accessibility pop-up window).
* **Moodle:** Moodle has made excellent strides in shifting a number of personalization and customization options to the user that were previously controlled by the course administrator and teacher roles. Block configuration in a course site is now user configurable and saved globally, and users may dock or collapse block content to simplify keyboard navigation and overall navigation.
* **SAKAI:**SAKAI provides only a few global and tool specific customizations, more personalization and customization options could be useful. However, the session time out and other alerts are accessible, though not the easiest to discover.

**3. Navigation**

**Rationale**

Navigation is the most important element of accessibility. Proper use of structural markup is critical to achieving effective, consistent, and logical navigation that allows keyboard users to navigate within the application effectively and efficiently, and to obtain necessary information quickly and easily.

**Criteria**

* **Page title:**
* Does the page title convey current information about where the user is?
* **Breadcrumbs:**
* Does the page provide meaningful breadcrumbs?
* Can breadcrumbs be easily located/navigated to?
* Can breadcrumbs be used for navigation (are they clickable)?
* **Navigation bars and menus:**
* Are the navigation bars and menus accessible?
* Are they used consistently across the application?
* **Navigation technique:**
* Does the page provide full ARIA landmarks?
* Are proper landmark types used for major components of the page?
* Does the page provide a logical heading structure for major components of the page?
* Is the main content area labeled with heading 1?
* Are there shortcut keys to move between major sections of the page?
* Is there a skip to main content or equivalent available to keyboard users?
* **Linearization and tabbing order:**
* Does the page linearize logically without breaking the sequential flow of the reading order?
* Is the tabbing order logical?
* **Link type and Link text:**
* Is the link focusable, and can it be activated with the enter key?
* Is link text understandable out of context?
* Are repetitive links given unique link text?
* If using image icon only as a link, does the image have unique alt text?
* If using image icon in the same anchor with link, is the image alt text empty (alt="")?
* Are the javascript-enabled links accessible?
* **Visible indication of cursor focus:**
* Can the element with the focus be easily identified visually?
* **Tooltips technique:**
* Is the tooltip accessible to keyboard users?
* Is the tooltip accessible to screen reader users?

**Results Summary and Comparison**

* **Blackboard:** Blackboard does a good job at providing heading one for the major section of each page. Blackboard offers flexibility for the instructor to design the course menu and layout of content. Navigation can be improved by offering unique page titles. Breadcrumb implementation is well done overall but is not entirely consistent.
* **D2L:** Breadcrumb navigation has been added to some interfaces and enhanced in others in the latest version (10) and it is easily navigable. D2L 10 makes good use of ARIA landmark regions for gross screen reader navigation and its heading hierarchies are very well thought through. Keyboard navigation is easy to follow visually. D2L 10 introduced new navigation drop-down menus. Implementation of these menus differs between top level navigation and page/main content regions. Keyboard and screen reader performance differs, as well, and may take orientation training to be used accurately and effectively.
* **Moodle:** Overall navigation is good with clearly identified page titles, consistent use of breadcrumbs easily located in the tabbing order, and block navigation marked by consistent heading level 2 identifiers. Visible indication of cursor focus could be more pronounced, and titles could be applied more consistently to links as tooltips, particularly in the case of user-created page content.
* **SAKAI:** Overall navigation is good with unique page titles, consistent main navigation preceded by an h1, and excellent visible indication of focus. Improvements to the keyboard accessibility of tooltips and sporadic instances of non-unique or descriptive link text would enhance ease of navigation.

**4. Forms**

**Rationale**

Once in the the LMS, real interaction with the LMS starts with forms and entering data. To work effectively and efficiently in the LMS, the user needs to concentrate on the data being entered and not be challenged or distracted by the complexity of the interface and navigation. In this context, it is critical that all the form controls be accessible so the user can enter data with certainty. It is equally important that the user be notified about any warning, error, or successful form submission, and that the user be able to identify easily places where errors have occurred and navigate to them.

**Criteria**

* **Labeling form controls:**
* Across the application, are form controls labeled correctly?
* Across the application, are form fields labeled correctly?
* Across the application, has the onchange event handler been avoided?
* **Form submission:**
* Across the application, are the submit/cancel buttons accessible?
* Across the application, are the submit/cancel functions used in a consistent manner?
* **Notification/verification:**
* Across the application, does it provide an accessible verification message on successful submission?
* Across the application, is the error notification accessible?

**Results Summary and Comparison**

* **Blackboard:** The submit/cancel buttons are accessible and consistently implemented across the application. Required fields, error notification, and submission confirmation are not consistently labeled, announced, or focused to a screen reader. Most controls have adequate labels but there are pages where more detail will improve the user experience.
* **D2L:** D2L is consistent in using properly associated labels with all form fields and in version 10 adds aria-required to required fields. Submissions are announced via screen reader accessible aria alert messages. Error notification area is focused and has hyperlinks to errored fields, but when focused the errors are visible only, they are not associated to the problem fields with labels (aria or otherwise).
* **Moodle:** Overall, form controls and fields are labeled correctly, although successful form submission confirmation and error notification could be improved in several areas. Form submission confirmation could be extended to include instructor-facing areas that involve assignment and activity creation, for example. Form submission error notifications are not always evident, and screen reader users may need to rescan the page to locate the notification even though user focus is placed into the first non-valid field.
* **SAKAI:** For the most part, form controls are explicitly and uniquely labeled. However, there is still room for improvement. Across the platform, there needs to be a more consistent use of verification and submission messages and focus, both visual and non-visual, should be placed on these messages when they appear.

**5. Help and Documentation**

**Rationale**

The concept of online learning and using an Learning Management system can be challenging to new users, in particular users with disabilities. Application developers use various styling and formatting effects to convey certain information to their users which is not necessarily apparent to users with disabilities. For example, some LMS use tab-style design for their layout. Sighted users can immediately see what tab they are on, but this might not be easily accessible to blind or low-vision users even though the necessary information is provided in the system. Screen reader users, for instance, might not be able to find the necessary information if they don't know where and how to look for it. Or keyboard users might not be able to utilize the built-in accessibility features if they don't know about them.

There are also tools and tasks that are not easy to use and require clear and straightforward explanation of the processes involved in those tasks. Documentation of supported accessibility features, a quick accessibility guide in fully accessible format outside the LMS, and accessible inline help/instruction are all essential for helping users learn, troubleshoot, and work effectively in the LMS. Whether internal or external to the LMS, the accessibility/usability of such help systems and documentation materials, as well as effective documentation of LMS-specific accessibility features and general accessibility guidelines, are vital to all users. Ultimately, it is very important for users with disabilities, and blind users in particular, to learn about the basic functions of the LMS in advance before they actually start using it. Moreover, understanding the general layout of the LMS can be extremely useful to some users with disabilities, and it is an important step in helping users with disabilities develop a strategy for navigating and interacting with the LMS.

**Criteria**

* **Inline Help**
* Is inline help provided across the application including for the editor?
* Are help tip links/buttons focusable and contextual?
* **Tutorials and Guides**
* Is the tutorial accessible?
* Is the tutorial available from within the application?
* Is the tutorial available from outside the application?
* Is the tutorial available publicly?

**Results Summary and Comparison**

* **Blackboard:** Blackboard does well at providing help inside the application and in context. The video tutorials provide subtitles and are publicly available with controls that are usable with a keyboard. The text based help is also well done and is provided to instructors and students within the application and can also be accessed without logging into the application.
* **D2L:** D2L has had well written and accessible inline help for the last few versions. Changes to help in the WYSIWYG editor are welcome and the editor announces available shortcut keys (to screen reader users, not clear how others users discover these, though). Tutorial guides are accessible and downloadable from D2L's corporate main web site. All links to help/information are unique, specific to the help topic.
* **Moodle:** Moodle continues to do a fairly good job of providing contextual help inline through graphical help links. Moodle has also made good progress since 1.9 in adding contextual links at the bottom of each page to relevant Moodle.org documentation outside the LMS for specific page content ranging from the course homepage and book modules to forum, assignment, and activity modules.
* **SAKAI:** SAKAI provides very good contextual help documentation throughout the platform. All help information is available both inside the application and through a publicly available website. Numerous SAKAI working groups are actively participating in an effort to continually update and improve the help documentation, including tool specific help information for users with disabilities.

**6. Common Modules/Tools (Student Facing)**

**Rationale**

The accessibility/usability of the common modules and tools of the LMS provide the necessary means by which student users create and interact with course content. The most essential and common modules/tools have been selected for evaluation, and the criteria for each tool is predicated on its functionality.

**6.1 Announcements**

**Criteria**

* **Navigation Features:**
* Can the user easily and reliably identify this tool/module and navigate to it?
* Does the announcements tool indicate how many new announcements have been posted?
* **Operation features:**
* Can user interact easily and reliably with applicable operations in this tool/module (read, filter, sort, delete, etc)?

**Results Summary and Comparison**

* **Blackboard:** The announcement tool in Blackboard integrates with e-mail, so when an Instructor or Teaching Assistant creates a new announcement then student users are notified via email. Once inside the module the newest announcement will be at the top of the page. The user experience can be enhanced by displaying the number of new announcements and allowing for filtering and sorting.
* **D2L:** "News" is organized by most recent first or by instructor preference. Items are identified by headings and can be dismissed easily (and recovered via search or viewing all). (Dismiss links are contextual, which prevents ambiguity for screen reader users.) Filters and search tools are accessible. An separate "updates" area on the home page indicates how many items have not been looked at/are new.
* **Moodle:** Content is clearly organized in a table format (thread titles, started by, replies, last post columns). Although the advanced search tool can be used to selectively isolate the Announcements forum and then apply specific search criteria, there is still no basic sort/filter specific to the Announcements forum that would help users quickly locate specific course Announcement content.
* **SAKAI:** Content is clearly organized both when viewing the list of announcements and the announcement details. Users can easily and reliably interact with most advanced operations (filter, sort, etc.). However, there is no way for student users to delete announcements. Improvements could be made to the way new announcements are displayed to make them more obvious to users. Currently, the new announcement indication is not present in the tool, but instead on the course homepage.

**6.2 Discussion**

**Criteria**

* **Navigation Features:**
* Can the user easily and reliably identify this tool/module and navigate to it?
* Can user easily and reliably navigate to different items in this tool/module?
* **Reading Messages**
* Can the user easily distinguish new threads?
* Can the user easily distinguish subthreads?
* Can the user easily distinguish replies?
* Can the user easily distinguish the parent message for a given reply or sequence of replies?
* Are the filter options accessible?
* Are the sort options accessible?
* **Replying to Messages**
* Can user reply directly via external e-mail systems?
* **Message Discovery**
* Can users be notified about new posted messages?
* Does it support RSS?
* Is the feature to subscribe/unsubscribe accessible?

**Results Summary and Comparison**

* **Blackboard:** This module can be integrated into the course depending on how the instructor sets it up: it can be a menu item which links to all discussion forums in the course, or it can reside in a content area as a link to an individual forum or all forums. While the filtering and sorting options provided by Blackboard are accessible, they can be enhanced by improving the heading structure to simplify navigation. This module can further be enhanced by increasing the functionality for screen reader users in the discovery of new threads, sub threads, and replies.
* **D2L:** There are multiple ways discussions can be displayed, configurable by the user. Some may be preferable to users with learning disabilities. The most accessible layout for screen reader users is to view messages unthreaded and in "reading" view. A tree menu of forums and topics has been added to help with navigation between forums and sub-topics. It has decent accessibility but could be improved by being an aria tree menu. Message threads have a shallow heading hierarchy (all messages and replies are H2 headings) which may make distinguishing replies a challenge for screen reader users.
* **Moodle:** Initial forum content is clearly organized in a table format (thread titles, started by, replies, last post columns); however, more work needs to be done to improve organization and navigation at the thread level, as there is still no easy mechanism for navigating from one post to another or, in the case of screen reader users, situating oneself contextually in a given discussion thread. Moodle's advanced search functions are very robust, and allow fine-grained control over a wide array of useful search options.
* **SAKAI:** The initial forum content is organized in a nested table layout that is easy for a user to understand once oriented, but an improved and easier navigation scheme is needed throughout the tool for users with disabilities. New messages are very distinguishable. Improvements and additions are also needed for un/under-supported options (RSS, External reply, and subscription options).

**6.3 E-mail**

**Criteria**

* **Navigation Features:**
* Can the user easily and reliably identify this tool/module and navigate to it?
* Can user easily and reliably navigate to different items in this tool/module?
* **Composing**
* Can user easily and reliably identify and interact with the applicable fields such as To, CC, Attachment, and Message body?
* Can user easily and reliably include recipients from the Address book?
* Can user easily and reliably style and format a message?
* Is the Add Attachment feature accessible?
* Is the Spell-check feature accessible?
* Is user notified upon successful message delivery in an accessible way?
* Is user notified about errors/warnings in an accessible way?

**Results Summary and Comparison**

* **Blackboard:** New in Blackboard Learn is the ability to e-mail externally. This functionality is enabled for students by the instructor. Screen reader focus can be improved in the tool to verify a message was sent. Users have the ability to send messages in mass to users with certain roles or group members, though the selection of individual recipients can be improved.
* **D2L:** D2L has done a good job in making all aspects of the e-mail interface accessible. However, selecting and managing contacts is a bit cumbersome (no accessibility issues, just a lot to slog though for screen reader uses, in particular). The editor's spell check feature is accessible, but is buried on an "advanced" menu and difficult to discover. Notification of submission success or issues is easy to discover.
* **Moodle:** Although some e-mail capability for short messages is built into the base installation’s messaging system, Moodle does not include a standalone e-mail tool in its base installation package. Various e-mail packages from the community, like Quickmail, are available as add ons.
* **SAKAI:** SAKAI current messaging tool is generally usable, but is used for internal course communication only. Composing a message will pose the most difficulties for users because the rich text editor is not fully accessible and has many custom shortcuts that need to be memorized. In addition, there is no confirmation message presented to users and focus is not given to error messages.

**6.4 Chat**

**Criteria**

* **Navigation Features**
* Can the user easily and reliably identify this tool/module and navigate to it?
* Can user easily and reliably navigate to different sections in this tool/module?
* **Text-to-speech (TTS):**
* Does the Chat Tool speak new messages as they appear (using built-in TTS or ARIA live regions)?
* **Chat log:**
* Can user configure how to display the chat thread? (most recent to oldest or oldest to most recent?)
* Can user suppress time-stamps?
* Can user select and copy the content of chat log with keyboard alone?
* Are clickable items such as hyperlinks or send mail clickable?
* Can user be notified audibly on new incoming posts?
* Is a private message labeled as private?
* Can user change the chat font size and style?
* **Chat room participation:**
* Can user discover who is in the chat room?
* Can user easily and reliably locate the desired participant and perform applicable functions, such as sending a private message?
* **Posting a chat message:**
* Are the basic text editing and message navigation features supported (e.g., word-by-word, home, end navigation, delete forward/backward, etc.)?
* Can a message be submitted upon pressing enter?

**Results Summary and Comparison**

* **Blackboard:** There are many areas where Blackboard can improve the chat module. The chat module cannot be started in Firefox or Internet Explorer without installing an out of date version of Java Access Bridge. Time stamps are not available to screen reader users for messages that are non-private. The user can copy the content of the chat log, but visually a keyboard user cannot see the cursor in the chat log. Users are not provided the functionality of increasing the font size.
* **D2L:** As with most LMS's, the D2L chat tool is very basic. It has nods to accessibility, which include good heading structure, audible alerts (a tone) for users entering/leaving and new messages. It is also possible to order the messages new to old to improve ability to find most recent message for screen reader users. ARIA has been used in other parts of the LMS to enhance interactivity, but beyond the standards landmark roles, the chat tool has not been updated significantly since the last version.
* **Moodle:** Users may choose the "more accessible interface" chat option that provides headings, lists, and properly labeled forms for input. The persistent problem with the Moodle chat interface since 1.9 is that it is not live updated but must be manually updated by the user by clicking the refresh button next to the send button in order to see new incoming chat messages.
* **SAKAI:** The SAKAI Chat room tool is very usable because it has a very simple structure and plain text. However, it does not offer live updates or many of the features (private messages, customizations, hyperlinks, etc.) provided in other LMS platforms.

**6.5 Assignments, Activities, Course Content, Learning Modules**

**Criteria**

* **Navigation Features**
* Can the user easily and reliably identify this tool/module and navigate to it?
* Can user easily and reliably navigate to different sections in this tool/module?
* Can user easily and reliably discover new assignments/activities and navigate to them?
* Is there a single place to view the outline of assignments/full syllabus of the course?
* Is it possible to determine what sort of content is linked to in outline/syllabus (links to PDF, PowerPoint, HTML content, etc.)?
* Is there a way for the user to bookmark a location in content?
* Is there a simple way to return to the last viewed content?
* **Uploads/Downloads:**
* Can the user easily and reliably operate the download feature?
* Can the user easily and reliably operate the upload and attachment features?

**Results Summary and Comparison**

* **Blackboard:** There is a Syllabus tool provided, but most instructors use the more convenient process of adding an attachment to an Item in a content area, there isn't a unified way for a student to view their syllabi across courses. The assignment module in Blackboard provides the student with a content editor to input their work, but depending on how the student is instructed to deliver their work the user may encounter difficulty in attaching a document.
* **D2L:** Similar to previous versions, course content is very well organized, navigable, and accessible. Course content types are easy to discover from their links, bookmarks can be set and what content has already been visited is indicated in the table of contents for a course.
* **Moodle:** Overall course content is easily accessible by means of the course home page's 3-column table format and blocks marked consistently by heading level 3 identifiers. Uploading and downloading content through the new File Picker tool poses some significant accessibility concerns that stem largely from the fact that it is a widget with non-modal dialogue that is treated as page content rather than a separate window. There is no way for a screen reader user to tell that the file picker widget is a separate window (and not part of the page content itself).
* **SAKAI:** Overall Course content is easy to navigate and understand. Improvements could be made to the consistency and the discoverability of file types, previously viewed content and new assignments. There is also the limitation of single file upload and download without external tools.

**6.6 Grade Book**

**Criteria**

* **Navigation Features**
* Can the user easily and reliably identify this tool and navigate to it?
* Can user easily and reliably navigate to different sections in this tool/module to view grades for various assignments?
* Can user easily and reliably access individual scores and any related instructor feedback for specific assignments that have been graded?
* Can user easily and reliably locate the current cumulative grade for completed work in the course?
* If grade tool offers statistics for student score relative to the class average, is this information accessible?

**Results Summary and Comparison**

* **Blackboard:** Blackboard provides an accessible table layout for students to view their grades and read the instructor's feedback and comments.
* **D2L:** The grade book is easy to navigate by keyboard and screen reader. D2L has done significant work to guarantee that comments, grades, and class statistics are fully screen reader accessible. As noted in our detailed analysis, there is a slight issue with screen reader navigation of the list of quizzes (column alignment of nested grade items in the grades overview table may cause some confusion), but overall this area of D2L is highly accessible.
* **Moodle:** Grade Book continues to be keyboard navigable as a table that makes effective use of table headings and scope to make grade book content accessible to screen reader users.
* **SAKAI:** The SAKAI gradebook tool is very accessible. The tool consists of only one page and most information is contained in one table. There are a few drawbacks that pose not accessibility issues. First, instructor feedback must be accessed in the tool where the grades item was created. Second, statistics which can be accessed by student users are only generated after the final course grade is calculated.

**6.7 Quizzing/Testing Components**

**Criteria**

* **Navigation Features**
* Can the user easily and reliably identify this tool/module and navigate to it?
* Can user easily and reliably navigate to different sections in this tool/module?
* Can user easily and reliably navigate between questions?
* Can user easily identify unanswered questions?
* Can user easily identify point value for each question?
* Can user easily identify required or optional questions?
* Can user easily obtain information about elapsed/remaining time?
* **Question Types (only minimum/common types listed)**
* Can the user easily and reliably interact with the multiple choice question type?
* Can the user easily and reliably interact with the true/false question type?
* Can the user easily and reliably interact with the short answer question types?
* Can the user easily and reliably interact with the blank question type?
* Can the user easily and reliably interact with the numerical/arithmetic question type?
* Can the user easily and reliably interact with the matching question type?
* **Saving Quiz Progress/Draft:**
* Can quiz results be easily saved without redrawing the page?
* **Notification and verification:**
* Does the quiz tool provide an accessible verification message on successful submission?
* Can the user easily and reliably detect and address error notifications?

**Results Summary and Comparison**

* **Blackboard:** Navigating to online tests is straightforward but the flexibility provided to the instructor to deploy online tests in different areas can create inconsistent navigation across different courses. Users can easily identify point totals, questions and unanswered questions. Labeling of the question type needs improvement because the screen reader user has no way of identifying it. Labeling of the answer or input field is also a problem because the screen reader user cannot tell if they are answering the right question.
* **D2L:** Some of the question types have accessibility issues but most are generally accessible. The option to have one question per "page" of the quiz (configurable by the instructor) can present difficulties for screen reader users, in particular, because there is no announcement that a quiz response has been successfully saved or that the next question has been loaded (both operations are dynamic, same-page refreshes of content).
* **Moodle:** Although quiz results cannot be saved without redrawing the page, Moodle does a good overall job of providing an accessible quiz tool and question types. Notification and verification are also effective in providing a list of any answered and unanswered questions before the user is presented with the option to "submit all and finish," and then confirm submission, before submitting.
* **SAKAI:** SAKAI needs significant improvements to the test and quizzes components. Many question types have inconsistent or unlabeled form controls and non-required questions are hard to discover. However, the notification and verification is excellent and users can easily find unanswered question (if the instructor allows this feature). Many of this issues have be patched in newer versions of SAKAI and are in the process of being back ported to older versions.

**7. Authoring Tools and Content Creation**

**Rationale**

Authoring tools, including HTML editors, file uploading tools, and helper tools for specific content creation, and multimedia handling, all significantly impact the way course content is designed in the LMS, as well as the accessibility of the content created by those authoring tools. The best time to address the accessibility of the content is when it is created. Toward this end, authoring tools should be accessible, feature-ful, easy to use, and prevent invalid and or inaccessible HTML code. The LMS can play an vital role in educating users about the accessibility of the content they create, upload, or reference. For example, authoring tools can warn or test for accessibility on the files being uploaded, prevent users from saving graphics without first specifying the type of image (stylistic/informative), prompt the user for alternative text in the case of informative images, or alert users about captions for video clips, etc.

**Criteria**

* **File uploading:**
* Does the tool perform an automated accessibility evaluation of the content or provide a wizard that directs the user toward accessible content?
* Does it allow multi-file selection and upload?
* Does the tool offer helper features, such as graphics upload and alt tag wizard, that help make content more accessible and educate users about the accessibility of the content they create?
* **Authoring tool:**
* Is the primary editor accessible?
* Does the editor support HTML format?
* Is the editor capable of stripping out all non-HTML markup (bad code)?
* Does the editor support wiki-style markup format?
* Does the editor support plain text format?
* **Grade Book (instructor-level):**
* Can the user easily and reliably navigate to different sections in this tool to manually enter grades for various assignments?
* Can the user easily and reliably navigate to different sections in this tool to enter comments and feedback on individual assignments?
* For group assignments, can the user easily and reliably isolate specific assigned groups for the purpose of assigning grades and providing feedback?
* Can the user easily and reliably change the weighting on specific assignments and assignment categories?
* Can the user easily and reliably create and deploy basic calculation formulae/grading schemes (e.g., for averaging grades, assigning extra credit, etc.)?
* Are other LMS-specific features, such as Release Conditions or Grade Visibility settings, able to be easily and reliably used?
* **Multimedia Content Handling**
* Can user easily and reliably use the LMS-provided multimedia player?
* Does the provided video player support captions?
* Does the provided video player support audio description?
* Can user play audio/video content with desired client-based multimedia player?

**Results Summary and Comparison**

* **Blackboard:** The default content editor is not accessible but the user can change to a text or HTML based editor. Blackboard does provide an area to enter alt text when uploading images. For multimedia playback Blackboard does not provide a media player, the new Blackboard Learn content editor provides for easy integration of embedded media from outside sources (Flickr and YouTube), so it is the responsibility of the instructor to embed accessible players. If media files are hosted in the LMS the user's system default player will launch.
* **D2L:** Overall, tools for creating content and grading are accessible in D2L. There are some issues with the WYSIWYG editor, noted previously, having to do with non-obvious ways to leave the editor, which may affect screen reader and keyboard-only users. And there is no accessible default media player -- though D2L content pages will accept custom players with some minor work by system managers or content creators. D2L added universal support for MathJax in D2L 10. This provides a straight-forward means to create accessible math (via MathML) within the LMS.
* **Moodle:** The introduction of the File Picker tool in Moodle 2 brings with it some significant navigation issues, as it is a widget that, when launched, is treated as page content rather than a separate window. There is no way for a screen reader user to tell that the file picker widget is a separate window (and not part of the page content itself). Other authoring tools, like the default TinyMCE editor, currently provide some accessibility options and prompts like the prompt for alt text that also warns users if they have not provided alt text for an image, for example. The editor could provide more extensive WYSIWIG accessibility options and prompts (prompts for defining table headings, scope, and summaries, for example, or for prompting users to distinguish decorative from informational images when adding images).
* **SAKAI:** SAKAI does not offer any features to help content creators improve or verify the accessibility of the content they upload. The primary editor also has a very limited feature set compared to other available WYSIWYG editors including wiki markup and HTML support. A new editor will be used in future versions of SAKAI that is more completely accessible an offers more features. The Grade Book tool is has a very limited feature set and is more of a overview tool. All grades and options must be changed in their native tool and not in the Grade Book tool. SAKAI offers no native multimedia support.