

IMS Learning Tools Interoperability: Enabling a Mash-up Approach to Teaching and Learning Tools

CHARLES SEVERANCE*, TED HANSS AND JOSEPH HARDIN

University of Michigan

This paper describes how we can move from a single Learning Management System such as Sakai, Blackboard, Moodle, or Desire2Learn to a cooperating collection of networked learning content and tools. The move from a single-silo of functionality to the ability to use tools from anywhere on the web will greatly increase the variety of tools available to teachers and learners and should enable innovation in teaching and learning using technology. While Web 2.0 is well understood for things like RSS feeds, real use of Web 2.0 mash-up approaches for teaching and learning has unique challenges that must be solved before these approaches will become common and mainstream. When students are in a course, they have an identity and role in that course and these identities and roles along with course information must be communicated to these tools out on the web so the course model is maintained in the tools. For example, when using a blog for the course, the teacher should be able to “administer” the overall blog while students can post to their particular areas of the blogs. This paper describes the issues and use cases around teaching and learning mash-up and introduces the IMS Learning Tools Interoperability standard that addresses these use cases.

Keywords: Mash up, web services, service oriented architecture, web 2.0.

INTRODUCTION

Currently most learning management systems (LMS) in the market are relatively mature and stable. LMSs systems are increasingly treated like other large-scale enterprise-wide applications. This means that the systems are managed

*Corresponding author: csev@umich.edu

much more conservatively - with fewer upgrades and less flexibility for instructional content developers who want to experiment with new software, capabilities, approaches or content. Even though such systems as BlackBoard, Moodle, Sakai or ANGEL Learning are designed to be easily expanded in a modular fashion, strict enterprise-wide production requirements tend to limit the ability for faculty and instructional staff to explore new approaches to learning in production environments. Demands for stability and reliability in the core system always slow the integration of new functionality.

There is a need for core LMS systems to allow interesting innovation while at the same time keeping those systems very solid in daily production. We need to create learning environments where we can balance innovation and stability. The solution is to use a service-oriented approach to separate the functions of a learning management system and distribute them across multiple systems. Taking a service-oriented approach to learning tools allows production to be handled by one set of servers and more innovative activities to be handled by a different set of servers.

When we take a standards-based service-oriented approach to learning tools, it also creates markets for software as a service and content as a service. If we can reduce the cost of integrating externally hosted content and software into LMS systems, we should see a diverse set of publisher content and software services available to teachers and learners. We will also see a set of open source learning tools that function across multiple learning management systems. And we will have a new, flexible way to work with the increasing amount of open content that is available to teachers and learners. The net effect, we believe, is a path that better reflects learner-centered and peer-learning approaches to learning.

When the service-oriented approach reaches full maturity, the LMS systems can provide the essential core functionality and external tools can provide many choices to teachers and learners. According to Dr. Andrew (Aggie) Booth of Leeds University, one of the principal designers of the Bodington Virtual Learning Environment (VLE):

"I certainly don't see the VLE or LMS continuing in the form that they currently are, where you have a monolithic application with lots and lots of different tools presented to teachers and students. You very quickly come to the limits of any tool that you use. The answer is to abandon the tool you have been using and to swap in another tool. Now I can see a situation where you extend this from a single tool to all of the tools of an LMS. I can see a situation where the VLE or LMS actually shrinks to a container into which you plug these learning tools." [Booth, A., 2007]

The IMS Learning Tools Interoperability [IMSLTI] is a standard under development that standardizes the protocols between the LMS system and the external tools

allowing external tools to function as if the tool were a native tool inside the LMS. The IMS Global Learning Consortium [IMS] is a nonprofit member consortium that creates standards for the development and adoption of technology-enhanced learning.

While the IMS Learning Tools Interoperability Specification was under development, the IMS Developer Network published an informal specification called SimpleLTI [Severance, C., 2008] in order to start a series of demonstrations and experiments to explore the architectural impact of a service-oriented approach to tools in a Learning Management System and inform the formal standards process. The IMS LTI working Group released a subset of the IMS LTI specification focused on the launch and roster provisioning aspects of IMS LTI called Basic LTI. With the development of Basic LTI, Simple LTI is deprecated and will be replaced by Basic LTI.

This paper describes IMS Tool Interoperability and some of the engineering work that has been done to insure that the ultimate standard works well and is easily implemented.

AN OVERVIEW OF IMS LEARNING TOOLS INTEROPERABILITY

The IMS LTI Specification includes the following scope:

- The provisioning and installation of an external tool in an LMS
- A tool launch protocol that securely establishes the user identity, course context, roster information, and role within course when the user launches the tool
- A series of run-time web services to allow the external tool access grades, files, and other data within the LMS as well as send data back to the LMS

Much of the inspiration of the IMS LTI Specification comes from the Facebook application protocol [Facebook] as well as the Blackboard Proxy Tool [BBProxy] facility for proxy tools.

The IMS LTI launch protocol is a simple REST [Fielding, Roy T., 2000] request that can either be used from an LMS system or more commonly submitted as a form in a browser. The LTI launch uses the OAuth [OAuth] pattern to securely sign the data that is part of the launch so the external tool can be assured that launches are coming from the LMS – even when the data is being passed through a form in the user's browser.

This makes the launch protocol very simple and powerful as well as very lightweight and efficient. The typical LTI Launch sequence when the data is passed through the browser is shown in Figure 1.

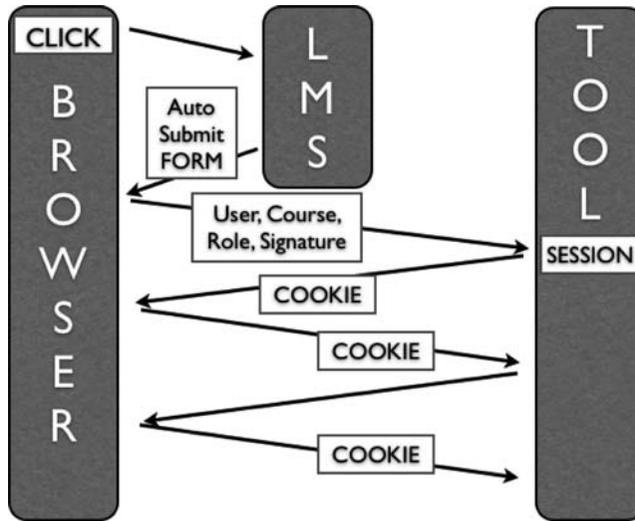


FIGURE 1
IMS LTI Launch Sequence.

First the user selects an external tool using the LMS navigation. The LMS then generates an HTML form and JavaScript to automatically submit the form to the external tool. The LMS places a whole series of hidden fields in the form that indicate the user, course, and role for the current launch and uses the OAuth standard to sign all of that material and includes the OAuth signatures in the form as well. When the browser receives the form, it is automatically submitted to the external tool. If JavaScript is turned off, the user must manually press the Submit button to continue to the external tool.

Once the external tool receives the form data, the tool verifies the OAuth signatures; uses the launch data to provision the user, course, and roster as necessary; creates a session for the user; and sends back an HTTP response including a cookie or other mechanism to maintain the tool's session. From that point forward the user simply interacts directly with the tool in their browser. The tool manages its session with the user's browser.

CASE STUDIES USING SIMPLE LTI

While the full IMS LTI and Basic LTI specifications were under development, we wanted to gain experience with the approach and protocols proposed for LTI.

With permission of IMS, we developed an informal specification based on an unapproved draft of the LTI documents during April 2008 [SimpleLTI] and released the specification in May 2008. We knew that the SimpleLTI would only be an approximation of how the final standard worked but it was sufficiently close that we could gain engineering experience which could be used to inform the development of the formal LTI standards.

Throughout 2008 and early 2009, we used SimpleLTI to develop several demonstrations and experiments to prove the concepts underlying LTI. Several students were funded to work on IMS LTI through the Google Summer of Code program in 2008[Google, 2008].

None of the experiments in this section are in permanent production – nor are they part of shipping products and do not represent any formal commitment on the part of the organizations listed below:

- A Sakai Tool for SimpleLTI
- A Moodle Module and Filter for SimpleLTI
- An ANGEL Learning tool for SimpleLTI
- Support for IMS SimpleLTI in Pearson MathXL
- Support for IMS SimpleLTI in McGraw-Hill’s Katana Server
- Support for IMS SimpleLTI in the IMS Common Cartridge Player from MIRLearn - Microsoft/LearnGauge [Holland, B., 2008]
- SimpleLTI Support in the Melete/Module tool in Sakai
- A PHP Framework for building SimpleLTI Tools
- A Python Framework for building SimpleLTI Tools

In addition to supporting SimpleLTI launch we did another experiment where the IMS SimpleLTI was added to the IMS Common Cartridge Specification. We produced an informal specification for including SimpleLTI references in an IMS Common Cartridge [IMSCC]. We used this informal IMS Common Cartridge Extension to produce cartridges from McGraw-Hill and Pearson with authenticated Simple LTI links to premium content embedded in the cartridge. These cartridges were loaded into ANGEL Learning and MIRLearn and demonstrated at the 2008 Educause Conference.

These experiments served to validate the use cases we envisioned when setting out to develop IMS Learning Tools Interoperability. Once we had working prototype software in hand we were able to identify and explore some unanticipated use cases for a flexible standards-based mechanism to “mash-up” content and functionality. We will describe some of the use cases for IMS LTI and IMS CC that were explored as part of these experiments.

It is important to reiterate that these are not shipping products – the screenshots below were taken from working technology demonstrations that were using IMS SimpleLTI to make the connections between the LMS systems and the content or tools. None of these screen shots represent any kind of commitment on the part of the mentioned companies in regards to any kind of delivered product. The purpose of these efforts was to research and understand feasibility of the functionality mash up approach [Severance, C., Hardin, J., Whyte, A., (2008)].

Software as Service – Pearson Education

Pearson Education provides a very popular product called MathXL [MathXL] that provides a wide range of features to support teaching and learning mathematics. Using IMS Common Cartridge and IMS Simple Tools Interoperability, we produced a course that includes module materials interspersed with references to the Pearson MathXL software. IMS SimpleLTI provided all of the single-sign-on, course provisioning, and roster provisioning between the LMS and Pearson servers. Other than setting a license key in the LMS provided by Pearson, no other configuration is required. A teacher could obtain an IMS Common Cartridge, load it into their course and simply begin using the content and software provided by Pearson.

This augmented IMS Common Cartridge was loaded into ANGEL Learning as well as the Microsoft/LearnGauge IMS Common Cartridge Player. Figure 2 shows the Pearson software running within the ANGEL Learning platform.

This demonstrates the use case where a publisher can create a course cartridge that includes both learning content to be stored in the LMS and served in the LMS



FIGURE 2
Pearson MathXL Running in ANGEL Learning using SimpleLTI.

as well as references to software that runs on publisher servers that appears to be part of the learning management system.

Content as Service – McGraw-Hill Higher Education / Katana

The Katana effort from McGraw-Hill Higher Education [Katana] provides teachers with reusable high-value content that can be used with any book from any publisher. The Katana design assumes that teachers will take the Katana content, load it into their LMS system, and then contextualize the content to the particular course in question.

The Katana modules are 10–20 minutes long and each one is self contained and authored to explain a particular concept. When Katana content is placed in an IMS Common Cartridge the content references use SimpleLTI. Each content object in the cartridge can be used separately – placed into the flow of the course as the instructor sees fit.

Figure 3 shows a Katana IMS Cartridge with LTI being played in the Microsoft/LearnGauge IMS Common Cartridge player.

Using IMS SimpleLTI allows McGraw-Hill Higher Education to take a very flexible approach to the business model around Katana. They can distribute cartridges very widely because the high value content is not in the cartridges. In addition, as the large media objects are not included in the cartridges, the cartridges are much smaller and errors in the content are much easier to fix.

Because each SimpleLTI launch includes organization, course, user, and role information, McGraw-Hill can use a very dynamic provisioning approach which could include; (a) giving away free sample content but knowing exactly which organization, course, and users who are making use of the sample content, (b)



FIGURE 3
McGraw-Hill Content Integrated into the MIR LMS Using SimpleLTI.

selling site-wide licenses to schools, (c) selling access to the content to individual students via a PIN code in a bookstore, and many others.

One hope for the future of high-value of content is that when standards such as IMS Tools Interoperability reduce the integration cost between LMS systems and high-value content to nearly zero, it allows publishers to spend more of their resources developing better content and lowering the cost of that content to the teachers and students.

Embedding a Learning Management System in Web Content – CloudSocial

There is an increasing trend towards universities publishing their course materials to be freely re-used as Open Courseware [OCW] and Open Educational Resources produced in projects such as the Connexions project [CNX]. Many teachers would also prefer to simply author their original content directly on the web instead of authoring within the confines and capabilities of the LMS.

One of the limitations of authoring outside the LMS is the loss of the collaboration, workflow, and tracking functionality provided by the LMS. LMS systems have the advantage that they provide course-contextualized tools that allow the learners and teachers to interact. When the students leave the LMS to view content on the web they leave all of their learning tools behind.

Our CloudSocial [Severance, C., (2009)] project intends to give teachers, students, and content producers the best of both worlds. CloudSocial allows a hybrid environment where the students spend part of their time within the Learning Management System and part of their time on the web. With CloudSocial, the LMS tools “follow” students around as they surf CloudSocial-enabled sites on the web.

Content authors can enable their web content using CloudSocial by embedding a few lines of JavaScript at the bottom of their web pages as follows:

```
<script type="text/javascript"
  src="http://api.cloudsocial.org/code/ile-
main.js">
</script>
<script type="text/javascript">
  ile_init("1975-5719-90-02");
</script>
```

This JavaScript is very similar to the kind of JavaScript that many web site owners already embed in their pages for tracking purposes using Google Analytics. You can visit www.appenginelearn.com to view the source and see a CloudSocial-enabled web site.

If a user visits the web site and has not been logged into CloudSocial nothing different happens. If the user visits the site after logging into CloudSocial, they see a menu of LMS tools that hover over the site. The menu of tools that appear is a combination of tools selected by the user, the instructor of the course, and the content provider. Figure 4 shows a web site with and without the CloudSocial tool menu.

CloudSocial does not provide the tools. The tools are all launched using IMS Tools Interoperability – so they can come from any source. Each of the menu items can come from a different server.

IMS Learning Tools Interoperability handles the CloudSocial single sign on so there is no need for users to maintain a separate CloudSocial account. The instructor configures CloudSocial and selects the tools that are to appear in the CloudSocial menu and then adds a link to CloudSocial in the LMS system using the IMS LTI capability in the LMS.

When students click on the CloudSocial link in the LMS, they are silently logged into CloudSocial and then redirected to the web page selected by the instructor. The user experience for the student is the same as having a link in the LMS that pops up an external web page – except that they student is logged in to CloudSocial and the LMS tool menu appears on web pages which are CloudSocial-enabled.

The student can navigate anywhere on the web including pages that are not CloudSocial-enabled. The student remains logged into CloudSocial until closing their browser. Anytime they encounter a CloudSocial-enabled site the LMS menu appears over the content.

Since the tools launched from the CloudSocial menu can come from any source that supports IMS LTI, it is possible to have the tools launched by CloudSocial come from the LMS system that did the original launch. Effectively CloudSocial allows LMS tools to “follow” students as they surf CloudSocial-enabled content.



FIGURE 4
CloudSocial Tools Integrated into an HTML Page.

The hope is that CloudSocial will be a facility that encourages an increasing number of teachers and other content producers to simply produce the first version of their learning content in the open and on the web rather than having the “open web version” of their materials separately produced as an afterthought. Until courses can be taught exclusively using open materials on the web, CloudSocial enables the seamless integration of CloudSocial-enabled open content into traditional LMS systems without requiring that the content be imported or reproduced in the LMS.

While the initial use case for CloudSocial is launching from a traditional LMS, it is possible to launch into CloudSocial content from any system that has a concept of users, groups, and roles. Perhaps in the future, we might see informal learning groups forming in something like Facebook and launching into CloudSocial-enabled content from Facebook.

Next Generation LMS Pilot – University of Michigan Medical School

The University of Michigan Medical School is embarking on a multi-year effort to rethink how it uses technology in teaching and learning. The effort intends to solve a number of difficult challenges that are un-met by current-day learning technologies. The goals of the effort include (a) lifelong learning for medical professionals (it is essential to update and upgrade skills throughout their careers), (b) self-regulated learning [White CB, Gruppen LD. (2007)] where the mentors and students collaborate to set learning goals and objectives and the students are given control over the path they take to achieve the learning objectives, and (c) portfolio based approaches where students produce, retain, reflect on, and publish tangible learning artifacts as evidence of their learning.

The themes of this pilot are very similar to how many educators see technology evolving over the next decade. What is unique about this case is that a very agile technical approach is being taken to explore the problem space. We start with the overall view of the high level pedagogy we want to ultimately accomplish and measure all efforts and experiences against that model. Within that context we create a series of small pilot experiments to see what new approaches the students and teachers find useful in the context of their existing learning experiences.

For example, we used an early version of the CloudSocial menu to layer LMS tools on top of existing web-based medical content used in a course for fourth year medical students as shown in Figure 5. The only change was to add the new menu for some of the students in the course.

Several weeks before the experiment we showed students the menu capabilities and asked what kinds of tools they might find useful in such a menu. The students requested tools to manage a to-do list and to allow them to bookmark pages and share those pages with other students in the course. Using the IMS

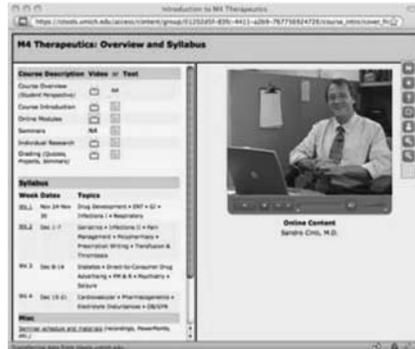


FIGURE 5
Using CloudSocial in Online Medical Materials.

SimpleLTI PHP Framework we developed two new tools based on student suggestions within two weeks and in time for the course to start.

During the four weeks of the course we met regularly with the students participating in the pilot and the faculty member leading the course. Often these meetings would result in requests for new tools or changes to existing tools. As each tool was so simple and written in PHP and running outside of the production Learning Management System – the technical team was usually able to respond to feature requests within 2-3 days and we could build a new simple tool in less than a week.

We also involved the educational technology staff of the Medical School in these meetings as well as the Assistant and Associate Deans of Education at the Medical School. Each different stakeholder had his or her own ideas as to how we could improve the software. While we were making changes to respond to the teachers and students in the course, we also were making changes to meet the needs of the institutional stakeholders and making changes to maintain our alignment with the overall pedagogical approach.

The fact that the entire process is very dynamic and agile and the developers are able to say “yes” to many of the ideas and deliver updated software with very quick turnaround time gives a real energy to the pilots and helps keep all of the stakeholders engaged in the pilot efforts.

Distributing Enterprise Learning Applications - SAMS

While the University of Michigan uses Sakai [Sakai] which is locally branded as CTools as its enterprise learning management system, there are many other departmental sites that serve particular localized needs that the central LMS simply cannot achieve.

An example of this kind of system is Student Assignment Management System (SAMS) [SAMS] that is used in Physics, Chemistry, and Mathematics courses in the College of Literature, Science and the Arts (LSA) at the University of Michigan.

SAMS is a system that assigns and manages grades and homework based on the CAPA [CAPA] Learning Management System. SAMS also is tightly integrated with the classroom response system where students have handheld devices so they can answer questions posed during lectures from their seats. SAMS provides a single point where all of this information is aggregated and made available to the students, professors, teaching staff and educational support staff. In addition, SAMS provides very detailed tracking to allow staff to monitor the performance of the students and Graduate Student Instructors on a daily basis and identify any problem spots with achievement or grading inconsistencies as quickly as possible.

With all of the complexity in the system, it is simply not practical to re-implement SAMS inside of the Learning Management System. Not only would that be a large investment of development effort, once SAMS were part of the enterprise LMS, the LSA staff would lose much of the agility and access to necessary data and reports to allow them to monitor their learning processes.

At the same time – it is very inconvenient and confusing to students to directly visit both SAMS and CTools – they need to learn each system’s navigation from the login process inward to the grades and other information.

The solution was to add support for SimpleLTI to the SAMS software and then add a pre-configured proxy “SAMS Tool” to Sakai. Once the pre-configured proxy tool is added to Sakai it appears to the Instructor exactly the same as if it were a native tool that was built into Sakai. Those teachers using SAMS simply add the tool to their course sites like any other built-in Sakai tool.

When the student selects the SAMS tool in Sakai, Sakai sends the user information, course information, and role information to SAMS using the SimpleLTI protocol. SAMS checks the security signatures and then directs the student into a course-contextualized SAMS dashboard as shown in Figure 6.

Neither the instructor nor the student has any indication that the SAMS dashboard is actually running on the servers in the School of Literature, Science and the Arts. Single-sign-on and roster provisioning is done transparently and securely using the SimpleLTI protocol.

This capability allows a very collaborative approach to building and deploying learning technology across a campus. Instead of the departmental system “competing” with the central LMS, the systems can be brought together to achieve the overall high quality teacher and student experience.

The Learning Management System at the University of Michigan is evolving to the point where it is effectively the student portal – the place where the Univer-

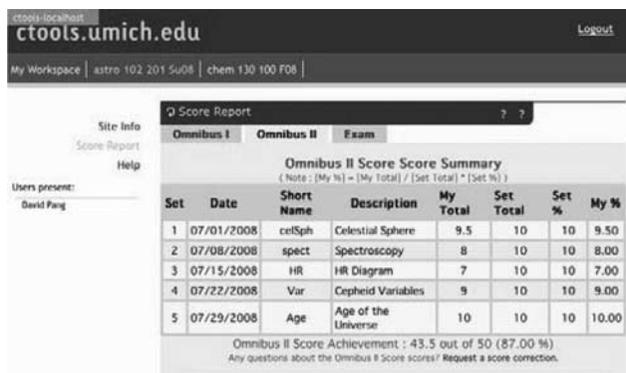


FIGURE 6
SAMS Dashboard in Sakai using SimpleLTI.

sity communicates with students beyond their courses, lectures, and grades. By allowing software from multiple on-campus sources to appear as if it were native in the LMS, it makes it possible for the LMS to work very cooperatively with the rest of the enterprise applications at a university.

Building Interactive Games and Activities – Google Application Engine

When an LMS supports IMS Learning Tools Interoperability, it empowers individual teachers to develop their own applications for use in their course. The goal in this situation is to make the development and deployment of a new application so simple that it works for an individual teacher for one lecture in a single course.

This is the goal of the CloudCollab open source Learning Management System. CloudCollab is a Google Application Engine[AppEngine] application where source code is provided so anyone can run their own instance of CloudCollab. CloudCollab includes a full implementation of a multi-tenancy capable Basic LTI framework. All the aspects of the LTI protocol and necessary multi-tenancy data structures are handled by the CloudCollab framework. The developer only needs to write a simple tool to meet his or her needs.

In the example below, a course is being taught using the book “Wisdom of Crowds” [Surowiecki, 2004]. The first example in the book is where a large crowd guesses the dressed weight of an ox at a fair. The interesting thing is that while the crowd is made up of experts and non-experts alike and their individual guesses are spread over a very wide range, the average of all of the guesses in the crowd is remarkably accurate within less than %1 of the actual weight of the ox.

The application show in in Figure 7 allows the instructor to perform this same experiment in class with students on their laptops within the LMS. The instructor

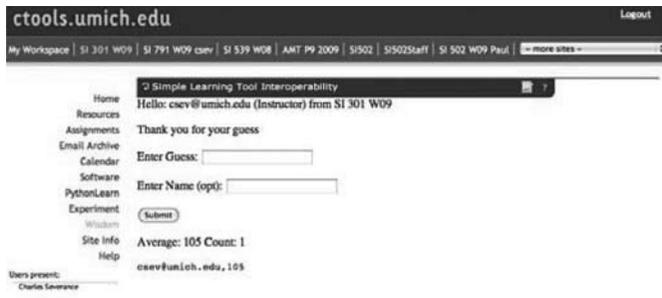


FIGURE 7
Google App Engine Application running in Sakai using LTI.

resets the game and poses a question – the students enter their guesses and when all the students have entered their answers, the instructor presses a button to show the average and all of the good and bad student guesses.

Using the CloudCollab framework, the Python code for the above application is 118 lines long and is running on the free App Engine production-hosting environment provided by Google. This particular application took less than one hour to write and deploy and was developed in the morning on the day of lecture and used in lecture on the same day.

With no further work on the part of the instructor, if left in production, this tool can be used from any LMS in any course, anywhere in the world. The IMS LTI framework built into CloudCollab handles all of the multi-tenancy details and gives each course its own instance of the tool.

In order to give teachers the kind of training needed to write these applications, we have provided a free web site with course materials and recorded lectures to teach Python and Google’s App Engine at www.appenginelearn.com. This material is also covered in the book titled, “Using Google App Engine” [Severance, C., (2009)] published by O’Reilly and Associates in May 2009.

STANDARDS IN PROGRESS

The Basic LTI and full LTI standards continued to progress throughout 2008 and 2009, informed by the SimpleLTI experiments. As of the writing of this paper, the Basic LTI standard is complete and under final review by the IMS Global Learning Consortium Membership. As part of the final review of the standards, a number of implementations of IMS Basic LTI are underway as part of the required multi-vendor interoperability testing:

- An open-source Blackboard building block [Vickers, S., 2009]
- An open-source WebCT PowerLink
- The Icodeon content Player [ICODEON]
- A Basic LTI tool for Sakai
- Basic LTI support for Sakai's Melete content authoring tool
- A Basic LTI Moodle Tool
- MIRLearn from Learnauge/Microsoft

A number of commercial LMS vendors are also in the process of developing their Basic LTI implementations as well and an open-source Basic LTI developer toolkit is available at <http://code.google.com/p/ims-dev/>.

The IMS Common Cartridge 1.1 Specification now includes support for representing Basic LTI links within a cartridge. The IMS CC 1.1 specification will be released about the same time as the IMS Basic LTI Specification.

The full LTI specification continues to progress and is expected to be complete early in 2010.

OPPORTUNITIES AND CHALLENGES FOR TOOLS INTEROPABILITY

The work and efforts in this paper reflect a two-year effort working with prototypes of IMS Learning Tools Interoperability with many of the major stakeholders in the LMS market place. The response to IMS LTI has been universally positive from the vendors of LMS software and external tools who have participated in these projects.

By producing a standard for tool integrations, IMS LTI has the potential to greatly reduce the costs for both LMS vendors and external tool providers. Furthermore, LMS vendors are eager to have a widely-used plug-in capability to encourage users to write as many extensions as possible. By encouraging a rich ecosystem of tool development for tools that can be easily integrated into any LMS, it takes a lot of pressure off the LMS vendors to continuously expand their feature sets to meet increasingly narrow market needs. By "out-sourcing" the non-core software, LMS vendors can focus on the performance, reliability, scalability and usability of the core tools that are heavily used. This allows LMS systems to meet the needs of being an Enterprise application.

As with any new technology, there will be issues and challenges to overcome.

- When tools come from multiple sources or vendors, they naturally will have different look-and-feel and user interaction patterns. Once we have the standards in place that allows tools to function, we will need to turn our attention to improving the user experience as the user transitions from LMS-provided tools to externally provided tools.
- When tools come from multiple sources, those systems will have varying performance, reliability, and scalability. If a user clicks on an external tool and the tool is down, the user will likely blame the LMS system for the failure of the external tool. Tools from multiple sources also make training and technical support more challenging.
- Most countries have strict regulations regarding the privacy of student learning data [FERPA]. When an instructor uses an external tool in a course and sends student information to that tool, he or she may be violating student privacy regulations. In actuality, teachers at the edge of innovation already disregard these regulations when they use services like Blogger[BLOGGER] and place student identity information in those services. One solution will be to host the external tools on on-campus servers where student privacy can be assured.

All of these issues are a natural side effect of the flexibility and options that IMS LTI enables. In a sense, the purpose of IMS LTI is to get us beyond the point where integrating tools into LMS systems is expensive and impractical. If we solve the technical problems so effectively that hosting tools outside of the LMS is commonplace, we are at the point where we are facing the truly important issues that must be solved to take the next step in using technology for teaching and learning.

CONCLUSION

In this paper, we have tried to show motivating examples to demonstrate the potentially transformative effect the IMS Learning Tools Interoperability Standard in the next decade of teaching and learning. While some of the examples reflect simple one-off experiments with commercial vendors that may take quite some time before they appear in shipping products, all of the examples in this paper are working code – there are no “mockups”.

We have seen how IMS Learning Tools Interoperability in conjunction with IMS Common Cartridge can significantly improve the interaction between publishers, LMS systems, teachers and students. By reducing the cost of integration

between these systems to virtually zero – we can focus on how to develop and use content as a service and software as a service – and how these can best improve teaching and learning. The entire process of distributing and reusing content can become much more agile and have a far greater impact on improved teaching and learning. As “software as a service” evolves it will be possible for a company or development team to specialize in one aspect of learning software and provide software that runs in any LMS system which supports IMS LTI.

IMS Learning Tools Interoperability also enables whole new ways to integrate between applications within the same Enterprise. This allows us to create a learning environment where there is not “only one way to do it”. Educational institutions can create a diverse learning environment for the student without forcing all learning software into the same system. IMS Learning Tools Interoperability also has the ability to greatly improve the use of multiple learning management systems in production at the same time and to provide a way to have a graceful transition from one enterprise LMS to another enterprise LMS if an institution so chooses. All of this flexibility leads to a better overall technology offering for our teachers and students.

Using IMS LTI also puts the technology back in the hands of teachers and allows rich experimentation with new ideas without requiring a wholesale replacement of the Enterprise Learning Management System or allowing constant experimentation on the production LMS servers. IMS LTI allows the production systems to be run conservatively and surrounded by a set of pilot tool servers or even a server for an individual instructor’s personal tool hosted on Google App Engine.

While it is not clear exactly what we will think of as a “Learning Management System” in 2020, it is likely that the system will take a service oriented approach to its tools and data. We hope and expect that continuing experimentation around IMS Learning Tools Interoperability will serve to help us move down the path toward the Learning Management System of 2020.

Thanks

This work would not have been possible without support from the IMS Global Learning Consortium, the William and Flora Hewlett Foundation, the University of Michigan Medical School, Microsoft, Pearson Education, McGraw-Hill Higher Education, ANGEL Learning, Blackboard, Wimba, Sakai Foundation, Google Summer of Code, and the Software Engineering for Information Systems Group (GESSI) at Universitat Politècnica de Catalunya. The authors also want to thank the reviewers of this paper for their helpful guidance and suggestions.

REFERENCES

- AppEngine, Google App Engine, Retrieved– January 11, 2009 from <http://code.google.com/appengine/>
- BBProxy, Blackboard Learn, Retrieved– January 11, 2009 from <http://www.blackboard.com/>
- BLOGGER, Retrieved October 19, 2009 from <http://www.blogger.com/>
- Booth, A., Severance, C., (2007) Sakai Video Report: Andrew Booth - Bodington Project, Retrieved– January 11, 2009 from <http://www.dr-chuck.com/media.php?id=80>
- CAPA, The Learning Online Network with CAPA, Retrieved– January 11, 2009 from <http://www.lon-capa.org/>
- CNX, The Connexions Project, Retrieved– January 11, 2009 from <http://cnx.org/>
- Facebook, Facebook Developers Guide, Retrieved– January 11, 2009 from <http://developers.facebook.com/>
- FERPA, Family Educational Rights and Privacy Act (FERPA), Retrieved October 19, 2009 from <http://www.ed.gov/policy/gen/guid/fpco/ferpa/index.html>
- Fielding Roy, T., (2000), Representational State Transfer, Retrieved– January 11, 2009 from http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- Google, (2008) Google Summer of Code – Retrieved October 27, 2009 from <http://code.google.com/soc/2008/sakai/about.html>
- Holland, B., (2008), The MIR IMS Runtime, Retrieved– January 11, 2009 from <http://www.codeplex.com/MIR>
- ICODEON, Icodeon Content Player – Retrieved October 27, 2009 from <http://www.icodeon.com/>
- IMS, IMS Global Learning Consortium, Retrieved– January 11, 2009 from <http://www.imsglobal.org/>
- IMSCC, – IMS Common Cartridge, Retrieved– January 11, 2009 from <http://www.imspjproject.org/commoncartridge.html>
- IMSLTI, (2009) IMS GLC: Learning Tools Interoperability, Retrieved– October 19, 2009 from <http://www.imsglobal.org/toolsinteroperability2.cfm>
- Katana - McGraw Hill Higher Education Katana, Retrieved– January 11, 2009 from <http://www.mhlearningsolutions.com/katana/>
- MathXL - Pearson MathXL, Retrieved– January 11, 2009 from <http://www.mathxl.com/>
- OAuth - An open protocol to allow secure API authorization in a simple and standard method from desktop and web applications., Retrieved– January 11, 2009 from <http://oauth.net/>
- OCW, MIT Open Courseware, Retrieved– January 11, 2009 from <http://ocw.mit.edu/>
- Sakai, Sakai Project, Retrieved– January 11, 2009 from <http://www.sakaiproject.org/>
- SAMS, Student Assignment Management System, Retrieved– January 11, 2009 from <http://sams.lsa.umich.edu/>
- Severance, C., (2008) Simple Learning Tools Interoperability, Retrieved– January 11, 2009 from <http://simplelti.appspot.com/>
- Severance, C., (2009) CloudSocial – Inverting the relationship between Content and the LMS, Retrieved– January 11, 2009 from <http://www.cloudsocial.org/>
- Severance, C., (2009), Using Google App Engine, O'Reilly and Associates, May 2009, <http://oreilly.com/catalog/9780596800697/>
- Severance, C., Hardin, J., Whyte, A., (2008) The Coming Functionality Mashup in Personal Learning Environments, *Interactive Learning Environments (Journal)*, Volume 16, Number 1, April 2008
- Surowiecki, J., (2004), *Wisdom of Crowds*, Doubleday.
- Vickers S., (2009), SPV Software Products, Retrieved October 28, 2009 from <http://www.spvsoftwareproducts.com/>
- White CB, Gruppen LD. (2007), *Self-regulated Learning in Medical Education*. Association for the Study of Medical Education: Understanding Medical Education.