LETSI LEARNING - EDUCATION - TRAINING SYSTEMS INTEROPERABILITY

Content Interoperability is the Wrong Problem to Solve

January 25th, 2010 *by Avron Barr*

Last week I attended a meeting in Washington to discuss the various content interoperability standards for elearning materials. Many of the well-known standards organizations where at the table along with a bunch of smart folks from the government. While the government's intentions were not made entirely clear, there was some talk of a proposal to create a repository of open learning materials for use by teachers and students nationwide. One could imagine that deciding on a single content format could be part the plan, at some point.

Content interoperability standards, like SCORM, AICC, and Common Cartridge, all attempt to assure that course materials from conformant applications will load into other conformant applications which will in turn deliver a consistent student experience across systems. SCORM and AICC also include standard formats for reporting back data on student completion and scores.

Today's elearning world has been shaped by standards. Before SCORM and AICC were created in the 1990s, elearning materials were developed and delivered within the same application suite. The content interoperability standards separated the creation of learning materials from the delivery application and thereby created an independent content development industry, opened the possibility of sharing development costs for broad-market materials, and accelerated elearning adoption by reducing the risks of vendor lock-in.

Problems with Content Interoperability

Today, we take for granted that there is a separate content publishing industry. A community of practice typically specifies that all systems and content purchased must conform to a standard, hoping thereby to achieve the economic and pedagogical benefits described above. There are some problems with today's status quo, however.

Most importantly, content interoperability never worked that well when you examine the details of rendering the student's learning experience. The only content that is easily portable across delivery systems are the simplest, lowest-common-denominator course materials. Innovative features from one system are not portable to another, since the innovative features are not part of the standard. Thus the standard itself, while opening up the market, can retard innovation. By requiring conformance to a content interoperability standard, a community can preclude the procurement of the most innovative learning materials.

Innovation is key to realizing elearning's true potential on the web – we have a long way to go. In various corners of the elearning world, teachers and content creators are exploring immersive learning environments, mobile platforms, multi-student collaboration activities, online tutoring, and intelligent tutoring systems. They are using online activities in constructive and discovery-based learning scenarios that fall outside of traditional computer-based instruction. However, to achieve portability, today's content standards make 20-year-old assumptions about pedagogy, about the student experience, and about publishers' business models that stem from a time before computers were connected by networks.

For example, the current standards assume that content is managed and rendered for the student by the central learning management system (LMS). The LMS concept stems from the client-sever era of computing – a monolithic application and data silo for managing student rosters; controlling access; importing and managing content; organizing materials; assigning lessons; rendering the student learning experience; tracking student progress; and sometimes even scheduling classes and classrooms. While there is always a need for record keeping, in the age of web-based software, there is no reason that the management system should also manage content and deliver training.

Current work on Common Cartridge and SCORM is trying to retrofit those standards by allowing a "package" to point outside of the LMS to a web-based activity. This is a hack – there is no need for the LMS to manage the content or for content to be imported to the LMS in the first place. It is hard to imagine that the most cutting-edge, complex, and data intensive online learning activities would ever be portable across LMSs. And how about the teachers and learners around the world who don't have an LMS?

Suppose the Content Stayed Put?

Imagine an entirely web-based model of delivering the student experience, whether that experience is a talking-head lecture or a multi-player online game for learning arithmetic. Suppose that as new learning materials are created, in whatever authoring environment, they were posted to a website, like the flickr.com site posts your photos:

- The website could deliver the intended student experience exactly as the creator intended. The experience could be a lesson, a quiz, a team chem-lab experiment in a virtual world, or a practice run through an air traffic control simulation.
- The learner may use a browser to access the web-based materials, or a software client (like Second Life), or an app on an iPhone or ebook. Teachers, tutors, and parents could have their own apps for assigning, tracking, reporting, and even observing student activity on the web.
- A registry could aggregate the offerings and offer not only Google-like search and education-specific metadata (e.g., grade level, learning objectives), but also peer reviews and Amazon-like star ratings and recommendations. The government could create a registry of open learning materials. Commercial publishers could offer similar online portals and could continue to establish institutional accounts with major customers.
- Data about the student, learning objectives, resource availability, and a host of other things useful to tailor the learning experience could be queried from an LMS and other

systems. (LETSI's pilot software project explored the use of web services to exchange such data, based on existing standards.) Similarly, completion and performance data could be returned to an institutional LMS or to a tutor's iPhone app, for example.
- For students who are not always online, the content, player, and score-reporting module could all be downloaded in advance.

Standards promote innovation by separating domains of innovation: for example, methods of producing and distributing electricity vs. appliances that plug into a standard wall outlet. Early elearning content interoperability standards assumed that there would always be an institutional learning management system to deliver the student's experience, and therefore the interoperability problem was packaging the content in a standard format that each LMS could interpret consistently.

In web-based learning architecture proposed here, the LMS could still keep track of institutional data: rosters, assignments, and content contracts – but it would no longer manage the content or render the student's experience. In some markets, LMS functionality could evolve over time into range of applications for planning lessons, shopping for content, assigning activities, tracking student progress, evaluating materials, and even coaching students online. But there is no distribution of content, no requirement for portability, and no standards-imposed restriction on the nature of the learning experience.

And the content interoperability problem disappears, replaced by a variety of data sharing protocols about resources (materials), competencies (learning objectives), activities (learner performance), and people (students, teachers, parents, tutors).

Tags: architecture, interoperability, learning 2.0, LETSI, LMS vs. Content, SCORM, SCORM 2.0, web services
Posted in LETSI News, architecture

Comments, excerpts only

Standage Says:

January 26th, 2010 at 7:40 am

It’s true we must standardize the interaction of systems, but Avron here is trying to help us see a different vision of what should actually be standardized. SCORM standardized the communication between static, local content and the LMS, and while people have always been free to build LMSs based on an open, modular architecture, it hasn’t really happened much, has it?

I wouldn’t say that the content interoperability issue is the WRONG problem to solve as much as it is INSUFFICIENT. The storage and delivery of content is such a small portion of what an LMS needs to do. LMSs also need to incorporate personal data, HR data, performance/competency data, content, content metadata, and sequencing/orchestration information. The trend in the past has been to create a one-stop-shop LMS that does all this, but there is plenty of existing software to satisfy each of these individual requirements. What we need is the framework necessary to integrate software components of these various types–this will no doubt involve standardizing communication between different software component types.

Our biggest challenge will be defining a framework that is concrete enough to provide a reasonable amount of interoperability while abstract enough to be useful to a wide community.

Aaron Silvers Says:

January 26th, 2010 at 10:33 am

I think registries are a notion whose time is finally coming. They solve a problem that was hard to identify years ago, even though the notion is far from new.

I’d like to shine a light on something I found out about called CloudSocial (http://cloudsocial.org/), which is based on IMS Simple LTI, to some degree.

I lean a bit with Mike on the structured/unstructured content side of things — I would love a structured content spec that determines the way forward, but there will always be a great amount of content that is unstructured and there needs to be ways to leverage it. Registries, to me, make a lot of sense in doing so. Projects like Mike’s proprietary SCORMCloud, LETSI’s Run-Time Web Services and perhaps this CloudSocial deal highlight still other ways to get content in.

This still doesn’t so much as carve out what the future holds… it’s a nominal advancement in cleaning up what everyone knows is out there right now given the systems that are in play.

In dealing with content, I largely agree with your post, Avron. I think registries, if tapped, become powerful and helpful in restoring content to a proper perspective in relation to [using your words] “resources (materials), competencies (learning objectives), activities (learner performance), and people (students, teachers, parents, tutors).” We won’t need the LMS as it exists today.

Tom Wason Says:

February 1st, 2010 at 4:35 pm

LETSI encapsulation of SCORM IMS
Encapsulation of Foundational Sets

Avron's concept of LETSI as moving past the current models of programmed instruction (by whatever name) into e-Experiences is interesting. It brings up a key point: how does LETSI relate to the existing standards? One ignores SCORM, IMS and IEEE-LTSC at one's peril. There is already good work afoot to build a bridge to LTSC relative to the CMI standard. What should be done about the other standards, SCORM and IMS?

Some time ago, as memory serves me, Crispin suggested a split "orchestration" specification. Some of it would serve the traditional programmed instructional needs; other instructions would serve what Avron calls "experiences." Furthermore, Crispin's model separates the two functional models. I suggest that we separate these capabilities even more strictly through the encapsulation of the older specifications. SCORM has a data model and a sequencing model. To attempt to build on these is extremely constraining: the current SCORM model is simply different from the experience model (eXperience?). The new model should take over just where the old one becomes difficult. Those who want to use the SCORM model extensively can use it as is; they have no need for a LETSI model.

Consider a stripped down SCORM model. This would have some minimal set of sequencing capabilities. These capabilities would support the most commonly used instructional strategies. There would be a few basic strategies. As starters consider a linear flow and a simple 3-way branch. You can build a lot with a 3-way branch; it may be that some of the derivatives would have separate names, but under the covers they would be accomplished using a 3-way branch (e.g., a two way branch quiz only provides answers a or b). This form is often used in direct instruction (e.g., the National Institute for Direct Instruction, http://www.nifdi.org/15/). Direct instruction presents small chunks with quick checks and appropriate instructional responses. It is a common model. More complex instruction can often be accomplished with complex modules such as Flash.

A minimal SCORM set is not as far-fetched as it might seem. The Korean Education & Research Information Service (Keris, http://english.keris.or.kr/es_main/index.jsp) promotes the use of only three SCORM-based models for all—I repeat, all—K-12 education. I have been exploring issues in encapsulation with SCORM based on the question: if I ask you for a SCO, what do you send me? I have suggested a pretty strong encapsulation model of SCOs in my recent Dr. Tom Guide to Reusable Content Objects (http://www.twason.com/drtomreusablecontentobjects.html). The idea is to separate the authoring and content management part from the deployment of courses and their pieces through encapsulation. Similarly, let us separate the SCORM part of a course—including its manifest—from the main body of the LETSI specification. It is close to being an encapsulated object within LETSI. A course that uses only the reduced SCORM version would simply be the LETSI outer instantiation shell within which is the encapsulated SCORM package. A fundamental SCORM model can be more easily managed with a simple tool; it might be attractive to tool vendors to be able to provide a simple LETSI SCORM tool.

So, to continue… LETSI would define a minimal SCORM set that it will require LMSs (or services) to support. This is a fundamental SCORM. This set need be only “pretty good.” Perfect is not nearly as important as small size. LETSI SCORM would be a compliant subset of SCORM. LETSI would create a hole in the LETSI model into which fundamental SCORM could be placed. LETSI compatibility would mean that the subset of SCORM that LETSI has defined must be supported in a compliant system. By keeping to a simple model this would not be onerous. LETSI would define a fundamental model and create a clear guide on the use of that foundation. If people wanted to put more in the “SCORM hole” there would be no guarantee that it would run. Or that it wouldn’t. A user could decide that he or she only wanted to use the foundational LETSI SCORM and wanted nothing else. The LETSI package/method should allow a quick extraction using a simple utility. A similar approach could be defined for IMS Common Cartridge. The objective is to support some well-encapsulated legacy capabilities, but to keep scope small.

Then LETSI can go…

…into the cloud.

–Tom Wason