

SCORM 2004 3rd Edition Tool

A little history

For a while now, there's been a prototype version of a Scorm Player available for Sakai. It was originally developed by Jon Gorrone at UC Davis for Sakai 2.1.x, and has recently been ported to 2.4.x by Sacha Leprêtre and others at le Centre de Recherche Informatique de Montréal (CRIM). This version was based on ADL's reference implementation of SCORM and the tool was written in JSF. It stores its data primarily by serializing the important ADL objects (the ScoDataManager, the Sequence Activity Tree, for example) to disk. Jon's version used an applet to provide communication between the client and the servlet, which was not ideal since it raises some deployment issues. The CRIM port (apparently) replaces this applet with AJAX code.

The new player

Here at UC Davis, we're currently developing a new version of the Scorm Player that we hope re-casts Jon's original prototype in a form that is more flexible and extensible. One of the primary motivations for this is the idea that Scorm is most effective when it's possible to do some data mining and evaluation on the scores produced by a user's session with the player.

Some highlights:

- The tool is written in Wicket and makes use of AJAX to replace the applet code
- The big data objects are stored directly in the database, where scores will be accessible to later tools
- The ADL code is now broken out into a more reasonable division between api and impl, so the services can be exposed beyond the tool itself in Sakai
- The Scorm content packages (zip files) are stored in Resources via the Content API, and make use of Content Hosting Handlers to access the underlying resources

A Roadmap for Sakai/SCORM Development

Jul 10, 2008:

The comment below from Sept 2007 still applies. We would be very happy to have any assistance that others in the community might be willing to give.

Sept 12, 2007:

Seems like it would make some sense to sketch out a bit of a roadmap in the next few weeks, taking into account the various efforts out there to develop a Scorm Player for Sakai as well as the community-at-large's requirements. I've been getting pinged on Scorm a bit more than usual lately, and as we get closer to having a version here at Davis that we're comfortable building off of, it will be easier for us to respond and communicate our direction, as well as incorporating others.

Ideally, I'd like to parcel out the work into reasonable chunks and put some kind of a schedule on when we're aiming to tackle each chunk so others can jump in and help out where it makes the most sense for them.

Status

Jul 10, 2008:

Currently we're working on replacing the non-JPA Hibernate persistence layer with a JPA compliant one. This is in response to some significant bugs we've seen on certain content packages because of problems with the mapping of nested objects in the current Hibernate configuration.

We're still in need of some help doing some load testing and optimization of the existing code to ensure that we can support large numbers of concurrent users.

Also outstanding is Gradebook integration.

Dec 13, 2007:

After some discussions with people at the Newport Beach Conference, it sounds like there is significant interest in having a new tool available in the next month. Sacha LePretre at CRIM has agreed to collaborate on moving the wicket player toward Portlet compatibility, with the eventual goal of being able to run it inside of Pluto.

Currently, the version we have is not implementing global objectives (or at least there's no UI for users to enter global objectives). Also, the SSP state persistence stuff is not completely implemented. Please get in touch with me (James) if this is going to pose a near term problem for any early adopters.

We're continuing to clean up the code and implement the ui for instructors to view the result data from each user attempt to playback a Scorm module/content package. Things are processing rather rapidly right at the moment, so if you're working off the Contrib code you'll probably want to do daily updates. We're going to do our best not to break it in trunk, but you never know...

Remember that you want to run the current SCORM.2004.3ED.RTE trunk code against the Sakai 2.5.x branch, not trunk.

Sept 12, 2007:

We're currently (as of September 12th, 2007) got a prototype/proof-of-concept running against the current Sakai trunk, which we hope to release as contrib around the 2.5 release date. There's still some significant work to be done rounding out rough edges, cleaning up the data model, improving the Content api communication, and of course, QAing.

Once this work's completed, the immediate goal is to then start focusing on talking to Gradebook, pulling out scores, etc.

The version that's in contrib is a little messy versus the one in our local repo. I need to scan through it and delete a bunch of files. Also, I've just introduced a dependency on a new contrib project for wicket that's not checked in yet – waiting on the contrib space itself.

Note that the current version also has a dependency on the contrib project entity-broker – this is not really essential to the current functionality, but eventually we'd like to decouple from the tool context and allow users to launch directly from a content package in resources, which will probably require entity-broker.

Mar. 19, 2010:

Not much has happened over the past few years w.r.t the SCORM 2004ed3 player. Edia made several changes and bug fixes to improve the code, but were unable to get commit access. As such, they posted their patches to their own SVN server. Additional notes on [Edia changes](#) are available.

Recently, there has been talk about reviving development effort on this SCORM player. Part of the inspiration for this comes from SoftChalk, who have a product called LessonBuilder that (as of version 6.0) has the ability to publish lessons directly to Sakai either as zipped web pages or SCORM packages. LessonBuilder includes support for scored interactive exercises. While the SCORM player does capture this data, ideally it would be posted to the Sakai gradebook in an appropriate place.

Permission has been secured from Edia to merge changes back into the David work. That clears the way to moving forward with improvements to the player. The first step should be to merge Edia changes back into the contrib code base. This will be more difficult now that James Renfro has moved on to other activities and is no longer available to consult.

A BOF and panel discussion is planned for the 2010 Sakai Conference in Denver.