

# Sakai SCORM Support

June 4, 2010

The SoftChalk LBGateway application that supports integration of LessonBuilder with Sakai was designed and implemented with support for the UC Davis SCORM player. During the course of implementation, several problems were noted:

1. SCORM packages are stored in an expanded form via the Content Hosting Service.
2. SCORM packages can only exist at the top level of a site collection.
3. There were some serious problems in CHS w.r.t renaming in Sakai 2.5.x that were fixed in 2.6.x.
4. SCORM package import and registration is slow.
5. Expanded SCORM packages are difficult to identify in CHS collections (later fixed with special file).
6. While the David player collects use data, it is not easily accessed.
7. No gradebook integration.
8. No assignment integration.
9. Requires Wicket module.
10. Davis is no longer supporting SCORM development, in fact, no one is.

While many of these issues were addressed with work arounds, the restrictions and lack of support made SoftChalk SCORM support increasingly difficult. As such, support was dropped for in it in the LessonBuilder 6.0 release.

I looked into making some small improvements to the Davis SCORM implementation. Unfortunately, the code had been forked and checked into an SVN server maintained by Eida, a Dutch Sakai consulting company. After months of emails, they agreed to allow their fork to be merged back into the Sakai contrib branches. By that time, SoftChalk had de-supported SCORM packages for Sakai, so there was no funding to actually do this merge. Getting the SCORM software back into Sakai contrib is necessary to more it forward at all. The departure of Michael Korkuska as Sakai ED further complicated this.

In my opinion, the David SCORM player needs considerable work. The Wicket UI front end is marginal and should be migrated to JSP, JSF, or RSF - being the most commonly used UI technologies in Sakai. AJAX/JSON might be another approach that would facilitate migration of the code base to Sakai 3. I think the entire front end should be ripped out, re-designed, and completely re-implemented based on user requirements.

In terms of the back end, James Renfro did his best at a very tough job. Migrating the SCORM reference model to make it work with Sakai kernel services was a BIG job and it took him a year to do it. Unfortunately, James is no longer with UC Davis and he took all design and implementation knowledge with him - there are no documents and very sparse JavaDoc to guide future developers.

Someone needs to dive into the SCORM player code base do the following:

1. Rationalize the code factoring
2. Document it's architecture and design.
3. Review and improve public APIs.
4. Fix known bugs reported in Jira.
5. Leverage newer Content Hosting Service capabilities.
6. Do a basic gradebook integration.

Personally, I'd add replacement of Hibernate for the Sakai SQL service, but that's a judgement call that all may not agree with.

At a guess, I think the above represents several months worth of work for a competent Sakai service/tool developer. That's not cheap, but Sakai as a whole is disadvantaged by not having integrated, open source support for SCORM. This is a very common checkbox item in LMS RFPs. There are commercial SCORM players available for Sakai, the most common being the Icodeon Player, but that's on a per-seat license basis that many schools can't/won't afford.

From SoftChalk's point of view, a wait and see approach is probably best. The scope of the work involved in larger than SoftChalk is likely to recoup. Since SCORM support isn't provided by the Sakai community, no one is using the David player. SoftChalk may wish to experiment with publishing to Icodeon (etc). Meanwhile, publishing Zip packages works quite well (except for a few, soon to be fixed bugs). SoftChalk may also wish to explore gradebook integration with uploaded Zip packages.

Feb. 10, 2011

I'm trying to get a feel for the status of SCORM in Sakai these days. While I read the recent thread about Wicket bugs and fixes therein, I haven't seen much activity to improve SCORM support itself or to address any of the outstanding issues, though perhaps I'm wrong.

I am asking on behalf of SoftChalk. As you probably know, SoftChalk has an application (LessonBuilder) that creates interactive lessons for students. These lessons can be packaged in one of three ways: ZIP, SCORM, and Common Cartridge.

As part of the integration work that I did for SoftChalk, I originally included support for the Davis SCORM player. After extensive testing, SoftChalk and I agreed to back out SCORM support for the following reasons:

1. Code was not stable and not well maintained
2. Target location for SCORM packages was restricted to top level of Resources.
3. Sakai integration was limited, at best (no assignments, syllabus, etc. integration)
4. Limited implementation of SCORM communication protocols.
5. No grade book integration.
6. Source code was not in the Sakai SVN repository (rather, in Edia).
7. Limited or no support from UC David and/or Edia.

SoftChalk is reconsidering support for SCORM in Sakai. Has the code been improved in any substantial way in the past year or so? Have any of the above issues been addressed? Would you recommend that SoftChalk use the Davis/Edia player? Are there any plans on the table to maintain/improve this player?

---

Thank you for your message. On behalf of Edia I would like to give some comments and answers on the subject. In general, the answer is: yes we do support.

At Edia, we picked up the trail for SCORM in Sakai in 2009 and we decided to take the UC davis SCORM player and improve it, mostly because we thought it would positively influence the adoption of Sakai in the Netherlands. Along with fixing a number of issues we also started working on the integration with the Sakai Gradebook which is fully functional at this time. The past months we have been focusing on making the player compliant with SCORM version 1.2. Three months ago, the Scorm player was awarded a 100% score at a dutch plugfest (<http://www.edia.nl/en/scorm-plugfest>)

During this whole period we have provided support to people that have either contacted us directly, or through questions on the Sakai mailing lists. We have also opened a repository and confluence page for those who want to install the Scorm player themselves.

I think we can say that the Edia version of the Scorm player is under continuous development. In relation to the issues you mention:

1. Code was not stable and not well maintained
  - we believe we have a stable and maintained version for 2.5, 2.6 and 2.7
2. Target location for SCORM packages was restricted to top level of Resources.
  - not adressed yet
3. Sakai integration was limited, at best (no assignments, syllabus, etc. integration)
  - not adressed yet
4. Limited implementation of SCORM communication protocols.
  - not sure what you mean?
5. No grade book integration.
  - this has been adressed and is functional
6. Source code was not in the Sakai SVN repository (rather, in Edia).
  - the code for the Edia version is in Sakai SVN, following our previous contact in march 2010
7. Limited or no support from UC David and/or Edia.
  - We provide community support where we can, meaning we do not allocate structural resources to the Scorm player. Of course, as a company, we are always open to discuss commercial services we could offer.

I hope this answers your question to some extend. Best regards,  
Jaeques Koeman



Jaeques Koeman  
jaeques.koeman@edia.nl  
Edia - Educatie Technologie  
Korte Prinsengracht 42hs | 1013 GT Amsterdam  
T 020 716 36 12 | F 020 716 36 13 | M 06 47 106 796 | [www.edia.nl](http://www.edia.nl)

---

Thank you for your response, Jaeques. Overall, your reply is very encouraging. Some replies to your comments/questions:

> 2 ... restricted to top level of Resources

Any plans to allow SCORM packages to be installed anywhere in a Resources collection hierarchy?

> 4. Limited implementation of SCORM communication protocols.

SCORM has a communication protocol (based on AICC standards) that allows data to be sent from the client browser back to the LMS for storage/processing. My understanding was that this protocol was never fully implemented. Perhaps that situation has improved.

> 5. Grade book integration

Ah, that's excellent news. It was a bit of a show stopper for SoftChalk>

> 6. the code for the Edia version is in Sakai SVN

Also very good news. It clarifies licensing issues, IMO.

> 7. We provide community support where we can

Very good to hear. While I have a lot of respect for Edia, there was a long period where nothing was getting done. It gives me a feeling of confidence that you are putting some level of effort into maintenance and bug fixing.

In the future, should I contact you or Roland concerning Sakai SCORM?

---

The following bugs were fixed in the past year:

Jira No.	Date Fixed	Ttitle
SCO-41	27-Oct-2010	Running ADL Examples: All binary files (Images/Flash etc) corrupted
SCO-49	27-Oct-2010	Uploading new SCORM object spams the user for each resource.
SCO-50	27-Oct-2010	SCO-49 Update code when unpacking zip not to send notifications
SCO-60	27-Oct-2010	Uploading scorm package generates email notification for each file
SCO-62	09-Dec-2010	Unicode unzip in scorm2004
SCO-63	27-Oct-2010	Wicket causes JVM crash on java 1.6
SCO-64	28-Oct-2010	Serialization Errors
SCO-65	09-Dec-2010	Ignore the value of the field cmi.success_status when synchronizing with gradebook.

---

Spoke with Sue Evans, President of SoftChalk. While she agrees that the above is good news for SCORM support, SoftChalk has decided to evaluate the Edia Player for compatibility with SoftChalk lessons, grade book support, etc. If demand for this increases in SoftChalk's Sakai customers, SCORM integration will be included again.