Draft copy. Please check back later for final version. Please do not cite or quote without contacting the authors first (albert@DLS.au.com)

Single Copy Re-use of Sharable Content Objects

Albert Ip
Digital Learning Systems

Ric Canale
The University of Melbourne

Abstract

This paper puts forward a generic solution to the problem of re-use of Sharable Content Objects (SCOs) from multiple content repositories without making new copies of the SCOs. The solution has already been partly implemented by Open Learning Australia using Avilar's WebMentor as the Learning Management System (LMS) and Harvest Road's Hive as the Learning Content Management System (LCMS). The use of this technique to enable multiple LMSs to access SCOs stored in a single LCMS has already been demonstrated at Plugfest 7, a forum sponsored by the Advanced Distributed Learning Co-Laboratory. This paper further develops previous work, provides technical specifications and proposes an extension to the IMS content packaging specification to encourage adoption of this approach by LMS vendors to enable delivery of SCOs from multiple repositories.

This technique has little overhead in terms of SCO development, management of courses and installation of courses. The model also enables the implementation of secure fetching of SCOs (via server-side authentication techniques) and hence is compatible with enforced digital rights management. The model includes a proposal for a technically minor extension to the content packaging specification. This is one of several proposed extensions that pose wider implications for the potential use of SCORM within higher education and training. By addressing key issues of pedagogical extensibility of SCORM and feasibility of large scale re-use of learning objects through SCORM extensions, the authors conclude that SCORM is sufficiently robust in its design to have potential to become the dominant open standard for content interoperability across all educational and training sectors.

Common acronyms used in this paper:

ADL	Advanced Distributed Learning
AICC	Aviation Industries CBT Committee
LCMS	Learning Content Management System
DTD	Document Type Definition
EML	Educational Modeling Language
IEEE	Institute of Electrical and Electronic Engineers
LMS	Learning Management System
LTSC	Learning Technology Standards Committee
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
XML	Extensible Mark-up Language
XSL	Extensible Style-sheet Language

Introduction

SCORM (Sharable Content Object Reference Model) is the most successful and widely accepted reference model of the IMS specifications for content packaging and AICC computer managed interface specification (also being part of the IEEE LTSC effort). The SCORM has been developed by Advanced Distributed Learning (ADL, http://www.adlnet.org/), an initiative of the Department of Defense, USA. Its development is informed by the IMS specifications (IMS, 2001), AICC specifications (AICC,) and the IEEE standards setting process (IEEE, 2001). The major LMS vendors currently support SCORM or at least have publicly announced their intention to fully support it. This means that SCORM compliant content is fully deliverable without modification on a number of different LMS.

A key design objective for SCORM is to enable re-use of course material. Working within the current specification, course designers wanting to re-use existing “third party” SCOs would need to make copies of SCOs in their courses. All the files used in a SCORM course are packaged together using the IMS content packaging specification. The organisation of the course and the relative location of the SCOs (including third party SCOs) are defined by the manifest file within the package. In this way multiple copies of SCOs are scattered in different courses running in the same LMS. Assuming copyright clearance has been obtained, the act of copying a SCO and its assets to be included in a SCORM package for another course is a trivial exercise. However re-use of SCOs by creating multiple copies has some drawbacks:

- SCO Maintenance.
  In the case of popular SCOs that are often used by many course designers, having many independent copies of a SCO being used by different courses makes it impossible to manage and maintain the SCO. Ideally, the creators of a SCO would want to continue to update the SCO and have the updates propagated to all courses using the SCO automatically and under some form of version management.
- Digital Rights Management.
  SCO developers would also want to be informed about the extent and quality of usage by other course designers for reasons of intellectual property rights management (referred to as Digital Rights Management DRM hereafter). For SCOs being made available on a commercial basis, tracking usage would also be highly desirable.

To overcome these problems, an alternative model is needed that enables re-use of content without making copies. In the proposed model, the following assumptions are made:

- content is in the form of SCOs or assets;
- SCOs and assets are lodged within a LCMS which handles digital rights management and version control;
- SCOs may be used in courses with different look and feel characteristics. To make re-use of content pragmatic and cost-effective, the "look and feel" of content should be determined by the template in which it is displayed; and
- SCOs may be hosted by an LCMS which is in a different internet domain or sub-domain from the LMS (e.g. lms.yourdomain.edu and lcms.mydomain.edu).

The last two assumptions pose technical problems.

An effective solution to enable SCOs to pick up their look and feel from a SCORM “template” has been previously reported by Ip, Radford, & Canale (to be accepted 2003). Briefly, the approach is to incorporate support for course and session-level data within SCORM. Course templates are then implemented as style sheets, determining the course look and feel to be applied to individual SCOs.

The last assumption in the list above creates a problem because, due to security requirements, modern browsers will block any JavaScript calls or communications when the scripts involved originate from different domains. The SCORM communication mechanism is typically provided as JavaScript by the LMS and there is no problem while the content and scripts originate from the one LMS. The cross-domain scripting problem arises when a SCO originating from a LCMS seeks to establish communications with the LMS that has retrieved and delivered that content as part of a course. This is because the initiation of communication with the LMS is the responsibility of the SCO which originated from the LCMS rather than the LMS. Typically, a SCO will try to locate the communication script from the document hierarchy of the browser and communicate via the found communication script. However, in the case of the SCO coming from lcms.mydomain.edu rather than lms.yourdomain.edu, the browser will block the JavaScript calls.

An early solution to the "cross domain scripting" issue is to hard-code a domain attribute into the SCO. Modern browsers which respect a "domain" declaration in the HTML will allow JavaScript communication if both the originating and target HTML pages of the Javascript call have the same declared domain. However, experimentation has shown that success with this technique is inconsistent across browsers. Furthermore, hard-coding a domain into the SCO forfeits the original intent for re-use of SCOs across a wide range of courses in different domains. Therefore a solution that addresses the cross-domain scripting problem as well as enables re-use of content, managed in a systematic way would be of interest to both content developers and those interested in DRM.

The IMS Digital Repository Interoperability model Version 1.0 Public Draft Specification (DRI) was approved in August of 2002. The DRI takes into account the needs of existing digital repositories including different content formats, technologies and practices. It is intended for both systems utilizing established interoperability technologies such as Z39.50 and repositories to use the XQuery and SOAP recommendations provided in the specification. These recommendations describe a collection of resources for exposing meta-data to allow users to search, gather, store and deliver assets. A DRI repository may contain actual assets or meta-data describing the assets. An asset and its meta-data do not need to reside in the same repository (Looms & Christensen, 2002). The LCMS described above is equivalent to a DRI conformant repository of the actual assets.

While DRI is very rich in features related to harvesting digital assets and providing search services to locate assets using type 2 data (Ip, Currie, Morrison, & Mason, 1999), the specification also provides simple mechanisms for storing and retrieving assets in the DRI implementation. However, the specification does not address the "cross domain scripting" issue. For example, it is silent on how to implement a second DRI repository that does not reside in the same domain as the first. As discussed above, mimicking the source domain of the asset (by hard coding the domain attribute within the asset if it is a SCO) is not a solution when the one asset must be used (without copying) by various LMS hosted by different organizations.

In summary, the cross-domain scripting limitation within browsers poses a serious impediment where implementation of multiple repositories is concerned and where the LMS is in a different internet domain from one or more repositories from which the LMS is fetching content.

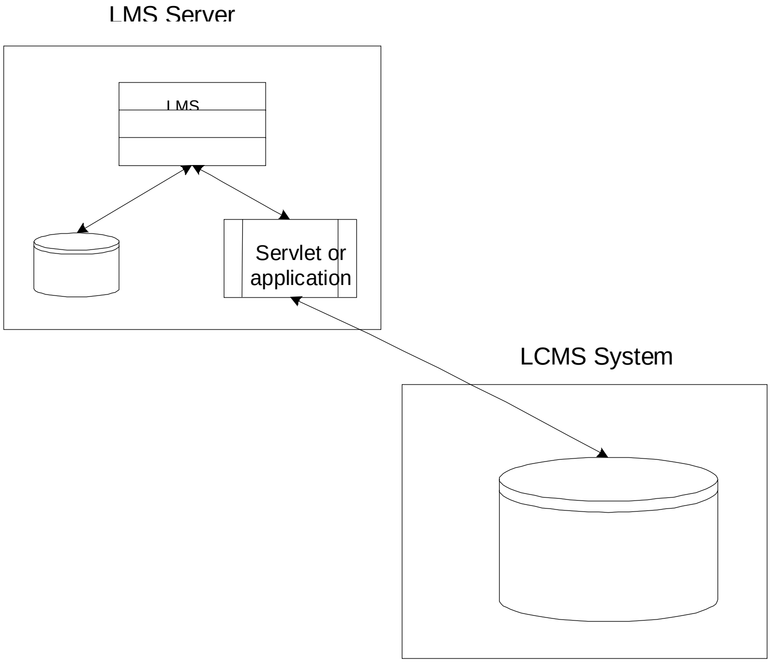
The SCO-fetcher Solution

Draft copy. Please check back later for final version. Please do not cite or quote without contacting the authors first (albert@DLS.au.com)

The SCO-fetcher is an implemented solution to the cross-domain scripting problem in the SCORM-LMS-LCMS context described above. It is anticipated that this model will create substantial efficiencies in the maintenance of course content because any change to the SCO may be deployed immediately to all courses using it and can be subject to version control on the LCMS.

The solution is designed to avoid the cross-domain scripting restriction inherent in Web browsers and can support the use of SCOs from multiple repositories in different domains while keeping the SCORM runtime environment intact. This has been demonstrated to be feasible during Plugfest 7 (Grobicki, 2002; Ip, Radford, & Grobicki, 2002).

Figure 1. Implementation overview. An application on the LMS server fetches SCOs from the LCMS and delivers it to the browser.



In a conventional SCORM implementation, a developer may not know the final location of a SCORM course at delivery time. The actual installation path is dependent on the specific LMS and its administration. Typically, when a SCORM course is installed into a LMS, a local path to the location of the installed course directory is required and is specified by the course administrator at installation time. The rest of the SCORM course is installed relative to this local path. Hence, the information in the manifest is a relative reference rather than an absolute reference. When a SCO is instantiated at the browser, the browser automatically retrieves the embedded assets by appending the current path for the SCO to the relative reference of each asset, in front of the `href` location specified in the package.

In a "pure" repository-based SCORM course (all the SCOs reside inside a LCMS), no change is needed in the manifest specification. Without modifying the current syntax of the IMS content packaging manifest file, the information to retrieve the LCMS-hosted SCO and asset can be stored in the `href` attribute of the `<file>` element. When a SCO which is stored in a LCMS is encountered by the LMS, a server side module (SCO-fetcher) fetches the SCO from the LCMS and delivers the SCO to the browser. The location of the SCO-fetcher is in the form of an URL. The SCO `href` becomes the parameter of this SCO-fetcher's parameter, appended to the end of the SCO-fetcher's URL. Effectively, it is the same as appending a local path to the beginning of the `href` of the SCO, albeit with a slight syntactic difference. In a repository-based course, the SCOs are not actually resident within the package. The semantic of the `<file>` tag needs a slight extension. The `href` attribute of this tag is the unique access ID of the SCO in the LCMS.

The result of this approach is that for the browser, the SCO appears to come from the LMS. There is no cross-domain scripting issue for the browser because the SCO and the SCORM API both appear to come from the LMS.

Supporting Multiple Repositories

When a course is composed of SCOs hosted by different LCMS, the SCO-fetcher will need mechanisms to identify which LCMS to contact and to pass the correct credential to the LCMS in order to retrieve the appropriate resources. Since the semantic of the `<file>` tag becomes a unique access ID of a SCO in a LCMS, we further extend the concept to provide a unique access ID across multiple LCMS. Part of the `href` attribute of the `<file>` tag can contain the additional parameter for indicating which LCMS hosts the SCO.

Supporting Mixed, Local and Repository SCOs

Once an institution has embarked on using this model, it is anticipated that there would be no reason to revert back to re-use mechanisms involving multiple copies of SCOs. However in transition from a multiple copy to a single copy system, there will be a period of time when both models need to exist side by side. Further, there should not be an issue with mixing SCOs originating from the native file system of an LMS with SCOs from a LCMS.

For a course that is wholly packaged within the LMS, the manifest file will contain relative referencing as described above. In a purely repository-based course, assets and SCOs are stored in a "flat" structure and may be retrieved using a unique ID. In a mixed local and repository-hosted course, if a SCO is hosted locally, its assets would necessarily be hosted locally to comply with the relative addressing in the manifest. If a SCO is hosted in a repository, its assets would also be hosted in a repository. (Note that a SCO's assets are not required to reside in the same repository as the SCO or necessarily with other assets of the SCO)

It follows that a mechanism to enable the LMS to distinguish between "local" and "repository" SCOs is needed. "Local" SCOs are those which are packaged within the course file and are to be installed into the native file system of the LMS. "Repository" SCOs are those which are to be fetched from a LCMS. This is best achieved in the manifest file by introducing an additional attribute to the `<resource>` tag. The exact syntactic representations of name and values allowed for this attribute is subject to further discussion but the authors suggest the name of the attribute may be `location` with the allowed values of `local`, `repository` and `remote`. By default, if the attribute were not defined for an item, it would be interpreted as `local`.

When a SCO with the "location" attribute having the value `repository` is encountered, the LMS will activate the SCO-fetcher to request the SCO from the LCMS, passing any necessary credential for digital rights management to the LCMS. When a SCO with the "location" attribute with a value of `local` is encountered, the LMS will append the local path in front of the `href` value to find the SCO in the local system for delivery.

Handling linked Assets

It has been observed that teachers very often use resources that are not originally created for educational use (Ip, Currie, & Morrison, 2001; Ip & Naidu, 2001), and support of this common practice may be regarded as a requirement for educational systems. For example, an art appreciation course may have a SCO that refers to a painting that is hosted on a museum web site and the museum's terms of use permit linking to the painting but expressly forbid storing the image elsewhere. In this situation, an absolute URL pointing to the painting from within the SCO is appropriate. In cases such as these, the SCO-fetcher model will continue to work without modification because when the browser launches the repository-sourced SCO it would detect an absolute referencing and correctly retrieve the painting from the museum site.

Discussion

The SCO-fetcher and proposed extension to the content packaging specification provide an effective solution to the problem of re-use of SCOs from multiple content repositories without making new copies of the SCOs. This model has little overhead in terms of SCO development, management of courses and installation of courses. The model also enables the implementation of secured fetching of SCOs (via server side authentication techniques) and hence is compatible with enforced digital rights management. The proposed extension to the content packaging specification is technically minor and is one of several developments, that when taken together, pose wider implications for the potential use of SCORM within higher education and training.

Previous work on SCORM by the authors and their co-authors has addressed the specific issues of supporting collaborative learning and enabling style-sheet support to overcome the presentation mosaic effect of re-used learning objects (Ip & Canale 2003, Ip, Radford & Canale 2003). Together with the current proposal to enable single copy re-use of SCOs sourced from digital repositories, two major impediments to wider implementation of SCORM can now be overcome. These are:

1. The pedagogical extensibility of SCORM; and

2. The feasibility of large scale re-use of learning objects (SCOs)

Much has been made of SCORM's narrow pedagogical model (solo learner with computer) but support for structured collaborative learning can be established using SCORM run-time communications and extension of the data model (Ip & Canale, 2003). Furthermore, the extensions proposed do not attempt to model the pedagogy being delivered (collaborative or otherwise), being limited to course and session data for the purpose of establishing a cohort of learners accessing the services of a collaboration server or other pedagogical tool or online services. The re-usability of SCOs that call on such services is preserved, provided that access to the required services is established at course installation time.

Based on our previous work on collaborative learning and a perceived need for future pedagogical extensibility, the current proposed modification to the content packaging specification includes the value "`remote`" as well as "`local`" and "`repository`" (as allowed values for the proposed attribute "`location`" in the `resource` tag). The intended use of "`remote`" is to establish bidirectional communication with specialist pedagogical services, for example in the support of structured collaborative learning. By comparison, the value "`repository`" results in unidirectional data transfer from the LCMS via the SCO-fetcher.

The support of SCO repositories and specialist pedagogical services are further intertwined in that both give rise to the cross-domain scripting problem described previously. Continuing with collaborative learning as an example, it is necessary to pass data (typically store and forward) among learners under a controlled context – and this is the main function of the "collaboration server". A distinction needs to be drawn between a SCO passing data back to the LMS (via the normal SCORM API) and a SCO passing data back to the collaboration server which may reside in a different domain. In general, the "request" data to the collaboration server poses no issue because this may be initiated by the SCO directly by calling the collaboration server. However, if "response" data comes back directly from the collaboration server the browser will block any further SCORM communication action on it as a violation of the cross-domain scripting rule. Preferably, both the "response" and "request" data would be sent via the SCO-fetcher to and from the SCO via an extended SCORM API.

Re-usability of content in the form of SCOs is a fundamental design objective of SCORM. Nonetheless the practical problems remaining are considerable. By enabling single copy re-use of SCOs managed by a LCMS as described in this paper, the key issue of SCO maintenance becomes the responsibility of the LCMS – a task for which the LCMS is purpose-built. The SCORM-LMS-LCMS integration removes a major obstacle to re-use for courseware developers and content creators. Another impediment to re-use is the presentation mosaic effect that is experienced when content originally from a course with one "look and feel" is re-used in a second course with a different "look and feel" because the HTML encoding includes information about how the content is to be displayed. A SCORM-based solution that enables course-level style sheet within course templates to be applied to SCOs is reported in detail elsewhere (Ip, Radford & Canale, 2003). The SCORM extensions proposed for style-sheet support are totally consistent with the SCO-fetcher described in this paper and they are to a great extent in common with the proposed SCORM extensions to support collaborative learning activities. Taken together, the authors believe the proposed extensions are

- light-weight in terms of technical overhead and complexity;
- consistent with and supportive of the design objectives for SCORM;
- solve significant implementation issues; and
- enable SCORM to be open to new developments in technology-supported learning environments.

The work reported here has focused on identifying key implementation issues and designing generic solutions that are consistent with the underlying model (SCORM). Experience to date supports the view that, with appropriate extensions, SCORM is sufficiently robust in its design to have the potential to become the major open standard for content interoperability across all educational and training sectors. Future work will continue to explore the potential for SCORM in higher education.

Reference

AICC. AICC CMI001 Guidelines for Interoperability . http://www.aicc.org/.

EML. (2001). Educational Modelling Language Project Website . http://eml.ou.nl/introduction/.

Grobicki, T. (2002). *True Re-use of SCO (video of presentation).* Paper presented at the Plugfest 7, http://147.160.150.139:8080/ramgen/video/plugfest/plugfest7/P7D1205T1130.rm?mode=compact.

IEEE. (2001). IEEE LTSC Website . http://ltsc.ieee.org.

IMS. (2001). IMS Global Learning Consortium, Inc. Website . http://www.imsproject.org.

Ip, A. & Canale, R. (2003). Supporting Collaborative Learning Activities with SCORM. *Proceedings of the EDUCAUSE IN AUSTRALASIA 2003 Conference.* 6-9 May 2003. Adelaide, Australia. Causal Productions. pp 669-678.

Ip, A., Currie, M., & Morrison, I. (2001, 22nd - 15th April 2001). *Resource Synergy: Improving the discovery of quality resources to support and information-dense learning environment.* Paper presented at the AusWeb01, Novotel Opal Cove Resort, Coffs Harbour.

Ip, A., Currie, M., Morrison, I., & Mason, J. (1999). *Metasearching or Megasearching: Toward a Data Model for Distributed Resource Discovery.* Paper presented at the Proceedings of the Fifth Hong Kong Web Symposium, Hong Kong.

Ip, A., & Naidu, S. (2001). Re-use of Web-Based Resources in Technology-Enhanced Student-Centered Learning Environments. *Campus Wide Information Systems, 18*(4), 153-159.

Ip, A., Radford, A. & Canale, R. (2003). Overcoming The Presentation Mosaic Effect Of Multi-Use Sharable Content Objects. *Submitted: ASCILITE 2003, 7th - 10th December 2003 Adelaide, South Australia.*

Ip, A., Radford, A., & Grobicki , T. (2002). *True Re-use of SCO (powerpoint).* Paper presented at the Plugfest 7, http://www.adlnet.org/adldocs/Presentations/P7D120502T1130.ppt.

Looms, T., & Christensen, C. (2002). *Advanced Distributed Learning Emerging and Enabling Technologies for the Design of Learning Object Repositories Report* : Advanced Distributed Learning Initiative.