

Supporting Collaborative Learning Activities with SCORM

Albert Ip, Ric Canale*

Digital Learning Systems P/L
Balwyn North Victoria 3104 Australia

* University of Melbourne

Victoria 3010 AUSTRALIA

E-mail: albert@dls.au.com, enricodc@unimelb.edu.au

This paper investigates the data elements required to fully and flexibly support a broad range of collaborative learning activities and proposes extensions to the SCORM data model to enable support for collaborative learning designs within SCORM. The analysis is partly based on a range of well established collaborative learning designs that cannot be implemented within the SCORM 1.2 specification. Educational technology standards should ideally offer teachers and students the greatest possible pedagogical flexibility, and encourage the full exploration of new pedagogies as well as offering other benefits through standardisation. The proposed extensions would enable support for collaborative activities delivered from within Sharable Content Objects. The proposal involves the addition of 14 course related data elements and 11 session related data elements to the SCORM data model.

1 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 LSTC effort). The SCORM has been developed by the Advanced Distributed Learning Network (ADL, <http://www.adlnet.org/>), an initiative of the Department of Defence, USA. Its development is informed by the IMS specifications, AICC specifications and the IEEE standards setting process. The major vendors of LMS (Learning Management Systems) 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 many different LMS.

ADL's SCORM offers the potential for widespread interoperability of content, and reusability of Sharable Content Objects (SCOs). Course providers with SCORM compliant content are able to deliver courses on any of the growing number of LMS that support the SCORM. With widespread adoption of the SCORM by content developers comes the possibility for creation of vast repositories of easily accessible content. While these intended outcomes offer enormous benefits to the education sector and the communities served by it, other issues of pedagogical design and flexibility need to be addressed as these issues impact substantially on educational outcomes and effectiveness.

The preparation of SCORM compliant content is assisted by a number of readily available tools if the content is in expository form and in the form of solo learning activities such as quiz, multiple-choice questions or rule-based simulation activities. It is well understood that the pedagogical model assumed by SCORM is primarily that of a single independent learner (Rehak, Oct 02, 2002). However, collaborative learning designs are becoming increasingly widespread within higher education and distance education. If SCORM is to become widely accepted as an e-learning

standard then it will need to cater for the widest range of pedagogical designs possible and at least support the most commonly used and most highly valued pedagogies deployed online. Collaborative learning activities represent a group of pedagogical designs not supported by the current version of SCORM (1.2). One approach to incorporate collaboration is to adopt Educational Markup Language (EML) with SCORM-based material sequenced within the collaborative activity (Dalziel, Oct 28, 2002). Another approach current practiced by some SCORM-compliant LMS (e.g. WebMentor®) is to enable the use of generic collaboration tools in parallel with the SCORM specification. Another solution, proposed here, is to enable collaborative learning activities from within a SCO by extension of the SCORM data model to enable SCOs to use collaboration services.

2 Overview of SCORM characteristics

A central design characteristic of the SCORM specification is the division of responsibilities between the LMS and the SCORM compliant content. It defines a data model to be implemented on the LMS to support various forms of data storage. It also defines a standard communication mechanism between the LMS and the content, which makes it feasible to re-use content with different LMS without re-work. Within SCORM, the SCO is the smallest grain size of re-usable learning object which can communicate with LMS to perform data exchange or tracking. A SCO is typically, though not necessarily a single Web page. The elements in the current data model, based on the AICC specification, are restricted to each individual SCO and each individual learner. Two different learners cannot share any of these data elements to pass information between themselves.

3 Collaborative Learning Activities

For this discussion the following definitions apply:

Table 1. Definitions of terms used.

<i>Term</i>	<i>Definition</i>
Activity	Actions that a learner is required to perform.
Learning activity	An activity designed to lead to a learning outcome.
Solo learning activity	A learning activity that can be performed by a single learner in front of the computer, e.g. multiple-choice questions and rule-based simulations.
Collaborative Learning Activity	A learning activity that involves more than one learner, where the learners are communicating with each other either as peer to peer or within assumed roles. We do not distinguish between whether the collaborative learning activity will occur synchronously or asynchronously. Our emphasis is on the fact that several learners are engaged in a learning activity or the activity requires the participation of more than one learner.
Collaboration tools	These are tools that enable learners to share information, discuss ideas and communicate for the purpose of collaboration. Some typical tools are asynchronous conference and/or chat tools.
Course	A unit of study, usually contributing towards a qualification.
Cohort	A group of students, all enrolled in the same course at the same time.

The distinction between a tool and an activity is an important one. Asynchronous conferencing is one of the commonly used collaboration tools (Hiltz, 1984; Ip, 1989). Collaborative learning activities require additional design (from instructional designers) or may use a combination of tools to promote the achievement of learning objectives. Some collaborative learning activities need further customisations by subject matter experts to provide an appropriate context for the collaborative learning activity while customization of the tool itself may be necessary to fully support the collaborative learning activity.

Two simple examples are used to illustrate typical uses of collaborative learning activities. The first example is the annotation tool, ANN. This tool enables students to annotate a body of text on an HTML page with their own notes and have the body of text together with their annotations made available to others for further annotation, discussion, feedback and review. The ANN tool is available for review at <http://www2.meu.unimelb.edu.au/oxygen/ann>

The second example collaborative activity is online debate. In this example, learners are divided into an even number of subgroups. The odd numbered subgroups mounting arguments for one side of a debate while the even numbered subgroups argue for the other side.

A central consideration to this proposal is whether a SCO based on a collaborative learning activity can still be a re-usable object. Provided that the SCO is designed to use the collaborative tool through a standard ("SCORM") data interface then it can be as re-usable as any content-only SCO. The context, instructions to learners, starting conditions or scenario of the collaborative activity are analogous to content in a content-only SCO. A SCO containing a collaborative learning activity is re-usable as long as the required collaborative tool is available and correctly configured for each instantiation of the collaborative SCO. The technical difficulty is to support the exchange of data **among** learners - after all, collaboration requires participants to share and sharing of data is fundamental. The concept of "active" content has not been extended to supporting the exchange of data among learners - not to mention the support of managing the data exchange rules.

From the perspective of pedagogy, collaborative SCOs may be more re-usable than content-only SCOs because of the educational value attached to the personalized nature of the collaborative activities. Each instructor and each cohort of learners using a particular collaborative SCO will create a unique collaborative learning experience with a strong sense of ownership of the experience and the learning outcomes. Although untested, it may be that the instructor's opportunity to value-add through moderation of the collaborative activity will make re-use of this type of learning object more acceptable to instructors than re-use of static content and may even become an attractive proposition to many instructors.

4 Technical Characteristics of Collaborative Learning Activities

Quasi-asynchronous activity

While solo learning activities may be performed by learners anytime, collaborative learning activities typically require a group of learners to proceed at an agreed pace. Even collaborative learning activities based on asynchronous communication will require a group of learners to work together in stages, completing some tasks at every stage at roughly the same time. For the learning activity to be collaborative and meaningful, the learners need to engage in each stage of the activity within an agreed timeframe. We call this quasi-asynchronous activity.

Role Assignment & Management

Since collaborative learning activities typically require "knowing" who the users are in order to forward appropriate information, some kind of authentication is required. Also, since learners in collaborative learning activities will interact in different roles (in our online debate example, either in the odd or even subgroup), the moderator needs an administrative interface to manage the role assignment and time based controls.

Activity support, Data sharing and Storage

Collaborative learning activities require services from online collaboration tools that enable management of learner roles, storage of interaction data (such as assignments and discussion postings submitted by a learner) and forwarding of selected data to other roles or learners. Due to the vast range of data that may be encountered and because the proposed services provide specialized support for a particular class of pedagogy (collaborative learning), it is unlikely

that this responsibility would lie with the LMS. We suggest the term “collaboration server” as a catch-all term to represent the services required to support collaborative learning activities.

5 Development and Activation Model for Collaborative Learning Activities within a SCORM framework

The following is an over-simplified workflow of courseware development in a SCORM environment without collaborative learning activities:

1. Subject Matter Experts create SCOs including SCOs which may include solo learning activities
2. Instructional Designers assemble SCOs into a course
 - Select relevant SCOs
 - Define the sequencing of SCOs
 - Produce a course package (including the imsmanifest.xml file which defines the structure of the course)
3. Course Administrator installs the course into a SCORM-compliant LMS
 - A copy of the course is copied to the LMS
 - The LMS reads in the course structure (as described in imsmanifest.xml file)
 - Appropriate access rights to the course are established and managed for learners and instructors
4. Learners begin interacting with course content asynchronously.

However, when some of the SCOs contain collaborative learning activities, the development and activation model will need to change and become:

- 1) Subject Matter Experts create SCOs including those SCOs which may include both solo and collaborative learning activities
 - a) The content for the collaborative learning activities may need to be supplied to the developmental collaboration server. This is similar to writing multiple-choice items (in a solo learning activity). A tool may be used to generate the software to support the interaction in the learning activity (analogous to coursebuilder in Macromedia Dreamweaver® for multiple-choice style questioning). Similarly for collaborative learning activities, the instructional designer and subject matter expert need to select the appropriate collaborative learning activities and supply the content. The software to support the interaction is "encapsulated" in the catch-all term "collaboration server". The content of collaborative learning activities may not reside exclusively within the SCO. It is more likely that part of the content actually resides within the collaboration server or may even be generated by learners and instructors during execution of the learning activity.
 - 2) Instructional Designers assemble SCOs into a course
 - a) Select relevant SCOs
 - b) Identify collaborative learning activities [for assisting course administrator to transfer the content of collaborative learning activities into the production collaboration server]
 - c) Define the sequencing of SCOs
 - d) Produce the course package in the form of imsmanifest.xml file
 - 3) Course Administrator installs the course into LMS
 - a) A copy of the course is copied to the LMS
 - b) Transfer (move or copy) the content data from the developmental collaboration server to the production collaboration server
 - c) The LMS reads in the course structure (as described in imsmanifest.xml file)
 - d) Activate and assign learners to the collaboration server
 - e) Register and assign course instructor(s) to the collaboration server
-

- 4) Learners begin interacting with most of the course content asynchronously.
- 5) Instructors group learners into study groups before commencement of collaborative learning activities
- 6) Instructors assist learners in collaborative learning activities

The current content packaging specification cannot support the use of tools such as a collaboration server especially if we include situations when the development machine is not the same as the deployment machine. For a solo learning activity, the aggregation model enables the content to be included in the package and installed into the production machine (or being traded). There is no provision in the current content packaging to specify an alternate server AND mechanism for the development server to transfer the content to the deployment server when such content cannot be packaged as in the case of collaborative learning activities.

6 Extending the SCORM Data Model to Support Collaboration tools and Collaborative Learning Activities

The SCORM (1.2) specification provides:

- a standard mechanism for a SCO to communicate with the LMS, and
- the cmi data model.

The current data elements are available to an individual SCO via standard API calls in the form of

```
LMSGetValue(dataElementName)
```

and

```
LMSSetValue(dataElementName, value)
```

The current data communication initialisation API call is `LMSInitialize()` which serves two purposes:

- 1) initialisation of the connection between the calling SCO and the LMS,
- 2) notification to the LMS that a SCO has been displayed to the learner.

In order to access the cmi data elements, a SCO needs to be initialised via the `LMSInitialize()` call. While the specification has not implied the `LMSInitialize()` indicates the launching of a SCO, a number of LMS vendors have taken this assumption which is valid given that only a SCO will be able to issue such a call at initialization and as specified in SCORM before v1.3, there is only one active SCO at any time. In some situations however, it may be necessary to access some data elements without indicating that a SCO is being initialised, e.g. for use by a sharable content asset.

This proposed extension consists of two parts:

- 1) broaden the communication between client and LMS.
 - a) introduce `LMSPreinitialize()` which enables the client (may be a course template or an asset) to establish communication with LMS without registering any SCO as being used
 - b) extend `LMSGetValue(dataElementName)` to include getting values in the extended data model after either `LMSPreinitialize()` or `LMSInitialize()`
- 2) extend the data model to support specification of the collaboration services and instantiation data for the collaborative learning activities:
 - a) some of the cmi data elements to be made available after `LMSPreinitialize()`
 - b) an extended data model using a new namespace.

In this model, collaboration tools and collaborative learning activities are supported by the following data elements:

```
[namespace].course.support.n.id,
[namespace].course.support.n.format and
[namespace].course.support.n.value
```

The namespace to hold the new elements is subject to community discussion. We shall leave it open and just indicate it as [namespace] in this paper.

Collaborative learning activities involve passing of data among learners to achieve a common learning objective. However, this is not a proposal to include all the data elements necessary to perform the actual collaborative activities within the SCORM data model. The actual data and rules related to the exchange of data to enact collaborative learning activities are much more complicated than a simple extension to the current data model. It is anticipated that purpose-built collaboration services will provide the necessary support for collaborative learning activities. We believe that the LMS should not include services for collaborative learning activities. We can't see a common data model that can be developed to support *all* collaborative learning activities. The purpose of the data model proposed in this paper is to store the information needed to initialize collaborative learning activities. It is also designed to support many student cohorts studying the same course simultaneously because this is a known requirement of many institutions. Each instantiation of a cohort within a course is referred to as a session and collaborative learning activities are typically based on groupings of learners from within the cohort of learners.

Table 2 shows the proposed data elements.

Table 2. Proposed data elements.

Element Name	Description
Course level elements	
<i>General</i>	
[namespace].course.id	the ID assigned to the course by the LMS during installation
[namespace].course.administrator.name	the name of the administrator who installed the course into the LMS
[namespace].course.administrator.email	the email address of the administrator who installed the course into the LMS
<i>Course Level Data</i>	
[namespace].course.instructor._count	number of instructor elements
[namespace].course.instructor.n.id	ID of instructor
[namespace].course.instructor.n.name	name of the nth instructor
[namespace].course.instructor.n.email	email address of the nth instructor
[namespace].course.message._count	number of messages for the course
[namespace].course.message.n.date	the date and time of the nth message
[namespace].course.message.n.text	the nth message in plain HTML format
[namespace].course.message.n.url	the url of the nth message
[namespace].course.support.n.id	the ID of the nth data element The following are pre-defined: 0 for conference Users are free to define other elements to meet local needs. A shorthand mapping is also defined: [namespace].course.support.conference[...] is mapped to [namespace].course.support.0[...]
[namespace].course.support.n.format	specifies the format of the support elements: txt = the elements in text (limited to 1024 bytes) url = the URL of the support
[namespace].course.support.n.value	the value of the nth element

Session level elements: When a course is run multiple times, each run is referred to as a session of the course. When a session-level data element is not defined, the element will take the value of the element with the same name in the course level elements.	
<i>Session Level Data</i>	
[namespace].session.id	the ID assigned to the session by the LMS during installation
[namespace].session.instructor._count	number of instructor elements
[namespace].session.instructor.n.id	ID of instructor
[namespace].session.instructor.n.name	name of the nth instructor
[namespace].session.instructor.n.email	email address of the nth instructor
[namespace].session.message._count	number of messages for the session
[namespace].session.message.n.date	the date and time of the nth message
[namespace].session.message.n.text	the nth message in plain HTML format
[namespace].session.message.n.url	the url of the nth message
[namespace].session.support.n.[...]	See [namespace].course.support.n.[...] elements. If [namespace].session.support.n.[...] is not defined, the value of [namespace].course.support.n.[...] will be returned
[namespace].session.start.date	the date of the nominal start of the session
[namespace].session.stop.date	the date of the nominal stop of the session

As the values of some of the elements in this data model are locally dependent on the organisation hosting the SCORM course, such information should be entered during the installation of the course or session and hence may be LMS specific. The impact of this on re-usability of the collaborative SCO is minor if it can be assumed that a system administrator installs and configures courses. The impact on the system administrator is non-trivial but routine. As shown in the suggested workflow for course activation (above), a trial configuration and activation on a developmental server is recommended prior to activation of the course. An evaluative trial of this kind would be standard practice for organisations engaged in significant online learning.

7 Typical Use Scenarios of the Proposed Data Model

Courses with a human instructor

In courses with an assigned human instructor (tutor or lecturer), students may contact the instructor to seek help with any course related issue, not just content (SCO) issues. It is also common for instructors to post a welcome message and special notices from time to time. The course level or session level elements `[namespace].session.instructor.n.name` & `[namespace].session.instructor.n.email` are designed to support these practices. When this information is included in the extended data model, the instructor name and email address are obtainable in the same way other data elements in the current data model may be obtained with the exception that the instructor name and email address are not necessarily linked to the initialization of a SCO. Instead, a "pre" initialization call is needed simply to establish communication with the LMS to obtain the instructor name and email address.

Collaborative learning activities with a global audience

Annotation (ANN) is a tool that allows readers to annotate HTML pages. It is a good example of a collaborative learning activity not limited to collaboration within a cohort of learners. Typically, any learner accessing the SCO can share their annotations globally. The learners' annotations are stored in a database under the control of the annotation collaboration server (a prototype can be found at <http://www2.meu.unimelb.edu.au/oxygen/ann>). The annotations are separate from the annotated documents. Once an annotation has been entered into the ANN system, any other learners can read the annotation and participate in further discussion. This is like a mini forum hosted specifically for discussion of a particular paper.

To create an annotation point on a SCO, a link is required. This link can be obtained by supplying the following details to the project website:

- subject matter expert's name,
- email address and,
- optional information such as the SCO Title and URL.

An email, which contains the link (in HTML code) for insertion into the SCO, is returned automatically. This particular activity, while requiring the support of a collaboration service, does not involve using the proposed extensions to the SCORM data model.

This link will render a yellow post-it icon on the document dynamically. The clean post-it (Figure 1) indicates an annotation point which has not had any comment yet. The "dirty" post-it (Figure 2) indicates an annotation point for which there are annotations. When readers click on the post-it icon, a new window will open where they will see all the previous annotations as well as text boxes for entering new annotations (Figure 3).

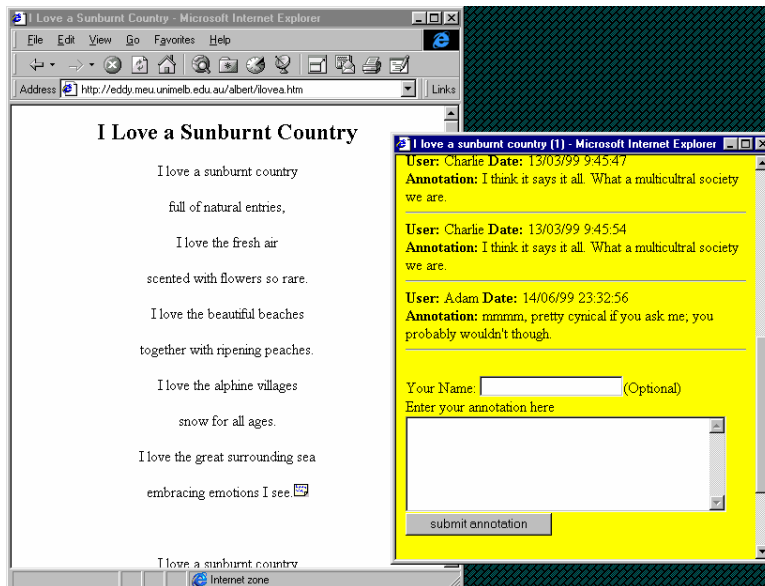
Figure1. Clean Post-it icon used by ANN



Figure 2. Dirty Post-it icon used by ANN



Figure 3. An annotation window opened on top of a base SCO.



Collaborative learning Activities limited to a cohort of learners

When a SCORM course is completely self-paced, there is no need to distinguish between different cohorts of learners. However, due to the quasi-asynchronous nature of collaborative learning activities, it is necessary to introduce the concept of sessions. Online debate is a good example of a collaborative learning activity which requires a defined number of participants. During the creation of a SCO containing an online debate, the subject matter expert (or instructional designer) would typically create a collaborative learning activity exercise by:

1. finding a collaboration service which supports the activity (e.g. <http://www.thisOnlineDebateService.com>)
2. Obtaining a useID (e.g. DebateID=12345). This useID is for linking the content to the specific collaborative learning activity.
3. supplying the content (e.g. the topic of the online debate)

To enable re-use of the SCO (e.g. among paired groups within the one course), each online debate in the course would require an instantiation of that particular use (specified by useID) from the collaboration service. The collaboration server would allocate an instantiation ID to each online debate being serviced. The student_ID together with the instantiation ID are sufficient to assign the learner to participate in the right online debate for their course. In other words, the collaboration service has been correctly configured to meet the course requirements.

In this example, we shall store the instantiation ID (say instance101) in one of the elements of the extended data model - ie.

```
[namespace].session.support.1.id = "instantiation"  
[namespace].session.support.1.format = "txt"  
[namespace].session.support.1.value = "instance101"
```

The following code segment illustrates how the extended data elements may be used.

```
//getting variables for use in building the call to a collaborative  
//learning activity service  
var studentID= API.LMSGetValue("cmi.core.student_id");  
var sessionID=API.LMSGetValue("[namespace].session.support.1.value");  
...  
//at the point of linkage to the collaborative learning activity  
<script>  
    document.write("<a  
href=http://www.thisOnlineDebateService.com?DebateID=12345&sessionID="+s  
essionID+"&studentID="+studentID+">Enter online debate</a>");  
</script>
```

The use of a session support value in the extended data model allows a cohort of students to engage in their group-based collaborative learning activities - not a global debate.

8 Conclusion

This study finds that the idea of a SCO (Sharable Content Object) can include a collaborative learning activity without significantly compromising re-usability, assuming support for its implementation from within SCORM. The idea of a collaborative SCO adheres to SCORM's basic design principles and opens up the potential of re-usability for a wide range of collaborative learning activities which are highly valued within the online education community. An extension to the SCORM data model has therefore been proposed to provide the necessary support within SCORM for collaborative SCOs. The proposal is designed to be fully compatible with the existing specification and is purely an extension of the existing specification. It is hoped that both the concept of the "collaborative SCO" and the proposed specification will encourage further interest in bringing collaborative learning activities within the realm of online educational content standards.

9 Acknowledgements

The authors would like to thank Daniel Rehak for his valuable comments and suggestions to this paper.

10 References

1. Dalziel, J. (Oct 28, 2002). Multi-learner environments . (online <http://www.cetis.ac.uk/content/20021002000737/20021028013942> last visit 30th October, 2002.)
 2. Hiltz, S. R. (1984). *Online Community*. New Jersey: Ablex Publishing Corporation.
 3. Ip, A. (1989). *A Study of the Potential of Electronic Bulletin Boards as Perceived by Teachers*. Unpublished M.Ed, The University of Hong Kong, Hong Kong.
 4. Rehak, D. (Oct 02, 2002). SCORM is not for everyone . (online <http://www.cetis.ac.uk/content/20021002000737> last visit 30th October, 2002.)
-