How to use IMS Learning Design and SCORM 2004 together

Colin Tattersall, Daniel Burgos, Hubert Vogten, Harrie Martens, Rob Koper
Educational Technology Expertise Centre, The Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
{cta,dbu,hvo,ham,rkp}@ou.nl

Abstract

Standardisation plays an increasingly important role in e-learning, requiring designers to make choices as to the route to be followed during the development of e-learning courses. IMS Learning Design is an e-learning specification which allows e-learning designers to describe Units of Learning – delimited pieces of education or training, such as courses, modules or lessons. SCORM 2004 is the latest version of Advanced Distributed Learning's reference model for e-learning, which describes a content model and run-time environment for Shareable Content Objects. IMS Learning Design and SCORM 2004 are often positioned as mutually exclusive alternatives. This article outlines the case for using the two together and examines approaches to achieving integration between Units of Learning and Shareable Content Objects.

1. Introduction

Two important pieces of the e-learning standardisation puzzle are IMS Learning Design (IMSLD) [1] and the Sharable Content Object Reference Model [2]. While the two have different natures – IMSLD is a single specification whereas SCORM 2004 is a reference model containing a number of specifications – their application areas and terminology overlap to a sufficient degree that confusion exists as to their relationship, and there is ongoing speculation in the e-learning community on when to use which one, and with which intended benefit [3, 4].

This article argues that despite some key differences, the two can be used together to give a useful e-learning combo. We analyse the relationship between the IMSLD and SCORM 2004, identifying why and where the two can be combined. This is followed by an examination of how different levels of integration can be achieved.

2. Analysis

IMSLD is a specification used to model Units of Learning (UoL) – "any delimited piece of education or training, such as a course, a module, a lesson, etc". A UoL goes beyond a collection of learning resources to represent the whole learning process, including learning activities, assessments, services and support facilities provided by teachers, trainers and other staff members [5, 6]. UoLs are content packages containing materials which can be 'played' by software able to interpret the materials. Although a single specification, IMSLD is designed to orchestrate learning arrangements, and so is linked to several other e-learning specifications.

SCORM 2004 is the latest version of the Sharable Content Object Reference Model (SCORM), consisting of a Web-based learning Content Aggregation Model (CAM), Run-Time Environment (RTE) and Sequencing and Navigation behaviour for learning objects [2]. If educational material is created according to the SCORM 2004 model, a SCORM 2004 compliant RTE will be able to 'play' the material and the expected run-time behaviour will result. The SCORM 2004 requirements on the content cover not only its structure and packaging but also requirements on implementing run-time behaviour so that communication between a running Shareable Content Object (SCO) and an associated Learning Management System (LMS) is facilitated.

These two brief descriptions illustrate similarities between IMSLD and SCORM 2004 – both can be used to guide the development of educational materials, both use a combination of specifications to achieve their goals and both lead to content packages which can be read into players and used to support learning.

However, these similarities mask fundamentally different views on learning. Several authors have pointed out that SCORM is currently centred on a single learner model [7-9] while IMSLD allows

learning flows involving groups of learners to be represented. Furthermore, IMSLD is able to model learning experiences involving multiple roles (eg tutor, learner, coach) and, drawing on the constructivist movement, places learning activities rather than learning content at the heart of its model.

UoLs typically incorporate content to help learners and/or staff carry out their activities, and this content can be in a variety of formats, including XML, {X)HTML, RTF, PDF etc. IMSLD does not prescribe a model to which content must adhere but focuses instead on specifying the learning process in terms of which roles perform which activities, when, and supported by which facilities. In this way, IMSLD can be seen as an orchestration layer into which SCOs can be slotted at appropriate points in a learning process. IMSLD was designed with such an integration in mind, and provides a placeholder in the 'environment' associated with an activity. Figure 1 shows this arrangement.

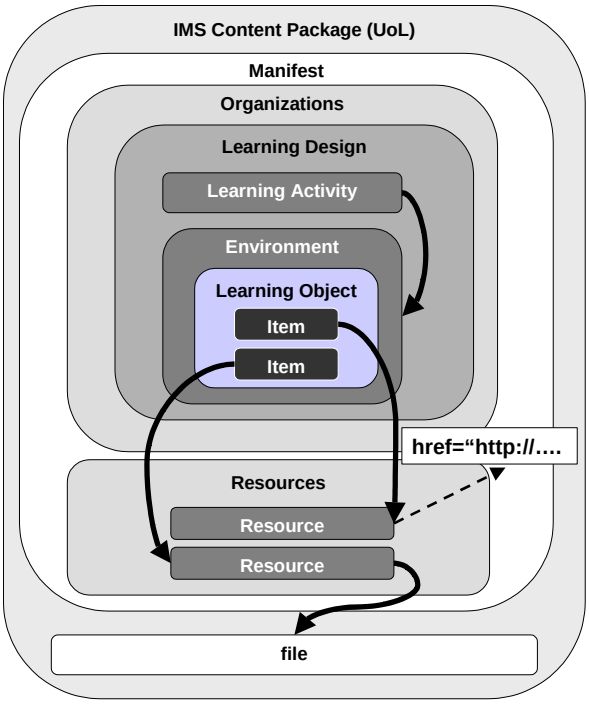

Figure 1: Key concepts from IMS Learning Design

A UoL is an IMS Content Package including a manifest, a learning design, resources and physical files. For clarity, figure 1 excludes some of the concepts inherent in a UoL (eg plays and acts), highlighting instead the concept of a learning activity and its associated environment – a structured collection of Learning Objects, Services, and (sub-) environments to be used when a role carries out the activity. The Learning Activity shown in figure 1 is associated with an environment containing a single Learning Object, which contains two items. Each item, in turn, points to a resource in the resources section of the content package. The figure illustrates that resources may reference files included in the package or content outside the package available through a URL.

Integrating SCOs into UoLs is a question of placing the SCORM 2004 content in the context of one or more learning objects in one or more environments in an IMSLD UoL. By integrating SCOs into UoLs in this way, a context for learning objects is established, specifying how learning designers intend learning objects to be used by learners. The next section explores different ways of approaching integration.

3. Method

Minimal integration involves simply referencing a SCORM-based LMS running a SCO from within a Unit of Learning, illustrated in figure 2.

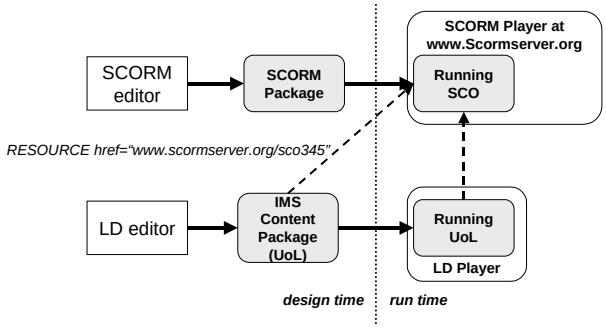

Figure 2: Minimally integrating a SCO in a UoL

During design time, a SCORM package containing one or more SCOs is authored using a SCORM editor and delivered into an environment which includes a SCORM 2004 aware run-time configuration. The run-time configuration is able to launch a SCO and handle the required SCO-LMS communication.

If this run-time configuration is addressable through a URL with parameters, it can be used from within a UoL to reference the SCO. Figure 3 shows an example of this minimal level of integration using the player which accompanies the CopperCore IMSLD engine [10]. The top-left-hand panel displays the title of a learning activity ("Read the resources contained in the environment"). The bottom-left-hand panel shows the environment associated with this learning activity (entitled "Link to running SCO"), which contains a single learning object "Follow me

to Moodle". Clicking on the learning object opens a new window and launches an example SCO running in the SCORM player available in Moodle [11].

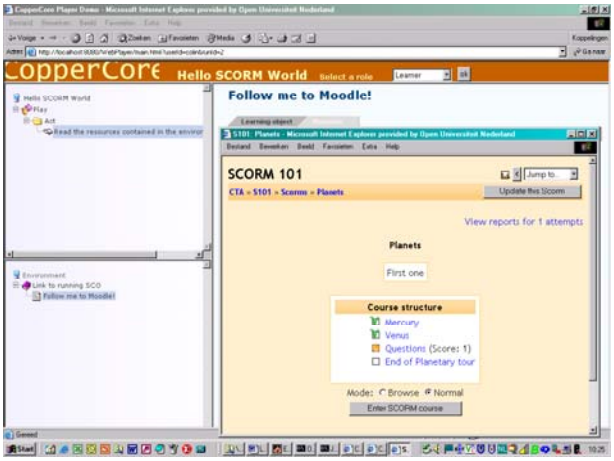

Figure 3: The CopperCore IMS LD player referencing a SCO running in the Moodle SCORM player

Minimal integration has simplicity as its main advantage – it can be accomplished straightforwardly with existing tools and content. Although the UoL shown is a toy example, we can imagine learning situations in which individuals in a cohort of learners are invited to share and discuss their preconceptions on planetary motion before being provided with content individually (a SCO) on the planets for self-study. Once digested, a second collaborative phase guided by a teacher could be used to encourage the learners to reflect on changes in their understanding. All this is possible using IMSLD concepts and software in conjunction with SCORM 2004 content and software.

Minimal integration does however suffer from some drawbacks. Learning designers may not know the final URL location of a SCORM course as they design a UoL. In order to avoid this issue, a variation could be used exploiting the fact that UoLs and SCOs are both packaged using IMS Content Packaging and that nesting of packages is allowed. In such *packaged integration*, the UoL and SCO are packaged together and instead of the UoL referencing content external to the package, the embedded SCORM package is used as the resource associated with the Learning Object. The package must be "disaggregated" into its constituent packages (one containing the SCO, the other containing the UoL) by some UoL Pre-processing mechanism. To avoid the need for this mechanism to examine the contents of the package in order to know how to process it, the IMSCP resource type should be set appropriately. Once recognised and disaggregated, the SCORM package is then delivered to a SCORM run-time configuration, and the UoL is delivered to the learning design player, with a link from the UoL to the running SCO being generated on the fly.

Perhaps the bigger drawback of these levels of integration is the lack of communication between the running UoL and the running SCO, limiting the benefits to be gained from the combination of IMS Learning Design and SCORM 2004. If information on the learner's status and progress in a SCO were to be available to a running UoL, it could be used to influence the learning flow following completion of the SCO. In this way, the learning activities presented to a learner could vary depending on learner-SCO interaction, such as including additional remedial activities, or skipping parts of the learning flow which appear on the basis of tests to be within the learner's competency level. Moreover, the SCO run-time information could be used during the execution of a SCO to trigger events in the UoL. Such *run-time integration* opens a number of possibilities for collaboration within and around SCOs; we can imagine a chat facility being opened (a Learning Service in IMSLD) between a tutor and a learner if the time spent on a particular SCO exceeds a certain threshold.

The SCORM 2004 Run-Time Environment Data Model specifies a set of data model elements which can be used to monitor SCO information, and an Application Programming Interface through which standardised communication between a SCO and an LMS can occur.

How might this information be used in a UoL? A large part of IMS LD's flexibility in orchestrating learning flows comes from its use of properties and conditions. Properties are used to record various types of information, which can be used in conditions to influence aspects of the learning process, including the ordering and visibility of learning activities and learning objects. The route to tighter integration involves learning designers defining properties and conditions which refer to the SCORM 2004 data model elements supported by data-enabled SCOs. The environment in which a UoL runs must be extended to include the capability to launch and communicate with a SCO and where necessary to map and synchronise SCORM 2004 data elements and IMSLD properties. Figure 4 shows the integrated situation.

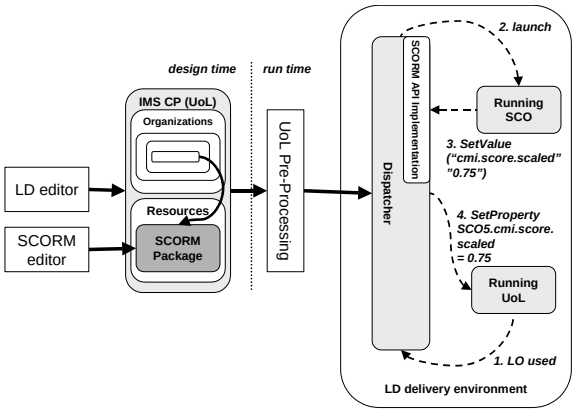

Figure 4: A deeper level of integration

Figure 4 introduces a new piece of functionality known as the Dispatcher. This is a central coordination and synchronisation mechanism responsible for ensuring that the right content is made available to learners and staff at the right time and that data is shared and mapped between underlying components appropriately. We can trace the sequence of events in the LD delivery environment to illustrate the functionality involved. When a Learning Object is used (step 1) a request is sent to the Dispatcher to Launch the relevant SCO (step 2). The Dispatcher acts here as an LMS, following the requirements for LMS-SCO initialisation and communication described in SCORM 2004. The learner interacts with the SCO, which at some stage in the interaction issues a call to set the value of cmi.score.scaled (step 3). The Dispatcher implements this functionality and also sets the corresponding IMSLD property (step 4). This property is used in an IMSLD condition, which is triggered by this change in value, causing a new activity to be shown in the running UoL.

4. Summary

This article has examined why and how IMS Learning Design and SCORM 2004 should be seen as complementary rather than alternative routes to creating e-learning courses. It suggests using IMS Learning Design to describe the orchestration of learning processes into which SCORM 2004 content is slotted at appropriate points. Learning designers benefit from the combination by being able to tap into the existing body of content developed according to the SCORM model while at the same time being liberated from the single learner, single role model through IMSLD's broader pedagogical scope.

We are currently implementing the approach described in this article as a generic solution to integrating IMSLD with other tools and content including SCORM, IMSQTI, ePortfolios, simulations and games [12].

References

[1] IMSLD, "IMS Learning Design Specification," vol. 2004, Version 1 ed. Boston: The IMS Global Learning Consortium, 2003.

[2] ADL, "Sharable Content Object Reference Model (SCORM) 2004," vol. 2004, 2004 ed: The Advanced Distributed Learning (ADL) Initiative, 2004.

[3] R. Ellaway, D. Dewhurst, and S. Cromar, "Challenging the Mortality of Computer Assisted Learning Materials in the Life Sciences: The RECAL Project," *Bioscience Education Electronic journal (BEE-j)*, vol. 3, 2004.

[4] M. J. Stiles, "How does content standardisation impact on staff support in the use of VLEs?," presented at LTSN Supporting Sustainable e-Learning event, University of York, 2003.

[5] R. Koper and B. Olivier, "Representing the Learning Design of Units of Learning," *Educational Technology & Society*, vol. 7, pp. 97-111, 2004.

[6] B. Olivier and C. Tattersall, "The Learning Design Specification," in *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, R. Koper and C. Tattersall, Eds.: Springer Verlag, 2005, pp. 21-40.

[7] P. Karampiperis and D. Sampson, "Designing Learning Services: From Content-based to Activity-based Learning Systems," presented at Proc. of the fourteenth International World Wide Web Conference (WWW2005), Chiba, Japan, 2005.

[8] F. Neumann and R. Geys, "SCORM and the Learning Grid," presented at 7th HLRS MetaComputing and Grid Workshop, Stuttgart, Germany, 2004.

[9] R. Wirski, G. Brownfield, and R. Oliver, "Exploring SCORM and the national flexible learning toolboxes," presented at Proceedings of the 21st ASCILITE Conference, Perth, 2004.

[10] H. Vogten, C. Tattersall, R. Koper, P. Van Rosmalen, F. Brouns, J. Van Bruggen, P. Sloep, and H. Martens, "Designing a learning design engine as a collection of finite state machines," *International Journal on E-Learning*, forthcoming.

[11] M. Dougiamas, "Moodle," vol. 2004, 2004.

[12] A. Little, "JISC ELF Toolkit and Demonstrator Projects," 2005.