What Is...

IMS Simple Sequencing?

by Warwick Bailey

In Brief

What is IMS Simple Sequencing?

IMS Simple Sequencing is a specification used to describe navigation paths through a collection of learning activities.

IMS Simple Sequencing is simple only because it restricts itself to the case of a single user in the role of learner and to a limited number of ways to control sequencing. More complex cases of multiple users and roles are not addressed by the specification.

A key benefit of IMS Simple Sequencing is that it allows an instructional designer to describe many different paths through one set of learning activities.

The ADL Organization demonstrated an early example of simple sequencing in 2002. An instructional designer took a training tutorial consisting of a collection of 48 activities – 23 lessons and 25 independent test questions. Simple sequencing was used to create six different paths through the entire set of 48 activities. One path took the learner through all the lessons then all the questions. An alternative path located sets of five questions next to their corresponding lessons. If the test pass conditions were not satisfied, the learner was routed back to the appropriate lesson.

IMS Simple Sequencing relies on the concept of learning activities, such as lessons and tests. These activities may then be associated with other activities into a hierarchy, resulting in an activity tree.

IMS Simple Sequencing also uses the concept of learning objectives. Each objective is associated with an activity. An objective includes a measurement similar to a test pass mark that determines if the objective has been passed or failed.

What is IMS Simple Sequencing for?

The main things that IMS Simple Sequencing allows you to do are:

• Author a learning experience so that a learning technology system can sequence discrete learning activities in a consistent way.

• Include a number of widely used sequencing behaviours in a learning path, such as making forward progress conditional on mastery of prior activities.

• Create multiple paths through a single set of learning activities.

• Create a sequence of activities so that a learner can leave and resume at their exit point at a later date. Activity status and progress towards objectives is saved between sessions.
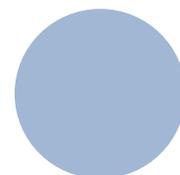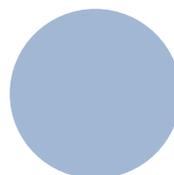
Technical Details

How IMS Simple Sequencing works

IMS Simple Sequencing works by a computer system reading instructions and determining which activity to present next to the learner.

The instructions are known as *sequencing information*. The process of determining the next activity is called *sequencing behaviour*.

Sequencing information is broken down into three categories: the *definition*, the *tracking* and the *activity state information*.

Definition information is written by an author adding instructions to an XML file. Currently the only XML file recognised for this purpose is the imsmanifest.xml file used in IMS Content Packaging. This file is used for describing collections of learning activities. The result is one or more activity trees integrated with a set of sequencing instructions.

Tracking information is collected when a learner works with activities: at *run time*. It is a record of the learner's attempts at an activity and the learner's progress towards an activity's objectives.

Activity state information is also captured at run time and records the status of an entire activity tree for a particular learner.

Sequencing behaviour is defined in terms of six processes: the *navigation*, *termination*, *rollup*, *selection*, *sequencing* and *delivery* processes. The combined behaviour, or *overall sequencing process*, operates as a repeating loop of the six processes working on the three information categories.

The loop typically starts when a learner clicks a hyperlink to move to the next activity. This event is mapped to the **navigation process**, which generates two further events. The first event triggers the **termination process** and ensures that information about the current activity is processed before the next activity is shown. The second event triggers the **sequencing process** in which a combination of sequencing controls, sequencing rules and limiting conditions identify the next activity to be delivered. This process then hands over control to a **delivery process** that checks that all the conditions required for identified activity still hold true. If all the conditions are satisfied the learner works with the new activity. When the learner is ready to move on, the whole loop is repeated.
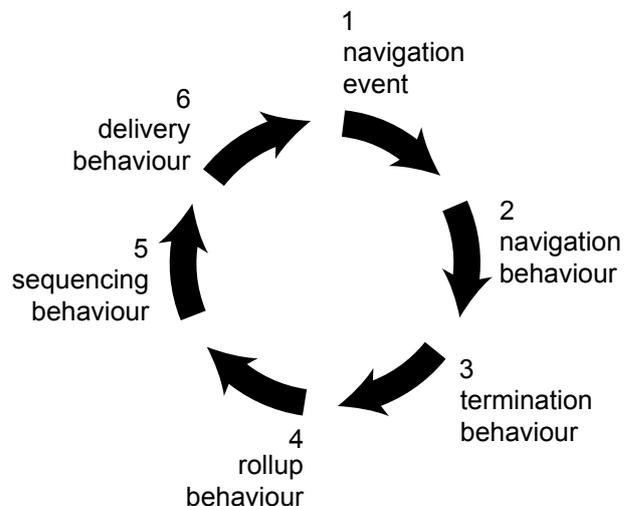
The **rollup process** is invoked when an activity terminates and has activities above it and next to it in the activity tree: it has *parent* and *sibling* activities. The status of the terminated activity is combined with the status of its siblings to determine the status of the parent. Rollup updates the tracking information records for both the terminated activity and its parent.

The **selection process** may be invoked when an activity has child activities. The procedure of choosing an ordered list of only some child activities from the total set of all children is called the selection process.

Requirements

In the example of simple sequencing demonstrated by the ADL Organization, it was suggested that three roles were required for creating a package of sequenced learning resources.

First, a content author would write a set of discrete lessons and tests in electronic form.

Second, an instructional designer would define one or more paths through the set of activities. Each path will have a different pedagogical rationale.

Third, a developer would add commands from the Sequencing Definition Information model as extensions to tags in an imsmanifest.xml file. This creates one or more activity trees integrated with a set of sequencing instructions that describe the paths defined by the instructional designer.

The point of distinguishing the three roles was to emphasize the importance of pedagogy. The order of lessons and tests is best defined by an education professional.

At the moment, common practice is for the developer to add XML extensions using a text editor or XML tool. It is anticipated that visual authoring tools for sequencing will become available to instructional designers. Such a tool has been demonstrated by Giunti Labs. Another tool is being developed as part of the Reload project.

Related specifications

IMS Simple Sequencing is currently related to one other specification, IMS Content Packaging. It is also incorporated into an application profile, the ADL SCORM 1.3.

IMS Content Packaging is related because the imsmanifest.xml file provides a ready structure for nesting activities into a hierarchy. The hierarchy, or *organization*, is composed of *item* elements, with each item element acting as a pointer to a real *resource*, such as web page that contains a lesson or test. IMS Content Packaging items are mapped to activities in IMS Simple Sequencing. The organization of nested items becomes the sequencing activity tree.

IMS Content Packaging allows for the inclusion of more than one organization hierarchy. This allows more than one activity tree for the same set of resources to be enclosed within a single IMS Package.

ADL SCORM 1.3 is related in that it integrates IMS Simple Sequencing into the SCORM Content Aggregation Model, the SCORM Run-Time Environment, and defines the application of IMS Simple Sequencing in the SCORM 1.3 Sequencing and Navigation Book.

One other IMS specification deals with the sequencing of learning activities: IMS Learning Design. Unlike IMS Simple Sequencing, IMS Learning Design supports a wide range of pedagogies and allows for multiple users and roles.

Implementations

Reload

Reload is a project funded under the JISC Exchange for Learning Programme (X4L). The project is planning an implementation of the IMS Simple Sequencing Definition model in its IMS Content Packaging tool.

Visit http://www.reload.ac.uk/

IBM Learnspace

The IBM Learnspace product includes a sequencing engine that is able to manipulate any content that conforms to the SCORM 1.3 application profile.

Visit http://www.lotus.com/products/learnspace.nsf

Giunti Labs

The Italian based e-learning company has demonstrated products that incorporate IMS Simple Sequencing both in content authoring tools and learning management systems.

Visit http://www.giuntilabs.com/

The Author

The author, Warwick Bailey, is writing an implementation of the IMS Simple Sequencing in Java. The sequencing engine is designed as a "plug in" for learning management systems.

Visit http://www.icodeon.com

Resources

People, products and services

The CETIS Educational Content Special Interest Group (EC-SIG) offers information, support and advice to people working with IMS SImple Sequencing and related specifications. Contact Sarah Currier at sarah.currier@strath.ac.uk, or go to:

http://www.cetis.ac.uk/educational-content/

Resources on the Internet

The IMS Simple Sequencing Specification is published on the IMS Global Learning Consortium website. The current version, Version 1.0 Final Specification, was released in March 2003. Visit:

http://www.imsglobal.org

The ADL SCORM Version 1.3 Application Profile Working Draft Version 1.0, which incorporates IMS Simple Sequencing, was released in March 2003. This is a public working draft for what is to evolve into the draft SCORM 1.3 document (due summer 2003). The Sample Run-time Environment 1.3 Beta 2 was released in April 2003. Visit:

http://www.adlnet.org

The Carnegie Mellon Learning Systems Architecture Lab publishes a "SCORM Best Practices Guide for Content Developers". The current version, corresponding to the Public Release of the IMS Simple Sequencing Specification and the Draft SCORM 1.3 Application Profile, was released in February 2003. Visit:

http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/

A further guide, "Simple Sequencing Services", exploring the technical implementation of the IMS Simple Sequencing was published in September 2002. Visit:

http://www.lsal.cmu.edu/lsal/expertise/papers/index.html

About this guide

This guide was produced by the CETIS Educational Content Special Interest Group and edited by Sarah Currier and Scott Wilson. For more information on CETIS, visit http://www.cetis.ac.uk/educational-content.

Warwick Bailey leads the CARET elearning Systems Group at Cambridge University. He is a board member of MIT's iLab Project and a developer with the Open Knowledge Initiative.

Warwick is also lead developer with a national eLearning project in Germany. He has extensive experience in curriculum innovation in technical education programmes in the UK and internationally. Visit http://www.icodeon.com

About CETIS

CETIS is the JISC's Centre For Educational Technology Interoperability Standards. For more information visit the CETIS website at http://www.cetis.ac.uk/

CETIS is funded by JISC, the Joint Information Systems Committee of the Universities' Funding Councils, and is managed by Bolton Institute, partnered by the University of Wales, Bangor.