SCORM 2004 Primer

A (Mostly) Painless Introduction to SCORM

Version 1.0

August 03, 2004

Written By:

Gord Mackenzie
Senior Business Strategist

Edited By:

Marianne Baeini
Content Specialist

Contents

Introduction

If you work in the e-Learning field, you've probably at least heard of SCORM. It's hard to avoid these days. For those wishing to know what SCORM actually is and how it affects the development of e-Learning content, there are over 700 pages of in-depth documentation available online from the ADL.

If you have the available free time and like to bury yourself in technical manuals, then please stop reading now and go to www.adlnet.org to obtain the full document set.

If you don't have the time or inclination to pore through 700 pages of dense technical details then this SCORM primer was written for you. This document attempts to illuminate the most critical aspects of the latest version of SCORM (SCORM 2004), and provide them to the reader in as painless a manner as possible.

Hopefully this document will be readable, friendly and informative. After reading this primer, you should be able to converse intelligently about SCORM, plan an e-Learning strategy in keeping with the main tenets of SCORM and grill vendors about their approach to SCORM and their true level of SCORM conformance.

The document is written in a question and answer format, so feel free to jump to what interests you and skip the stuff that doesn't (but you were going to do that anyway, weren't you?).

Let's get started…

Why Should I Care About SCORM?

The SCORM is important for a number of reasons, and I'll go into more detail about those reasons later in the document.

However, the main reason why the SCORM is important is because it makes Web-based learning content and the Web applications that control and display that content to learners (Learning Management Systems or LMSs), interoperable in a standard way.

If you have a SCORM 2004 conformant LMS, you can be fairly certain that learning content that is also SCORM 2004 conformant will integrate successfully with your system.

If you develop material internally, this means that you can purchase virtually any SCORM conformant authoring tool that meets your needs, and as long as the content it generates is 100% SCORM conformant, you can be pretty sure that you'll be able to successfully launch and track it from your Learning Management System.

If you work with third party vendors to create learning content, the vendors you select to create training material for you (or from whom you wish to buy off-the-shelf content) don't need to be intimately familiar with your LMS. Any vendor that can create SCORM content can create online learning material that will run on your LMS, with no additional special knowledge of your specific system. And it doesn't matter what authoring tools they use internally. As long as the end-result is SCORM conformant content, you don't need to care.

If that was all that SCORM did, it would be incredibly important to the e-learning community. But the SCORM does far more than this (which we'll get into a little bit further along).

What is SCORM?

SCORM stands for Sharable Content Object Reference Model.

But seriously, what *is* SCORM?

It's a reference model.

No, really… what is it?

Ok, hold on. Let me explain what we mean by reference model.

The SCORM pieces together a comprehensive picture (or model) of how a Learning Management System might serve up Web-based learning content to learners in a standard way. It creates this picture by referencing a variety of sometimes unconnected technical standards, specifications and guidelines developed within the e-Learning community at large (there are a number of organizations developing these standards, and I'll talk more about them later).

In a way, the SCORM doesn't really give the e-Learning community anything new; it just brings together existing standards, specifications and guidelines from a variety of sources to create a bigger picture of how an online learning content system might work.

The SCORM focuses on the interface between content and the Learning Management System. In other words: how learning content gets into an LMS system, how it gets presented to learners, and how the learner's progress within that content is communicated to the LMS.

The SCORM presents a model that system developers (those creating Learning Management Systems) and content developers (those creating Web-based learning content) can follow equally. If implemented successfully, it means that learning content developed to be SCORM 2004 conformant will be easily integrated into any LMS which is also SCORM 2004 conformant.

If you'll excuse an overly simple analogy: a VHS tape will play in any VHS Video Cassette Recorder. It doesn't matter who made the tape, or who made the VCR. VHS tapes play in VHS players. Period. End of story.

Of course, things are never that simple in the world of software. But this type of interoperability is a big part of the vision behind the SCORM.

What do you mean by "Sharable Content Object"?

The concept of Content Objects comes from the mysterious world of software programming. To answer the question you'll have to listen to a brief history lesson. Sorry. I'll make it as brief as I can.

Back in pre-historic times, when slide rules were still cool, computer programmers wrote software using programming languages known as "procedural". Procedural programming languages were sort of like a book. They started at the beginning and they went through until the end. The computer would execute the first line of code in the program, and then go on to the next, and so forth. This worked for a while… and things were good.

As time went on, however, computer programs became more complex. Users had to interact with the programs and make choices and selections. Procedural programs had to branch and jump from section of code to section of code - rather like a reader flipping through a book and having to go back and look things up, jumping from chapter to chapter, going to the index to find something specific, etc.

The more jumping around a program did, the more difficult it was to execute in a procedural language. The idea of programs starting at the beginning and working through to the end was pretty much obsolete.

Sometimes programs were built with so many jumps and cross-references that the term "spaghetti" was used to refer to code that was so convoluted and interlinked that it was difficult to make sense of. All that jumping around and cross-referencing made it hard to both maintain and to re-use portions of programming code. It was tricky to copy bits of one "book" and stick it into another "book".

The old metaphor of the book simply didn't work anymore, and a new metaphor was needed. Then some smart cookie noticed that humans tend to perceive the world as being made up of objects. Objects are discrete, identifiable elements that have known features and functions. Instead of writing programs that worked procedurally, new "object-oriented" programming languages were developed that used the object metaphor.

You can build bigger, more complex objects out of a collection of smaller, simpler objects. For example, a car is a complex object made up of hundreds of smaller objects all working in concert. While each individual object may have been crafted to work within that particular vehicle, parts can be re-used in new vehicle designs. Standards can be developed so that parts can be interchangeable.

In the same way, programming code can be developed as individual self-contained objects that have known attributes and functions which work with other objects in an understood manner. This allows for ease of maintenance and the ability to easily re-use "objects" from one program in another program.

Still with me?

Then let's go back to e-learning. Before the SCORM, most e-learning courses were developed like old-style computer programs. They worked in isolation, and they did what they were supposed to, but it was difficult to re-use portions of the content or to combine them in new ways.

Someone wondered if the idea of "object-oriented" could be applied to e-learning content. What if you thought of a Web-based course as a collection of "learning objects" or of "content objects"? Each learning object would be a small single "nugget" of information, knowledge or learning that could be combined to make a course.

If you did this, you could share learning objects between courses and you could easily remove or add objects to existing courses. The Learning Objects or Content Objects would be "Sharable".

One way to think about the concept of Sharable Content Objects is to imagine a child's set of Lego building blocks. Lego blocks come in different sizes and shapes, however, they have a standard interface (bumps on the top and holes on the bottom) which allows the builder to easily assemble them and take them apart again.

So, Sharable Content Objects are encapsulated bits of Web-based learning content that have a standard interface that can be assembled and re-assembled in an understood manner.

Who Created the SCORM?

The SCORM was created by an organization known as the Advanced Distributed Learning initiative (ADL). The ADL was originally setup by the United States Department of Defense (DoD). In 1999, the White House, through Executive Order 13111, tasked the DoD to take the lead in working with other Federal agencies, academia and industry to develop common specifications for technology-based learning to help meet national education and training needs and provide best practice guidance to other Federal agencies. The DoD established the first ADL co-laboratory to begin the development of technologies that enhance learning and performance across the DoD and other Federal agencies. Since then other United States Federal agencies and academic institutions as well as international partners in the United Kingdom and Canada have joined the initiative.

The ADL is certainly more than just the SCORM. The eventual goal of the ADL is to create Intelligent Tutoring Systems that can provide individualized instruction and learning for a distributed workforce or student group. An Intelligent Tutoring System is a computer system that can evaluate the learning needs of an individual student and provide them with the right instruction at the right time in the way which best fits the needs of the individual learner. The full vision of Intelligent Tutoring Systems is still out of reach; however, the SCORM is a step towards this goal.

Why was the SCORM Created?

The SCORM was created and continues to be guided by six key high-level requirements:

- **Accessibility**: Make learning content accessible and available from remote locations.
- **Adaptability**: Allow instruction to be tailored to the needs of the individual learner and the organization.
- **Affordability**: Reduce the time and costs involved in developing and delivering learning content.
- **Durability**: Learning content should not need to be significantly redesigned, reconfigured or reprogrammed to keep up with the evolution of technology.
- **Interoperability**: Learning content developed in one location with one set of tools can be used in another location with a different tool set or platform.
- **Reusability**: One piece of learning content can be used in multiple applications and contexts.

The SCORM operational principles evolved out of the core requirements listed above and provide the following important capabilities:

- The ability of a Web-based LMS to launch content that is authored using tools from different vendors and to exchange data with that content.

- The ability of Web-based LMS products from different vendors to launch the same content and exchange data with that content during execution.
- The ability of multiple Web-based LMS products/environments to access a common repository of executable content and to launch such content.

Does SCORM Handle Offline Content Like CD-ROMs or Traditional CBTs?

No.

The SCORM makes the assumption that the Web is the best method to distribute learning content for ease of access and re-use. The SCORM therefore only deals with Web-based learning content and does not take into account other delivery mechanisms. This is important for business stakeholders to remember as many corporations still use non-Web-based computer-based training of one kind or another (e.g. distributing learning content on CD-ROM, DVD-ROM, etc.). The SCORM is not intended to embrace alternative methods of delivery, including online systems that are not Web-based.

Is there more than one version of the SCORM?

Yes. The SCORM is an ongoing initiative and has evolved and changed substantially since its first release. As well, remember that the SCORM refers to other standards and specifications, some of which are also evolving. There are currently three main releases of the SCORM:

- SCORM 1.1
- SCORM 1.2
- SCORM 2004

It is important when talking to a vendor, whether a courseware developer or Learning Management System provider to know which version of the SCORM they conform to. It is not enough to know that a vendor produces "SCORM Conformant" content, for example. There are significant differences between the versions, and knowing which version a vendor adheres to is critical.

To be SCORM conformant in a given version, an LMS provider does *not* have to explicitly provide support for previous versions of the SCORM, so an LMS that is SCORM 2004 conformant may not fully support SCORM 1.2 or SCORM 1.1 conformant content.

What happened to SCORM 1.3?

The SCORM is made up of three main components or documents (we'll go into more detail on this later). Each document handles a different aspect of the SCORM.

The ADL decided with the latest release to start versioning the documents independently of each other. All three documents in SCORM 2004 are currently at version 1.3, so in a way, SCORM 2004 *is* SCORM 1.3. But from hereon, each document will have its own version number and may be released separately. This may make things more complicated in the future when trying to determine what is meant when a vendor tells you that their application and/or content is "SCORM conformant".

With All These Versions, is SCORM stable?

Since SCORM is an evolving model, this is a valid question. There were large scale changes between SCORM 1.1 and SCORM 1.2. SCORM 2004 refines a lot of the existing material and adds an entirely new major section to the documentation. If the SCORM continues to grow and change, is it sensible to adopt it as a standard for content or Learning Management Systems?

The simple answer is, yes. The latest version of SCORM is stable and the ADL has promised that we will not be seeing major changes to the existing standards in the near future. The ADL is now focusing its attention on ways to expand the scope of the SCORM and to add new aspects to the existing documentation. To be sure, there will probably be future updates, but they should not be earth-shattering, and it is likely that content which is SCORM 2004 conformant will be interoperable for some time to come. Systems built to be SCORM 2004 conformant will probably not need major overhauls due to the next release of the SCORM document set.

In McGill's opinion SCORM 2004 is a stable foundation for the development of content and applications.

What is an LMS (according to SCORM)?

There are many products in the marketplace that claim to be Learning Management Systems, but a precise definition of what a Learning Management System is can be hard to come by.

The SCORM defines an LMS quite narrowly, as the SCORM is only interested in Learning Management Systems insofar as they relate to the delivery and tracking of Web-based learning content and learner's interaction with that content.

The SCORM defines an LMS as: "a server-based environment in which the intelligence for managing and delivering learning content to students resides … the LMS determines what to deliver and when, and tracks progress and performance as the learner moves through the learning content."

Most Learning Management Systems also provide a mechanism to enroll students in courses, allow the tracking of live courses (whether online or offline), and may provide additional resources, features or tools. None of these features are covered by the SCORM, and being "SCORM Conformant" is no measure of an individual Learning

Management Systems complete feature set, but only that it manages and delivers Web-based SCORM content in a standard way.

What are the main components of the SCORM?

As mentioned earlier, the SCORM is actually a set of related documents. There are three main SCORM documents, plus an overview document. The three core documents that make up the SCORM are:

- Content Aggregation Model
- Run-Time Environment
- Sequencing and Navigation

The Sequencing and Navigation document is the newest of the three (having been introduced with SCORM 2004), and is the most likely to see changes in the near term. The Content Aggregation Model and Run-Time Environment are the most stable of the documents and may not see major revisions any time soon.

What is a Content Aggregation Model?

The SCORM Content Aggregation Model (CAM) document deals with the assembly, labeling and packaging of Web-based learning content. The CAM lays out the rules and mechanisms by which individual assets or files can be combined into Sharable Content Objects (SCOs) and how SCOs can be combined to form Organizations. An Organization may be any grouping of SCOs and could refer to a lesson, a module, a course, a curriculum or virtually any other grouping that would make sense to your organization.

The CAM also lays out specifications for adding Meta-data to your Organizations, SCOs, and assets. Meta-data is data about data. If your course were a book, then the meta-data could be compared to the card catalog entry which describes that book. Each level of detail within your course may have its own meta-data, so that individual SCOs and even assets could be searchable within your system and discoverable.

The concept of discoverable content is extremely important to the philosophy behind the SCORM. The ability to find the right content based on the needs of the learner (or the developer) promotes re-use and just-in-time training. Adding keywords and other descriptive meta-data to SCOs, assets and Content Aggregations allows them to be searched and found when needed.

There are basically 5 major topics covered by the CAM document:

- The Content Model
- The Content Package
- The Manifest
- Meta-data

What's the Content Model?

The Content Model describes the SCORM components that make up a learning experience and explains how those components can be combined (or aggregated). The components of the Content Model are:

- Asset
- Sharable Content Object (SCO)
- Content Organization

One or more assets combine to make a SCO, and one or more SCOs combine to make a Content Organization.

Assets are the lowest level of organization and they represent the individual files that make up a learning resource. The assets could be: graphics, HTML documents, XML documents, Flash files, audio/video files, etc., etc.

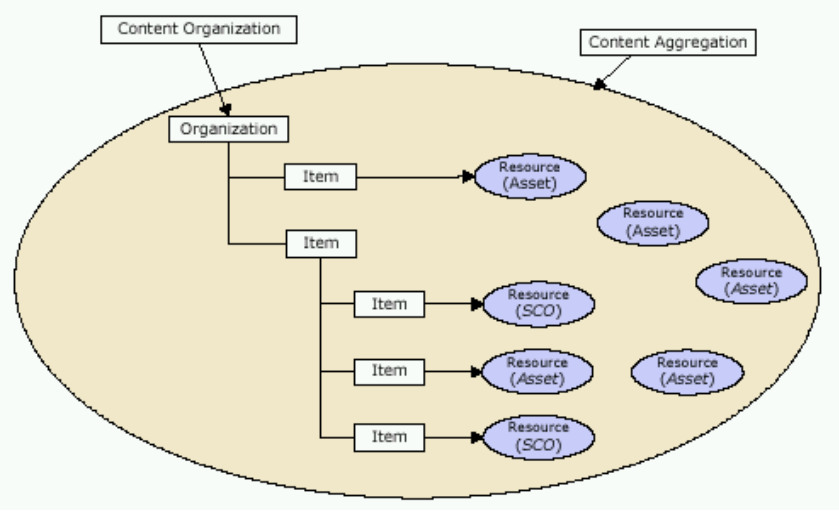
We've already talked about SCOs, so you should know the basics of what a SCO is by now. The CAM document defines a SCO as "a collection of one or more assets that represent a single launchable learning resource that utilizes the SCORM RTE to communicate with LMSs."

The LMS must be able to launch a SCO and that SCO must be able to communicate with the LMS. SCOs are the lowest level tracked by the LMS. By comparison, assets are not tracked by the LMS, and may not be launchable.

A Content Organization is essentially a collection of SCOs and/or assets organized in a particular fashion. A Content Organization could be a portion of a course, a complete course or even a set of courses. In most cases a single Content Organization will be developed for each course.

The CAM document defines Content Organization as "a map that represents the intended use of the content through structured units of instruction (Activities) … The Activities represented in a Content Organization may consist of other Activities (sub-Activities), which may themselves consist of other Activities." A Content Organization is a tree-view with any possible levels of branching. The "leaf nodes" or end-points of the tree-view will have an associated SCO or launchable asset which is used to perform the activity.

The diagram below is taken from the ADL's SCORM Content Aggregation Model (CAM) Version 1.3 document and illustrates a Content Organization:



What's the Content Package?

The SCORM Content Package is the collection of all files required to run the content presented in a standard format. Content Packages enable learning content to be exchanged between developers and LMSs. When a developer completes a piece of learning content (using whatever SCORM authoring tools or systems that they prefer), they can package that content up into a Content Package and then deliver it to a SCORM Learning Management System (or Learning Content Management System).

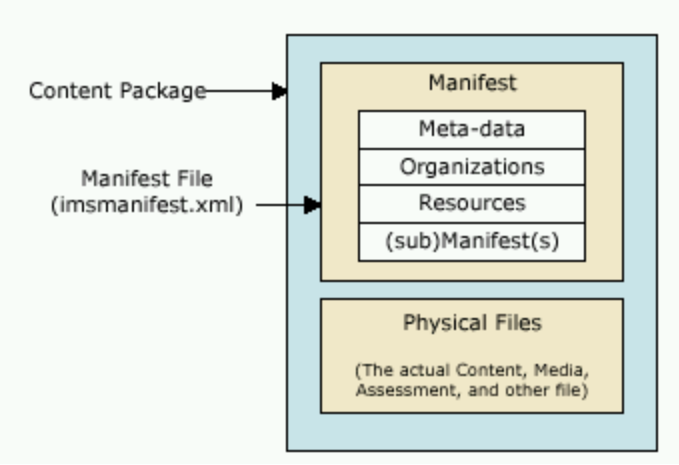
A Content Package could contain a portion of a course (such as a module, chapter or lesson), a complete course, or could represent a collection of courses. But the package should be able to stand alone and it should contain all the information needed to deliver the content contained within it.

Usually, a Content Package is presented as a single PIF or "Package Interchange File". This is simply a compressed archive file (.zip) that contains all elements of the Content Package. This .zip file can be easily distributed across the Internet and the contents can be extracted from the archive upon receipt.

A Content Package is comprised of two main components:

- The Manifest: Also known as the IMS Manifest, the Manifest is an XML document which describes all content, its organization and the resources contained in the content package.
- The physical files: The Content Package contains the actual files that make up the learning resources.

The ADL's SCORM Content Aggregation Model (CAM) Version 1.3 document illustrates this idea:



What is the SCORM Run-Time Environment?

The second core document of the SCORM is the "SCORM Run-time Environment" (or "RTE"). This document describes how Learning Management Systems launch SCOs and communicate with them at run-time.

It is the responsibility of the LMS to launch SCOs based on requests from the learner and the sequencing rules of the Content Organization. The LMS will only launch one Content Object (SCO) per learner. In other words a learner cannot have more than one Content Object open at any given time.

The SCORM does not detail how navigational controls are presented to the learner to allow them to select content objects. Controls could be embedded in the SCO or presented in a frame or even in a separate window. The SCORM only specifies what type of navigational requests a learner can make and how the LMS should respond when such a request is made.

The LMS can launch either launchable assets (which are not tracked and do not communicate with the LMS) or Content Objects (which are tracked by the LMS and must communicate). In order for the LMS to track a student's progress with a Content Object, that Content Object must be able to report back to the LMS. In some cases the Content Object needs to know things about the learner (e.g. their name, whether they've taken the content before, their last score, etc.). So the communication between Content Object and LMS must be two-way. This communication is achieved through the SCORM API.

What's an API?

API stands for "Application Programming Interface", and it is basically a defined set of methods that a Content Object (in the case of SCORM) can use to communicate back and forth with an LMS. An LMS must provide an implementation of the SCORM API for use by Content Objects.

SCORM Content Objects use ECMAScript (commonly known as JavaScript) to find and call the functions of the LMS-provided API. All communications with the LMS are initiated by the Content Object.

The Content Object must find the API and then tell the LMS when it has launched successfully (Initialize()) and when it is being closed (Terminate()). The Content Object can get certain values (GetValue()) from the LMS database, and can also set certain values (SetValue()).

Here's a list of the functions that the API provides to the Content Object:

- Initialize()
    - This function is used to initiate communication between the SCO and the LMS. All SCOs need to call this function as the first thing they do upon launching.
- Terminate()
    - This function is used to terminate the communication between the SCO and the LMS. This is usually the last thing that a SCO will do.
- GetValue()
    - The GetValue function allows SCOs to get data from the LMS as well as being able to determine what types of data are available. The SCO can get any value from the SCORM RTE Data Model made available by the LMS.
- SetValue()
    - The SetValue function allows SCOs to save data to the LMS for storage. The SCO can save values from the SCORM RTE Data Model as made available by the LMS.
- Commit()
    - This function allows the SCOs to request that any data transmitted to the API be stored in the LMS. Prior to calling this function, data may be stored in the API cache and not written to the LMS database. This call ensures that data is written to the LMS. The Terminate function will also do this, however, calling Commit() is a precautionary method which protects against an interruption in the communication between SCO and API prior to the SCO Terminating properly.

What's the SCORM RTE Data Model?

The SCORM RTE Data Model is a common set of defined data fields or data elements that allow SCOs to communicate with different LMS environments. This set of data includes information about the learner, interactions that the learner had with the Content Object, completion status, objective information and more. The data elements defined by the Data Model are the elements that the SCOs can read or set using the GetValue() and SetValue() functions as listed above.

In previous versions of the SCORM, certain data elements were "optional" which meant that LMSs did not have to implement all the data elements, only the mandatory elements. This meant that the onus was on the SCO to question the LMS through the API to find out whether a specific optional data element was supported or not. With the release of SCORM 2004, all data elements in the data model must be supported by conformant LMSs. All the data elements are optional for SCOs, however. The only requirement on SCOs is that they call Initialize() and Terminate(). A SCO is not required to get or set any data values. However, if they do use data from the LMS, then they must conform to the SCORM Data Model.

All data elements within the SCORM RTE Data Model are prefaced with a "CMI" (which stands for "Computer Managed Instruction" and is a holdover from the original organization which designed the Data Model standard (The AICC or Aircraft Industry CBT Committee). Although the Data Model originated with the AICC, it was recently submitted and approved by the IEEE LTSC as an official standard. In the process, some important changes to the data model were made since the last version of the SCORM was released.

Can SCOs Store Their Own Progress Information?

Some larger SCOs may need to store their own progress. For example a SCO made up of 10 pages may want to track that a learner had viewed 3 out of the 10 pages. The LMS does not care about this level of progress within the SCO. The SCO is the smallest unit that the LMS tracks or cares about.

However, the SCORM RTE Data Model does provide SCOs that ability to store their own data. A SCO can use either of the following data elements to store internal progress information:

- Location
    - The cmi.location data model element is intended to represent a location within the SCO (e.g. page 3 of 10). This could be a "bookmark" or a "checkpoint" which would allow a learner to return to the point in the SCO at which they last left the SCO.
- Suspend_data
    - It is the SCO's responsibility to provide a way in which a learner can suspend the current learner attempt. If a learner leaves a SCO at mid-point, the state of the learner's progress within the SCO can be saved in the cmi.suspend_data element. This data element could be used if the information to be stored is more complex that a simple location that might be stored in cmi.location. In either case (Location or Suspend_data), however, the SCO must explicitly inform the LMS that the learner is not finished with the SCO by setting the Exit data element (cmi.exit) to "suspend" or "logout". If the Learner's experience with the SCO is over, then any run-time data like Location or Suspend_data may be thrown out by the LMS.

What Sort of Data Can a SCO Access About a Learner?

Well, not much really. A SCO can find out the learner's unique identifier within the LMS system (cmi.learner_id) and a SCO can get a learner's name using cmi.learner_name. A SCO can also determine a learner's preferences using cmi.learner_preference (which includes audio level, language, delivery_speed, and audio_captioning). Not much else about an individual learner can be determined by the SCO using the standard SCORM RTE Data Model.

What is the SCORM Sequencing and Navigation?

This is the newest document in the SCORM set. It was added with SCORM 2004 and is the most likely to see major changes in the near term. While other aspects of the SCORM can be considered stable and unlikely to change substantially, the same cannot necessarily be said about the Sequencing and Navigation document.

The Sequencing and Navigation book is primarily based on the IMS Simple Sequencing (SS) Specification. Despite being called "Simple", the IMS SS is quite complicated and allows for a great deal of control. It is only called simple because it defines a limited number of commonly used sequencing behaviors.

The core concept of SCORM Sequencing is the Activity Tree.

What is an Activity Tree?

An Activity Tree is a defined structure of learning activities used to describe the hierarchical relationship of a learning experience. Previously we described how the Content Aggregation Model developed a Content Organization. The Content Organization is the basis of the Activity Tree. Each of the items indicated in the Content Organization correspond to a learning activity. Rules for sequencing can be attached to each element or activity to determine how an LMS should sequence the activity.

A SCORM-conformant LMS translates the Content Organization into an Activity Tree. An individual Activity Tree is created for each learner experiencing the course or learning activities and includes the tracking status information for each activity in the structure.

Each learner's experience with the same content structure may therefore be different, based on the sequencing information that the content developer defined in the Content Package as well as the learner's specific interactions with experienced content objects.

What's an Activity?

An Activity or Learning Activity is described in the SCORM as "a meaningful unit of instruction". This may seem a little vague. If we think of SCOs and assets as learning resources, then an Activity may be a learning resource, but it can also be a collection of sub-activities (each of which may be either a collection of further sub-activities or a learning resource).

This means that an entire course would be considered a single Activity (made up of many sub-activities).

Activities that have no sub-activities are known as "Leaf Activities" and they are always associated with a learning resource or Content Object. These activities are called "Leaf Activities" because they represent the end point of a branch on the Activity Tree.

What's a Cluster?

A cluster is a learning activity that has sub-activities or child activities. For example, a module that is made up of three lessons is a cluster. The cluster includes the module (the parent) and the lessons (the children) that comprise it. An activity which has no children (a leaf node) is not a cluster. In the example of the module and chapters above, if the chapters have no sub-sections, then they are not clusters themselves. However, if one of the chapters was made up of three sub-sections then that chapter and its sub-sections would be a cluster as well. The parent activity in a cluster will contain the information about the sequencing strategy for the cluster.

What's an Attempt?

A learner's attempt to complete an Activity is known as an Attempt. If a learner is experiencing a specific Content Object, then that Object's Activity is being attempted… and so is the that Activity's parent (if it has one), and the parent Activity's parent, etc. as far as the Activity Tree reaches.

If the tracking status of an Activity changes (for example a Content Object is completed) then the tracking status of its parents may also change. This is called "Rollup".

What are Learning Objectives?

Learning Objectives are separate from Learning Activities. An Activity may have one or more Learning Objectives associated with it. Each Learning Objective associated with an Activity will have a set of tracking status information that allows learner progress toward the Learning Objective to be tracked. Learning Objectives can contribute to "Rollup", and can therefore influence the completion of their parent's Learning Objectives.

There are two types of objectives: Local and Global. Local Objectives are associated with a single Learning Activity and cannot be referenced by other Activities (except via standard Rollup). Global Objectives are not directly associated with Activities, but Local Objectives can reference Global Objectives. Multiple Local Objectives can reference the same Global Objectives, allowing Global Objectives to be shared and referenced by multiple activities.

In practice this feature can provide the learner with different ways to complete a course. The learner could, for example, complete any one of a set of activities to achieve course completion. The completion of one activity makes the completion of the other activities unnecessary for overall course completion. For example, completing any one of Content Objects A, B or F is all that is required for course completion.

What are Sequencing Control Modes?

Each Activity Cluster can define one or more Control Modes which can serve to constrain the way in which a learner accesses the sub-activities within the Cluster. The available Control Modes are as follows:

- Sequencing Control Choice
    - If set to True, this allows the Learner to freely choose any sub-activity within the Cluster
- Sequencing Control Choice Exit
    - This is a constraint on the Choice mode listed above. If this is set to True, it prohibits the Learner to target activities which are not descendents of the affected activity for a navigation request. For example, if the Activity were a Lesson made up of three Content Objects, the child Content Objects could be navigated to, but the Learner could not freely navigate out of the Lesson.
- Sequencing Control Flow
    - If set to True this indicates that the Learner can navigate through the child activities of the cluster using "Continue" and "Previous" to move from sub-activity to sub-activity.
- Sequencing Control Forward Only
    - This is a constraint on the Flow control listed above. Learners may only use the "Continue" navigation request to move forward through a set of child activities.

What are Sequencing Rules?

Each Activity can have zero or more Sequencing Rules applied to it. The Rules are evaluated at specified times during various sequencing behaviors. Based on a set of Conditions being true, certain actions may be triggered.

The Sequencing Rules allow the learner's experience of the Activity Tree to be dynamic and personalized based on their progress and success at completing activities and objectives. Depending on the content objects and objectives that they complete or fail to complete, they may be presented with a different set of activities or Content Objects.

Conditions for the rules can be the satisfaction or partial satisfaction of a Learning Objective, the completed or attempted status of an activity, etc. The types of actions that are possible based on the rules are:

- Pre-Condition Actions: These occur when traversing an Activity Tree to determine an Activity for sequencing.
    - Skip – The activity is not considered a candidate for delivery during a flow sequencing request.
    - Disabled – The activity may not be the target of any sequencing or delivery request.
    - Hidden from Choice – The activity may not be the target of a "Choice" sequencing request.

    - Stop Forward Traversal – The activity will prevent activities following it (in a preorder traversal of the tree) from being considered candidates for delivery.
- Post-Condition Actions: These Actions occur when an attempt on a given activity terminates:
    - Exit Parent – Process an Exit Parent termination request.
    - Exit All – Process an Exit All termination request and return an Exit sequencing request.
    - Retry – Return a Retry sequencing request.
    - Retry All – Process an Exit All termination request and return a Start sequencing request.
    - Continue – Return a Continue sequencing request.
    - Previous – Return a Previous sequencing request.
- Exit Action: Apply after a descendant activity's attempt terminates:
    - Exit – Unconditionally terminate the activity.

Can the Order of the Content be Randomized?

Yes. Activities within a cluster can be presented in a randomized order. The randomization can occur every time the learner accesses the activity or only once.

This Simple Sequencing Seems Very Complex…

The specifics of the specification are certainly complex and not "simple" by any means. What can be done with the Simple Sequencing Specification is not really overly complex and encompasses fairly standard sequencing rules and methodology employed by many legacy CBT and WBT applications. The difference is that old CBTs and WBTs usually dealt with sequencing in a proprietary and unique fashion for each course. The SCORM Sequencing and Navigation specifications create a common way to achieve these same standard sequencing behaviors in a generalized fashion.

Hopefully authoring tools will be developed that will conceal the low-level details from content authors and instructional designers and allow them to focus on the different ways in which they want their content to be presented, flowed and delivered, as opposed to the specific structure of the conditions, rules, activities and objectives.

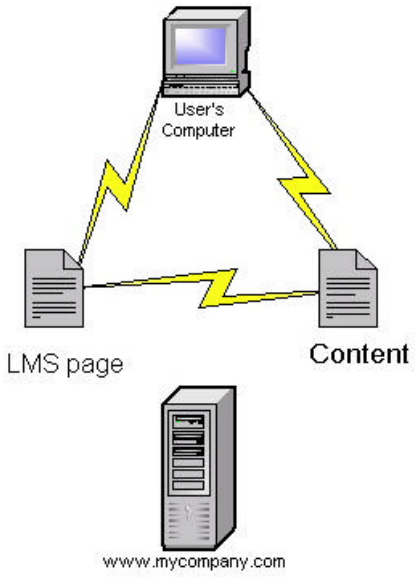
The SCORM Sequencing and Navigation specifications allow for a wide array of possibilities and the ability to personalize a learning experience based on a learner's progress through the content. It is a significant step forward in the SCORM, but it is not for the faint of heart. If you are not using an authoring tool that handles the nitty gritty aspects of Sequencing, it is strongly advised that you employ an experienced developer with solid programming experience to create your Sequencing.

What's the Cross-Domain Scripting Issue?

This is a problem with the basic premise of SCORM that comes to light when your content is located on a separate server from your LMS. In many e-learning implementations, content is imported onto the same server that the LMS resides on. In this scenario, you don't have to worry about Cross-Domain Scripting.

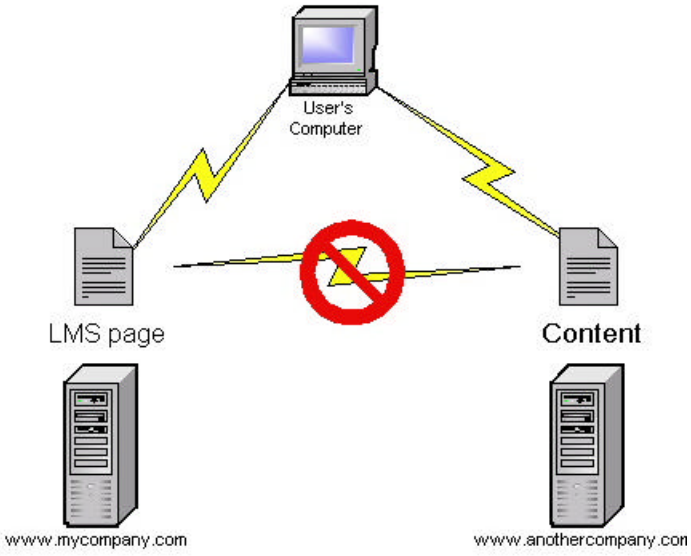
However, two of SCORM's main tenets are *accessibility* (learning content should be accessible and available from remote locations) and *interoperability* (content developed in one location with one set of tools can be used in another location with a different tool set or platform). Having learning content located in one or more distributed content repositories and launched by one or more remote LMS's seems like it fits within the vision of the SCORM. While it may indeed fit within the philosophy, modern browser security restrictions have thrown a bit of a wrench into this vision.

As previously mentioned, the SCORM run-time environment relies on SCOs communicating with the LMS using an ECMAScript or JavaScript API provided by the LMS (usually in a parent window or frame). So there are two pages that need to talk to one another. One is the content, and the other is the LMS. If both pages are on the same server, there is no problem and they can talk to each other using JavaScript:



LMS Page and Content
Can Communicate

If, however, the content is on a different server then the two pages will not be able to talk to each other using JavaScript (at least not without a workaround):



Cross Domain Scripting Issue - LMS Page and Content Cannot Communicate via JavaScript

To understand this, you should know that JavaScript is a scripting language that is executed on the client-side. JavaScript is interpreted by the user's browser and executed on the user's machine (not on the server). This means that the execution of JavaScript is controlled by the user's browser, and different browsers may interpret JavaScript or place rules upon it in different ways (for example, most browsers provide a security feature that allows users to disable JavaScript altogether).

Allowing pages on two different servers to talk to each other is a security risk, and most modern browsers have disabled this functionality altogether.

If you want to host your content on a server different from your LMS, you will basically have to "trick" the user's browser into believing that the content and LMS are in fact on the same server (even though they are not).

Internet Explorer browsers still provide a loophole which allows pages within the same top-level domain (e.g. content.mycompany.com and lms.mycompany.com share the same top-level domain, although their second level or sub-domains are different) to alter their domain information so that they appear as if they are both on the same server (e.g. mycompany.com). This is called the document.domain solution, and it only works on Internet Explorer. This feature has been continually restricted by Microsoft with subsequent releases and security patches. There is no way to know how long this loophole will last.

Other ways of tricking the user's browser include:

- using a proxy server to redirect the HTML Response from the LMS and the content server so that they appear to be coming from the same server; or
- using a software application on the LMS server to rewrite and re-direct URLs so that content pages would appear to come from the LMS server.

The ADL has prepared an in-depth paper discussing this issue and possible solutions for it. If you or your team are experiencing problems with the Cross-Domain Scripting issue, I would recommend obtaining this document from the ADL Website ("Cross-Domain Scripting Issue Version 1.0": http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=912).

In summary, however, the Cross Domain Scripting issue is not a bug, or a flaw in the SCORM, but a security feature in modern browsers that needs to be accounted for.

What's Not Covered by the SCORM?

The SCORM has had a major impact on the development of e-Learning content, authoring tools and systems; however it is not all-encompassing. The SCORM is actually relatively narrowly defined and was not intended to cover all aspects of e-Learning. So knowing that a system, tool or content piece is SCORM conformant is not enough to ensure that it will meet your needs.

In short, the SCORM does not answer (nor is it intended to answer) all questions about e-Learning standards. Your organization will need to develop or adopt additional standards to deal with these non-SCORM issues.

Here is a brief list of things that are not covered by the SCORM:

- Disc-based Media
    - The SCORM does not deal with how learning content could be distributed (and tracked) on disc, rather than through the Web. The SCORM is strictly a Web-based online model.
- Look and Feel.
    - The SCORM does not touch on the area of look and feel. How content looks to the learner, or how an LMS navigational interface is displayed is not covered.
    - There are no standards for screen size of content.
    - Since there are no SCORM standards for how content should look, this means that content which is re-used may look out of place when used in a different context.
- Where and How Navigational Controls are Presented to the Learner
    - The SCORM states that navigational controls can be either in the SCOs themselves or presented by the LMS. Since there is no standard or recommendation, it is possible for LMSs and SCOs to provide redundant navigational controls. Since each SCO could create navigational controls with their own look and feel, it is also possible to have inconsistent navigational controls that could confuse the learner.
- Abstracting Content from Presentation

    - Using XML, Style Sheets and other methods, it is possible to keep content (text, images, video, etc.) separate from the way it is graphically presented to the learner. This allows the same content to be dynamically re-purposed and presented with a new layout or treatment based on the needs of the Learner.
    - The SCORM does not address this.
- Mobile Learning
    - The SCORM does not directly address how learning content could be delivered to wireless or mobile devices.
- Instructional Integrity
    - SCORM conformance in no way implies that content is instructionally sound.
- The technology or platform of a Learning Management System
    - A SCORM Learning Management System can be developed in virtually any Web technology (as long as an API is made available through ECMAScript/JavaScript).
    - An LMS could be developed in Microsoft .NET, Java J2EE, PHP, Perl, etc., etc.
- The performance of a Learning Management System
    - Being SCORM conformant is no measure of how well a Learning Management System will perform.
    - An LMS can be 100% SCORM conformant, certified and tested, and still perform extremely poorly or be unable to handle a large load of students.
    - SCORM conformance only indicates that an LMS tracks and launches content in a standard way.
- Collaborative Learning
    - The SCORM does not explicitly deal with scenarios where students are working together or sharing a learning experience online.
- Virtual Classroom
    - The SCORM focuses on the delivery of self-paced (asynchronous) learning material. It does not address synchronous or real-time virtual classroom scenarios like Centra or NetMeetings.
- Live Training
    - The SCORM has nothing to do with live face-to-face training in a traditional classroom setting. Most LMSs will provide features to track live training, but this is outside the scope of the SCORM.
- Student Registration / Enrollment
    - The SCORM has nothing to do with how students enroll in courses or register for training.
- Multimedia Formats
    - The SCORM does not deal with specific multimedia formats or versions, such as Flash, Director Shockwave, QuickTime, 3D viewers, mp3s, etc. etc. As long as the content is Web-based and can be served up by a standard Web browser, the SCORM makes no recommendations or restrictions on media types.
- Content that makes use of server-side scripting or components

- Content which makes use of server-side scripting or components (Java servlets, ASP, etc.) and/or database access are beyond the scope of the current SCORM. Such content is not strictly SCORM conformant at this time, as it is not truly portable. Using a variety of methods it is possible to create workarounds that will allow such content to be launched and tracked by a SCORM RTE, but the SCORM does not explicitly deal with type of content.

References

1. Advanced Distributed Learning (ADL), *Cross-Domain Scripting Issue Version 1.0.* 2003. Available at: www.adlnet.org.

2. Advanced Distributed Learning (ADL), *Sharable Content Object Reference Model (SCORM) 2004 2nd Edition.* 2004. Available at: www.adlnet.org.