eLearning

(Research and Review)

A Report

Submitted by

VINKLE KUMAR SRIVASTAV

(Programmer and Research Associate)

****

**Neurosurgery Skills Training Facility,**

**Neurosurgery Education and Training School (AIIMS+CSE-IIT-D)**

**Dept. of Neurosurgery, All India Institute of Medical Sciences,**

**Ansari Nagar, New Delhi, Pin-110029, India**

**November 2012**

# Table of Contents

1. Introduction 1

1.1 Types of learning delievery. 2

1.2 Learning Technology Ecosystem. 3

2. eLearning Standards 4

2.1 Why Standards. 4

2.2 How Standards evolve. 5

2.3 Introduction to ADL and SCORM 6

**2.3.1** SCORM Run Time Environment 7

 **2.3.1.1** Launching Content Objects 7

 **2.3.1.2** Run Time API 7

 **2.3.1.3** Run Time Data Model 8

**2.3.2** SCORM Content Aggregation Model. 8

 **2.3.2.1** Content Packages 9

 **2.3.2.2** Manifest Files. 9

 **2.3.2.3** Resources 10

 **2.3.2.4** Organizations 10

 **2.3.2.5** Metadata 11

 **2.3.2.6** Shareable Content Objects (SCOs) 11

**2.3.3** SCORM Sequencing and Navigation 13

 **2.3.3.1** Sequencing 14

 **2.3.3.2** Navigation. 15

3. Choosing an eLearning Platform 16

1. Introduction

eLearning is the use of computers using server based software system to create and deliver educational material to customer, prospects, employees and students. So it is about new blend of resources, interactivity, performance, support and structured learning activities.

 **Key Benefits.**

* Anywhere, anytime Learning.
* Maximize efficiency and reducing complexity by providing common access point for content.
* It manages the delivery and tracking of courses, manages critical points like safety issues, operating procedure and environmental standards.
	1. Types of learning delivery

Figure below shows the difference between synchronous and asynchronous delivery.

**Figure 1 (Synchronous Vs Asynchronous Learning)**

Typically there are 3 kinds of learning delivery.

* **Learner led delivery -** This is asynchronous on demand learning.
* **Instructor led delivery –** In this content is presented by the instructor either live or virtually.
* **Embedded –** It is instructor led learning which include performance through Just in Time help.
	1. Learning Technology Ecosystem.

Learning technology ecosystem has many moving parts. We have a courseware and we have different structure for that like Course. Unit, Module Asset etc. Sometime there is a server side display engine. There is User, Client or Browser. There are dynamic web application and web media that play into the web browser. There is a need of administrators to keep track of courses, assessments and activities. There is need of Profile management of the users and providing different level of access to different level of users.

A typical learning technology ecosystem is shown below.

fig.2 Learning Technology Ecosystem.

Courseware.

Course

Unit

Module

Asset

User/Browser

Internet

Profile/Registration

(People, Rosters)

Tracking

(Data Management)

Reporting

Testing

Content/

Course

Admin

Web Apps/

Web Media

* Content Format/

 Packages

* Protocols
* Data Files

`

**2.** **eLearning Standards**

eLearning is attaining significance in this Internet world due to the obvious advantages of anywhere anytime learning, to reach the unreachable and so on. It is a well known fact that explosion of technologies were due to the wide spread adoption of common standards. Standards impose certain order providing more uniform and precise access and manipulation to e-Learning resources and data. There are number of organizations working to develop specifications and standards such as IMS, ADL, ARIADNE, IEEE, ISO etc to provide frame work for e-Learning architectures, to facilitate interoperability, content packaging, content management, Learning Object Meta data, course sequencing and many more.

* 1. Why Standards.

Standards are desirable for interoperability, convenience, flexibility, and efficiency in the design, delivery, and administration. They provide consistent on-line dimension for all courses being designed so that all authors/ faculty are able to customize the on-line materials with minimal lead-time. Standards also impose a certain order on chaos resulted due to proprietary products from different vendors. This order provides more uniform and precise access andmanipulation to e-learning resources and data. So we need eLearning standards due to following reasons.

1. To enable interoperability on different platform.
2. Protection of investment on content development.
3. Exchange of content locally and globally.
4. Simplify integration and focus competition.
5. Reduce cost & create market for buyers and sellers.
6. Increased reach and flexibility enabling learners to engage in the learning process anytime, anyplace and on a just-in-time basis.
7. Decreased cost of learning delivery, and reduced travel, subsistence costs and time away from the job.
8. Increased flexibility and ability to respond to evolving business requirements with rapid roll-out of new and organizational-specific learning to a distributed audience.

2.2 How Standards evolve



 Typically standards evolve as Research and Development. Particular vendor find themselves developing integration more than one for multiple customers and rather than making that a services offering they began to making a product and began to develop reusable way to doing it.

So sometime because of this and sometimes independently specification consortium like AICC (Aviation Industry for CBT Committee), ARIADNE in Europe or the Doblin Core group which was group of library science experts to standardize on metadata framework.

So these groups define specification or documentation describing how one could potentially implement a system to do one of these components for integration.

From their things typically move forward to groups like ADL (Advance Distributive Learning initiative).Their People develop test beds like software that validate your implementation of the description, in fact compliance with the intent of the specification. They may also develop reference models which club together different specifications into one.

Things evolve from there to international standards bodies like IEEE or W3C. So these are international standard bodies which create accredited standards like ISO standards.

2.3 Introduction to ADL and SCORM

 ADL was founded in 1997 to standardize and modernize training and education delivery for U.S. Dept. of Defense (DoD).

The main focus of ADL is.

* Develop and implement learning technologies across DoD and federal government.
* Collaborate with government. Industry and academia to promote international specifications and standards for designing and delivering learning content.

 **ADL vision.**

Provide access to the highest quality education and training, tailored to individual needs, delivered cost-effectively, anywhere and anytime.



**ADL’s high level requirement**

1. **Accessibility**. Ability to locate and access instructional component from multiple locations and deliver them to other locations.
2. **Interoperability**. Ability to take instructional component developed in one system and use them in another system.
3. **Durability**. Ability to withstand technology changes over time without costly redesign, reconfiguration or recoding.
4. **Reusability**. Ability to to use instructional component in multiple applications, courses and contexts.

**SCORM (Shareable Content Object Reference Model)**

SCORM is nuts and bolts for the web based learning. It is a compilation of technical specifications for web-based e-learning. The SCORM standards are governed and published by the Advanced Distributed Learning Initiative (ADL).

* It describes how does content communicate with backend LMS through SCORM Run Time.
* How to package up the course, so that it can be moved from system to system.
* How to describe the course with metadata in a consistent way.
* How to sequence the course on the learner based interaction with the course.

So SCORM consist of three main categories.

* + - **SCORM Run Time Environment.**
		- **SCORM Content Aggregation Model.**
		- **SCORM Sequencing and Navigation.**

**2.3.1 SCORM Run Time Environment**

The SCORM Run-Time Environment (RTE) handles requirements for launching contentobjects, establishing communication between learning management systems (LMSs) and shareable content objects (SCOs), and managing the tracking information that can be communicated between SCOs and LMSs.

The SCORM Run-Time Environment (RTE) specification has implementation requirements for both learning management systems (LMSs) and shareable content objects (SCOs). It defines a common content object launch mechanism, a common communication mechanism between content and learning management systems, and a common data model for tracking a learner's experience with content objects.

**2.3.1.1 Launching Content Objects**

The RTE Launch process provides a common way to deliver web-based content objects to learners. Content objects that are engaged by learners are called learning activities. The RTE launch model requires that only one learning activity be accessed at a time per learner.

The launch process defines procedures and responsibilities for the establishment of communication between launched content objects and the run-time environment. The communication mechanism is standardized with a common API and data model.

In SCORM, there are two types of content objects: SCOs and Assets. Both can be engaged by a learner, but only SCOs utilize the RTE API and Data Model to communicate with the launching LMS. Characteristics of launchable activities are further described in the Activities and Trees section.

It is the responsibility of the RTE/LMS to "take away" a content object, usually in response to a learner or system triggered navigation event. Navigation events are further described in the Navigation section.

**2.3.1.2 Run-Time API**

 The Run-Time *API* (application programming interface) is the communication mechanism for informing the LMS of the communication state and between a content

object and the LMS (e.g., initialized, terminated, and/or in an error condition), and is used for retrieving and storing data (e.g., score and time limits) between the LMS and the SCO.

Only launched SCOs (not Assets) make use of the RTE API.

API usage guidelines:

* The function names are all case-sensitive.
* The function parameters are all case-sensitive. Data model parameters are to be represented as lower case.
* Each call to an API function, other than Support methods (GetLastError, GetErrorString, GetDiagnostic), sets the error code.
* All parameters passed between the SCO and the API instance are treated as ECMAScript characterstring. All parameter values shall be compatible with the data types and formats described by the data models that use the API for communication.

**2.3.1.3 Run-Time Data Model**

The Run-Time *Data Model* provides a standard set of data model elements used to define the information being tracked by for a SCO. In its simplest form, the data model defines data model elements that both the LMS and the SCO are expected to "know about". The LMS must maintain the state of the SCO's data model elements across learner sessions, and the SCO must utilize only these predefined data model elements to ensure reuse across multiple systems.

Both the RTE and a SCO rely on the data model for functionality. Values for the data model elements are set and retrieved by SCOs via the RTE API's GetValue() and SetValue() functions. Some elements are read-only (set only by the LMS).

**2.3.2 SCORM Content Aggregation**

For content developers, the biggest question is often "How do I make my existing e-learning content SCORM conformant?"

Content is generally compatible with SCORM if:

* It can be delivered via a web browser
* It can be self-contained (i.e. packaged with all dependencies wholly in a ZIP file)
* It does NOT depend on server-side scripting languages (such as JSP, ASP, and PHP)
* It does NOT depend on external files or external URLs
* It does NOT depend on downloadable components that must be installed by an administrator

General steps for making e-learning content SCORM conformant:

* Ensure content meets SCORM compatibility requirements (above)
* Organize all content files (including dependencies) into a single directory structure
* Define and describe the content using an XML manifest file as described by SCORM
* Package all the content and necessary files into a ZIP file

SCORM conformant e-learning content can be packaged, deployed to, and delivered via any SCORM conformant learning management system (LMS).

**2.3.2.1 Content Packages**

A SCORM content package is a self-contained ZIP file containing certain contents defined by the SCORM standard. The file is known as a Package Interchange File (PIF) and it contains all files needed to deliver the content package via a SCORM run-time environment and/or learning management system (LMS).

Mandatory Content Package contents:

* XML manifest file (imsmanifest.xml)
* All schema/definition (.xsd and .dtd) files referenced by the manifest file
* All resource files used by the content package and its learning activities

All the files that make up a content package must fit within the same directory tree structure within the PIF file. Your package may group everything into a single directory, or it may use sub-directories within the root (e.g., one directory for each SCO). All files used by the content package must be within the content package. References to external files or absolute URLs are not allowed.

**2.3.2.2 Manifest Files**

SCORM content packages contain an XML manifest file that describes the package and its contents. The manifest file is a structured inventory of the content of the package. The name of the manifest file is always ***imsmanifest.xml*** and it must appear in the root of the content package.

Mandatory Manifest File contents:

* Unique identifier
* Minimal metadata describing the package and its SCORM version
* One or more resource definitions listing all files required to launch and deliver each resource
* One or more organizations of learning activities

Optional Manifest File contents:

* Sequencing information for organizations
* More metadata for the content package, resources, and organizations

All references to files in the resources must be relative to the root of the content package. Referencing external files or absolute URLs is not allowed.

**2.3.2.3 Resources.**

In SCORM, a Content Object is a web-deliverable learning unit. At its most basic, a content object is just an HTML page or document that can be viewed with a web browser. A content object can use all the same technologies a web page can use (e.g., Flash, JavaScript, frames, and images). However, resources cannot be pages that require a server-side engine to process, such as ASP, PHP, or JSP pages.

Content objects are defined in a content package manifest file as a Resource, along with all the files it depends on. A Resource definition provides information about your learning object and how it may be used by a run-time environment.

Resources come in two flavors: *Assets* and *Shareable Content Objects* (SCOs).

An asset is a simple resource, such as a static HTML page or a PDF document, or collection of files, such as images and a style-sheet, which does not make use of the run-time API defined by SCORM. Therefore, an asset does not communicate with the run-time environment delivering it.

**2.3.2.4 Organizations**

A content package can define one or more organizations that describe how the resources are logically organized into a learning experience. An organization defines a hierarchical activity tree.

An Activity that has no children is called a *leaf*. Only leaf activities are actually delivered to a learner. And because they are delivered to a learner, each will reference a single launchable resource (i.e., a web-deliverable learning unit).

An activity that is not a leaf is the parent of a *cluster*. A cluster is a parent activities and all of its sub-activities. Activities can be nested within other activities, with no limit on depth.

Each activity has a title that is used when displaying the activity, for instance, in a table of contents.

Multiple activity trees may be defined for a single content package. The purpose of defining multiple organizations would be to allow a learner to experience the content in different ways.

The activity tree(s) should not be confused with the physical structure of the content package. The structure of content (defined by resources) and structure of learning activities (defined by organizations) are separate.

By default, a learner experiencing a content package will choose an activity from the tree to launch.

**2.3.2.5 Metadata**

Metadata can be used to describe elements of a content package in its manifest file. Metadata allows learning resources to be found when stored in a content package or in a repository. When a learning resource is intended to be reusable, it is a best practice to describe it with metadata. Describing learning objects with metadata facilitates their search and discovery across systems.

Metadata can be *context-independent* or *context-dependent*. Context-independent metadata usually refers to immutable metadata that describes digital assets, content objects, and the such. Context-dependent metadata usually refers to metadata that makes sense only in the context of a particular context organization.

**2.3.2.6 Shareable Content Objects (SCOs)**

A Shareable Content Object (SCO) is a launch able learning object (resource) that communicates with the run-time environment that launched it. A SCO must be designed so that it can be launched in a standalone web window or in a frame in an HTML frameset. A SCO is special, because, when launched for a learner in the learner's web browser, it will communicate information back to the LMS that launched it, often a remote server. This communication allows the LMS to track information pertaining to the learner's experience. A SCO represents the lowest level of granularity of a learning resource that an LMS should track. SCORM does not impose any particular constraints on the size of a SCO. A SCO can be a single web page or a collection of web pages (as long as the collection of pages can be considered a self-contained single unit).

Each SCO should be reusable and independent of its learning context. To achieve such reuse, a SCO should be "self-contained" and not reference or link to other SCOs.

Launched SCOs may be launched in a browser frame or a popup window. A SCO should not close the window they are launched in, unless it determines it "owns" the window.

Any "passive" asset can be converted to a SCO by declaring it as a SCO in the manifest file and ensuring it exhibits required SCO behaviors.

Required SCO run-time behaviors:

* Find the RTE API instance provided by the LMS
* Use the API instance to initialize communication with the LMS
* Use the API instance to terminate communication with the LMS

Recommended SCO behaviors:

* A SCO should be reusable in different learning contexts
* A SCO should be independent of visual constraints, such as window size
* A SCO should reliably transmit learner data so that it is not lost if closed unexpectedly
* A SCO should communicate its completion status
* A SCO should NOT launch new browser windows without closing them when done
* A SCO should NOT link to other files in the content package not listed as resource files of the SCO in the manifest

Restricted SCO behaviors:

* A SCO may NOT interact with the run-time environment in any way other than the provided run-time API
* A SCO may NOT attempt to change the size or appearance of the run-time environment it is launched in
* A SCO may NOT close the top-level browser window it is launched in unless it is the only thing in the window
1. Typical SCO Lifecycle
2. SCO is launched by a SCORM Run-Time Environment (RTE) (often an LMS)
3. SCO finds RTE provided API
4. SCO begins communication with the RTE API (via a call to Initialize())
5. Learner begins interaction with the SCO
6. SCO sends and retrieves data via the RTE API (via calls to Get/SetValue())
7. Learner ends interaction with the SCO

SCO ends communication with the RTE API (via a call to Terminate()).

**2.3.3 SCORM Sequencing and Navigation.**

A learner experiencing SCORM content does not necessarily exercise complete control over which learning object they experience and when. Content developers determine how much control is given to the learner based on organization and sequencing information defined in the content package's manifest. When accessing SCORM content, a learner will experience only one learning object at a time. SCORM Sequencing and Navigation define the ability of a learner to navigate from one learning object to another and the sequence in which learning objects may be experienced by a learner. In SCORM, a learning object (SCO) is not allowed to explicitly link to another. Rather, when a learner wishes to experience an activity they use navigation controls provided by the LMS (e.g. Start, Continue, and Quit) to issue navigation requests. The navigation requests are processed by the LMS to determine the sequence of learning activities. Most of the complexity burden associated with SCORM navigation and sequencing lays with the implementers of a SCORM run-time environment (RTE) or learning management systems (LMS). Even for content developers, though, a basic understanding of how SCORM content is delivered at run-time is necessary.

**2.3.3.1 SCORM Sequencing**

SCORM Sequencing defines the behaviors and data model used by the SCORM run-time environment to determine how a content package will be delivered as a learning experience. It defines the functionality that a SCORM-conformant LMS must implement to process sequencing information at run-time based on learner interaction with content objects.

SCORM Sequencing depends on the following concepts:

* An activity tree representation of learning activities
* The Sequencing Definition Model
* The Sequencing Behaviors

**Activities and Trees**

An Activity Tree is a run-time representation of the <organization> structure defined in a content package's manifest file. The <organization> element is the root of the activity tree and each of its <item> elements corresponds to a learning activity.

Each activity in an activity tree represents an instructionally relevant unit of learning. An activity is a "meaningful unit of instruction" that may consist of a launch able learning object (leaf) or be composed of other learning activities (cluster).

**2.3.3.2 SCORM Navigation**

SCORM Navigation defines how learning and system-initiated navigation events are triggered and processed, resulting in the identification of learning activity for delivery. Navigation is the process by which a learner and an LMS cooperate to identify navigation requests to realize a learning experience. For a learner to access a course or any of its activities, a navigation request must be issued . The result of each navigation request (ideally) is one of two things: an activity is delivered to the learner or the current activity is taken away. Note that only one activity can be experienced by the learner at a time.

How the LMS knows which activity to deliver in response to a navigation request is defined by the content package's activity tree and sequencing information. By default, a learner experiencing a content package will choose an activity from the tree to launch.

 **Navigation Events**

Navigation events can be triggered by a learner through user interface devices provided by the LMS or content or directly by SCOs. The LMS will determine the type and style of the user interface presented to a learner at run-time for navigation.

Navigation Events, their source, and description:

* **Start** - LMS; request to identify the first or "starting" activity of a tree, typically generated automatically by the LMS when the learning begins a new attempt
* **Resume All** - LMS; request to resume a previously suspended attempt on an activity tree
* **Continue** - LMS or SCO; request to identify the "next" logical learning activity (in relation to the current activity) available in the tree
* **Previous** - LMS or SCO; request to identify the "previous" logical learning activity (in relation to the current activity) in the tree
* **Choose** - LMS or SCO; request to "jump" directly to a specific learning activity in the tree
* **Abandon** - LMS or SCO; request to prematurely or abnormally terminate the current attempt on the current activity
* **Abandon All** - LMS or SCO; request to prematurely or abnormally terminate the current attempt on the root activity of the tree
* **Suspend All** - LMS or SCO; request to "pause" the current attempt on the root activity of the tree
* **Unqualified Exit** - LMS or SCO; indicates the attempt on the current activity has finished normally and not as the result of another navigation event
* **Exit All** - LMS or SCO; indicates the current attempt on the root activity of the tree has finished normally

The result of processing a navigation event will be one of the following:

* If the effect of the event is to end the current attempt on the activity tree, the LMS will process an Exit All request and return control to the LMS
* The request will be honored and the sequencing system will be invoked, resulting in one of the following:
	+ A learning activity is identified for delivery
	+ No learning activity is identified for delivery
	+ An exception occurs resulting in undefined LMS-handled behavior
* The request will not be honored and the LMS will take no action until another
	+ 1. Choosing an eLearning Platform

It’s true that some aspects of eLearning platform can be handled by simply putting files on internet file server and using a spread sheet or simple database to track who is taking the training. But most organization needs a more robust system to achieve their goals, so developing our custom system to meet those kinds of need can quickly become a huge and expensive development task.



So a commercial or open source system is often a most efficient and cost effective way to go. Commercial LMS are generally cost prohibited for small organizations. So these systems make more sense with large number typically thousands of users.

So when choosing a LMS first of all we need to understand few definitions like LMS, LCMS, CrMS or VLE. It can be be explained with the help of table given below.

|  |  |  |
| --- | --- | --- |
| **Category** | **Primary Function**  | **Used By** |
| **LMS (Learning Management System)** | A system primarily designed to manage & deliver asynchronous e-learning | Business & Government. |
| **LCMS (Learning & Content Management System)** | Content Authoring + delivery asynchronous learning.  | Business & Government. |
| **CrMS (Course Management System)** | Manage all aspects of live instructor led classroom training | Higher Education. |
| **VLE (Virtual Learning Environment)** | Virtual Business meetings & Collaboration. | Formal Education Environment. |

So we have identified that we have hybrid kind of requirement i.e. we need learning management system also we need some sophisticated tools for elearning content creation. We need VLE for organizing web conferencing, also we need integarion of LMS with some Content Management system (CMS) for better indexing, searching publishing and archiving different kinds of content .

We need LMS system for basically Patient Education as well as for specialized medical / surgical field. So we have identified our high level reqirement which are listed below

**High level requirement for Patient Education.**

* Need for the **content authoring** tool for transforming DICOM images into SCORM conformant eLearning content for online Diagnosis of patient.
* Patient profile management with complete Diagnosis report.
* Interaction with Patient using Web Conferencing, Wikis, IM and discussion forums.

**High level requirement for Neurosurgery Education.**

* Content Authoring tools for development of eLearning content related to Virtual

 Medical Simulations (VMS).

* Centralized Repository.
* Knowledge Centre.
* Administration.
* Course Management.
* Facility for Providing Webinars and Live exam.
* News Forum.
* Profile Management.
* Mobile Portability.
* Grading & Testing.

So depending upon our high level requirement we have constructed a system matrix for evaluating different kinds of open source Learning Management Systems.

So system matrix is shown below.



So currently we have installed moodle on the local server and evaluating different aspects of it. We will also evaluate different Learning Management system and will choose which will best satisfy our requirement.

We have also integrated moodle with Joomla which is Open source CMS because it is providing some of unique features such as.

* Centralized User Profile of Joomla and Moodle
* Single Sign on for Joomla and Moodle.
* Provide the Third party tools integration for both Joomla and Moodle.