Master’s Dissertation on

Educational Applications and Content Development for Engineering Students

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology

by

Subhasmita Mahalik
Roll No : 113050073

under the guidance of

Prof. D. B. Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Declaration

I declare that this written submission represents my ideas in my own words and where other’s ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Subhasmita Mahalik
(113050073)

Place: IIT Bombay, Mumbai
Date: 20th Oct, 2012
Acknowledgement

I would like to use this opportunity to express my deepest gratitude for my respected project guide Prof. D. B Phatak for his constant encouragement and invaluable guidance. I would like to thank him for spending his precious time towards this work. I would like to thank Nagesh Karmali for his guidance and the suggestions that he gave for my MTech Project. I am grateful to him for suggesting proper direction to work and motivation. I am thankful to all the time spent in the meetings and the discussions.
Abstract

Engineering education is the activity of teaching knowledge and principles related to the professional practice of engineering. It includes the initial education for becoming an engineer and any advanced education and specialization that follow. Technology education in primary and secondary schools often serves as the foundation for engineering education at the university level. Technology can refer to material objects of use to humanity, such as machines or hardware. Most distance learning programs include a computer-based training (CBT) system and communications tools to produce a virtual classroom. Because the Internet and World Wide Web are accessible from virtually all computer platforms, they serve as the foundation for many distance learning systems. But the learning objects created by one instructor should be reusable and compatible with Learning management systems. Therefore there should exist a certain number of standards to define various parts of learning objects. SCORM is a sort of “standards package” to achieve this.

SCORM specifies how learning content should be coded, how others can later "discover" that content, how it fits into a sequence of learning activities, and how its appearance through the delivery media can be customized for the individual learner. The SCORM standard makes sure that all e-learning content and LMSs can work with each other. Specifically, SCORM governs how online learning content and Learning Management Systems (LMSs) communicate with each other. SCORM does not speak to instructional design or any other pedagogical concern; it is purely a technical standard. SCORM is a really powerful tool for anyone involved in online training.
Contents

1 Introduction ........................ 1
  1.1 Traditional learning vs Active Learning ......................... 1
  1.2 Importance of e-learning standard .......................... 2
  1.3 Motivation ........................................ 2
  1.4 Problem Statement .................................... 3
  1.5 Techniques used for effective e-learning ..................... 3

2 Study of SCORM ......................... 5
  2.1 Introduction ........................................ 5
  2.2 Anatomy of a SCORM package .............................. 6
  2.3 API Wrapper ......................................... 9
  2.4 Understanding the SCORM API ............................. 10
  2.5 Data Transfer ....................................... 10
  2.6 API methods ......................................... 11
  2.7 CMI Data Model ...................................... 12
  2.8 Inter-SCO Data Storage ................................ 13
  2.9 Sequencing and Navigation ............................... 14
  2.10 Exiting SCOs and Courses ............................... 15
  2.11 ADL SCORM Overview ................................ 16

3 Creation and Deployment of SCO .......... 18

4 Benefits and Drawbacks of SCORM ....... 19
  4.1 Benefits ........................................... 19
  4.2 Drawbacks ......................................... 20

5 Conclusion and Future Work ............... 21
Chapter 1

Introduction

Distance education or distance learning is a field of education that focuses on teaching methods and technology with the aim of delivering teaching, often on an individual basis, to students who are not physically present in a traditional educational setting such as a classroom. It has been described as ”a process to create and provide access to learning when the source of information and the learners are separated by time and distance, or both.” The types of available technologies used in distance education are divided into two groups: synchronous learning and asynchronous learning though the expansion of the Internet blurs these distinctions.

Synchronous learning technology is a mode of delivery where all participants are ”present” at the same time. It resembles traditional classroom teaching methods despite the participants being located remotely. It requires a timetable to be organized. Web conferencing, videoconferencing, educational television, Instructional television are examples of synchronous technology, as are direct-broadcast satellite (DBS), internet radio, live streaming, telephone, and web-based VoIP.

The asynchronous learning mode of delivery is where participants access course materials on their own schedule and so is more flexible. Students are not required to be together at the same time. Mail correspondence, which is the oldest form of distance education, is an asynchronous delivery technology and others include message board forums, e-mail, video and audio recordings, print materials, voicemail and fax. In this section we describe some of distance education techniques and steps to develop virtual reality applications.

1.1 Traditional learning vs Active Learning

Learning can be either Active learning or Passive. Active Learning means getting students to participate actively in their own learning. Instead of asking them to sit quietly while we lecture, we are asking them to develop ideas, to practice solving problems, and to talk to and listen to their peers. But in passive learning, the learners simply sit and listen to lectures and don’t challenge themselves or the facts. The student was viewed as an empty vessel, anxiously waiting to be filled with the vast knowledge of the learned instructor. The instructor lectured, the student took notes and then regurgitated the knowledge on an exam. Granted there are some advantages to passive learning or people wouldn’t keep doing it.
For example, you can present a lot of information in a short time. Concepts and content can be prepared in an organized and consistent manner. Lastly, older adult learners are used to this method and may have less anxiety with a passive style. The major disadvantages include: tedious, boring learning and the lack of opportunity to consistently stimulate higher-level cognitive skills. Active learning is much different [1]. It is practice: trying something, seeing how well or poorly it works, reflecting on how to do it differently, then trying it again and seeing if it works better. Active learning works because the learners play an active role in the learning process and their brains are stimulated in different ways. Studies show that people retain 20% of what they hear; 40% of what they hear and see; 60% of what they hear, see, and do; and 80% of what they hear, see, do and teach. Knowing this, it makes sense to engage these highly inquisitive and playful minds in different ways.

1.2 Importance of e-learning standard

Why e-learning standards are important and why you should care whether the content that you create is standards-compliant? The primary goal is to make course content and learning tools compatible among various e-learning delivery systems. The advantage is that you can create portable content that can be used in any e-learning delivery system without the need to modify the content. If the content and the e-learning delivery system both comply with the same standard, then the learning system will interpret and deliver your standards-compliant content [1].

1.3 Motivation

There were lot more courses and lot more course providers. The big problem was that the course providers could not share the courses. All the courses were different. They did not communicate. Then came the brilliant idea that there should be standard. Then they invented SCORM.

Now the question is What is SCORM? Before we talk about SCORM lets think a moment about communication. When we communicate, we begin with many assumption without even realising it. Imagine you are a teacher in the classroom. One of the most important part of your job is to communicate. But what if the students did not speak english and you didn’t speak their language. Nobody would understand you. Inorder to understand each other we must have basic assumptions on how to communicate. And thats what SCORM is: A basic set of assumptions that allows courses and LMS to communicate. Now getting back to our basic question: What is SCORM? SCORM is the best and popular standard for building online content. If you want to build or buy an online course to share with learners it got to be SCORM course. The word SCORM is acronym. It stands for sharable content object reference model. When a course is sharable you have lot more freedom on how to use that course in the organisation. You can take it from one system to another without changing it. Lessons in SCORM are called objects. Its like sections in a text book. You can have any number of sections. You can set moves for those lessons. For example you can make sure that the learner
have to understand one lesson before going to another. To sum up the SCORM course is a collection of lessons, packaged with instructions that are understood by online learning management systems called LMSs. A SCORM course is like a good teacher. They both have a lesson plan and track the progress of the learners. They both know who you are and call you by name. They both track your progress and record when you study and when you leave. Both the teacher and the SCORM course record your score.

1.4 Problem Statement

Varieties of courses are offered in an educational institute by different course providers. Since each course provider uses different platform and different tools for creating their course content, it becomes difficult to share contents of one course with other course providers. To solve this problem, a single standard format should be followed. This gave rise to a model known as SCORM that contains a set of technical standards for e-learning software products. So the aim of this project is:

- To understand the SCORM standard
- To develop interoperable and reusable objects called SCO using Reload Editor
- To test how the SCORM standard is used in developing courses for engineering students
- To add the features of navigation and sequencing in the SCORM object
- To test the SCORM object produced by using Reload Player

1.5 Techniques used for effective e-learning

1. Offer learners more than clicking the next button. Rapid authoring tools like Articulates Engage provide cool, easy-to-use templates to help structure content in a highly engaging presentation style [2].

2. Use branching. This is a technique where the learner makes a choice and different decisions take them down different paths. Simulating a real-life experience through decision-making increases knowledge retention.

3. Don’t just test at the end. Pepper questions in the form of Knowledge Checks throughout the course. Forcing learners to think, rather than just read, is much more active.

4. Use audio. The more senses you can get firing the better. Don’t waste time on narration that simply reads the onscreen text though or they’ll tune you out in no time.

5. Make it fun. If 50% of education is supposed to be entertainment, it’s hard for people to learn when they’re snoring.
Things to consider while developing e-learning courses:

- **Sequencing** is defined as logical and appropriate organization of critical concepts in order to best facilitate learning of new materials. Subject matter presentation approaches, moving from the general to the particular or from the particular to the general, have been debated in distance education [1]

- **Modular Course Structure** for Distance Education Materials: division of the subject matter into parts or segments - course units. It is necessary to make it easier for the student to fit study into the normal, very active, adult lifestyle

- **Learner Control**: flexible and adaptable in that learners can study anywhere and at anytime.

Distance education pre-produced course materials require structure that will keep students focused and interested.
Chapter 2

Study of SCORM

2.1 Introduction

Evolution of electronics, computer systems and information technologies, together with the worldwide accessibility to the Internet have made available an incredible set of applications. The issue of component interoperability of heterogeneous components is relevant to promote reuse of services and develop e-learning systems. E-learning is grouped into two core components: The Learning Management System (LMS) and The Learning Content Management System (LCMS). LMS manages all the activities surrounding learning deployed via Web, such as user authentication, course deployment, and management of collaborative synchronous and asynchronous environments.[1]

LMS also interacts with the run-time environment and traces the student learning process while the e-learning course is offered.

The server component implements the RTE by J2EE technology, while the client component needs a Web browser supporting ECMA script and Java applets to enable the communication between didactic contents and an e-learning platform.

The Middleware Machine node is composed of five software components. The Front End component presents the appropriate features and contents to the students and teacher using the components on the Middleware Machine node. In particular, Front End uses Content Sequencer to show the suitable knowledge content to the students, and uses Tracer component to trace the learning process. The Content Deliverer component allows the instructional designer to deploy single e-learning activities or whole courses. The Collaborative Environment Manager component is used to manage the cooperation among students and teachers. The RTE Machine node contains the software components to manage and deploy linear and adaptive didactic contents as well as to trace learning processes. The Synchronous Collaborative Manager component on the Service Machine Node provides means to handle and use synchronous and asynchronous collaborative environments.

The SCORM specification, developed by Advanced Distributed Learning, is a set of rules that learning management systems (LMS) and learning content follow in order to be compatible with each other. This theoretically allows the content to be loaded into, launched and tracked by any learning management system using a common rule set.[5]
2.2 Anatomy of a SCORM package

- **Asset**: Assets are electronic representations of media, text, images, sounds, HTML pages, assessment objects, and other pieces of data. They do not communicate with the LMS. Assets will likely be your most reusable items; they can be redeployed, rearranged, repurposed, or reused in many different contexts and applications.

- **SCO**: E-Learning content is delivered as a single unit called a Sharable Content Object or SCO. SCOs are independent, self-contained, transportable packages that represent the lowest level of granularity that is tracked by a learning management system. In other words, SCOs are mini-applications that when launched from an LMS report, among other things, one score and pass/fail status. It should remain relatively small such that they can be easily reused. SCOs are web-based and script-enabled consisting of HTML, Javascript and any other client-side technology (images, Macromedia Flash, etc). And since the spec indicates that SCOs shall be transportable, it could be inferred that they should also be platform independent.

- **Aggregation**: An aggregation is a collection of related activities. An aggregation may contain SCOs or other aggregations. In this Guide, an aggregation is defined as a parent and its children. An aggregation is not a physical file; it is a structure within a SCORM manifest where sequencing rules are applied to a collection of related SCOs or aggregations.

- **Organization**: The organization is the part of a content package where SCOs are ordered into a tree structure and sequencing behaviors are assigned to them. The organization outlines the entire structure you have created for
the content that you intend to deliver as a single content package. Each organization is a top-level aggregation, also referred to as the root aggregation in this document.

- **Curriculum or Course:** While a curriculum or course is outside the scope of SCORM, SCORM-conformant content can be part of a curriculum or course that is managed by your LMS. A curriculum consisting of multiple independent components. It consists of some SCORM organizations and a combination of other learning experiences (such as, collaboration sessions, labs, lectures). A curriculum typically includes courses, lessons, and assessments using a variety of delivery media and instructional strategies.

- **Content Package:** The content package may contain more than one SCO and a single manifest file which describes the contents of the package. The LMS will use the manifest file to properly find and import each of the SCOs. Some authoring tools will create the entire content package after you load your SCOs and assets into the tool. The ADL version of the RELOAD is an example of such a tool. RELOAD provides a graphical interface for creating a content package and managing sequencing and other values contained in the manifest file.

  For a SCO to be uploaded to the LMS, it must be contained in a Content Package, a single compressed file conforming to the Process Interchange Format (PIF) specification. In most cases, this is simply a .ZIP file and may contain multiple SCOs.

  The Content Package contain a Manifest File, an XML file containing information about the included SCOs and their organization. By reading the manifest
file, the LMS can gather information about the SCOs that are contained in the package and will be able to launch them when appropriate.

Components of Content Package

A SCORM content package contains two principal parts:

1. The XML manifest file that describes
   - All of the SCOs or assets you want to include in the package
   - A representation of the content structure diagram created (called the organization)
   - The sequencing rules
   - Metadata for the SCOs, aggregations, and the package itself

2. All of the actual SCO and asset files for the content package

• **Manifest:** An Extensible Markup Language (XML) file called a manifest organizes the content package. The manifest is a detailed set of instructions, structured in a manner specified by SCORM, that organizes your content package and tells the LMS when, how, and what content to deliver to your learners.[8]

The manifest file is always named imsmanifest.xml and it always appears at the top level of a content package (zip file), regardless of the structure of the rest of the package.

The basic structure of a manifest file contains:

**Metadata node:** The metadata section is where additional informative data about the course is placed. At its simplest and most common form, it only contains the schema and schemaVersion elements:

```
< metadata >
    < schema > ADLSCORM < /schema >
    < schemaversion > 20044thEdition < /schemaversion >
< /metadata >
```

**Organization node:** The organization consists of multiple activities (SCO or aggregation) represented by item elements. This structured representation of the content is typically called the "Activity Tree." Inside the organization and item is where all the sequencing is defined.

```
< organizationsdefault = "ORG – SAMPLE" >
    < organizationidentifier = "ORG – SAMPLE"
        adlseq:objectivesGlobalToSystem = "false" >
        < itemidentifier = "AGGREGATION1" >
            < title > SampleAggregation < /title >
            < itemidentifier = "SCO1" identifierref = "RES1" >
                < title > SampleSCO < /title >
            < /item >
            < itemidentifier = "SCO2" identifierref = "RES2" >
```

8
Resource node: SCORM content typically consists of web-delivered assets. These assets may be HTML, images, Flash objects, audio, video, etc. All of these assets are listed in the resources section of the manifest file. A resource is a grouping of related assets. There are two types of resources: SCO resources and asset resources. If a set of assets is shared by multiple SCOs, then an asset resource listing may be created to remove repetition of assets in the manifest file. The asset resource is referenced using the dependency element.

2.3 API Wrapper

It is a file will locate the SCORM API instance and contains all the functions required for the content to communicate with the LMS. These functions are standard JavaScript functions and may be used just like any other JavaScript functions in a web page. To be conformant, a SCO must make, at a minimum, two calls. The
first, doInitialize(), must be called to initiate communication between the LMS and the SCO. The second, doTerminate(), must be called at some point before the SCO exits. [8]

### 2.4 Understanding the SCORM API

Once its been launched, a SCO and the LMS need a communication channel through which to pass data. This is done through an object known as an API (Application Programming Interface) that is the conduit for all of the SCO to LMS and LMS to SCO communication. APIs can be APIs developed using Java applets, ActiveX controls and even pure JavaScript.

The SCO will be opened by the LMS in either a new browser window or framed within a page in the LMS browser window. Before communication can occur, the SCO must find the API and make contact.

The SCORM specification indicates locations where the SCO should look for the API, requires that the LMS expose the API in one of those locations and that it be a Document Object Model (DOM) object named API\_1484\_11 or API depending on the SCORM version you are following. Since the SCO is launched in either a new browser window or a frame within the LMS window, it will use a very simple process such as: look in my parent window (if framed) or in the window that opened me (if a new window) and see if there is an object named API\_1484\_11 (or API). If the API is not found, the SCO can expand its search by looking in the parent of the window that opened it, the parent of its parent, and so on according to the spec.

Once the SCO finds the API, it can invoke methods of the API to send data to and receive data from the LMS. If the API is not found, the SCO should alert the user that the connection to the LMS failed and no communication will occur. The ADL has made an API wrapper publicly available for use when developing SCOs. This wrapper is a file that you can include with your SCO that contains pre-written javascript functions for finding and accessing the API as well as for sending and receiving data.

### 2.5 Data Transfer

Once the SCO has found the API, they must both speak the same language if any communication is to occur. The SCORM specification has defined a small set of methods that must exist in the API and be available for the SCO to use.

The methods are accessed by the SCO via JavaScript code with the syntax of `objectname.methodname(argument(s))` where objectname references the API itself, methodname is the method being used (the API methods are explained below) and argument(s) are the data passed to the method. In all cases, the SCO initiates interactions and data transfer by invoking these methods. [8]
2.6 API methods

- **Initialize** Initializes communication with the LMS. No other API methods should be called by the SCO until Initialize has been successfully called. When completed, the resulting returnValue will contain true if the method was successful, false if it was not.

- **Terminate** Terminates communication with the LMS. No other API methods should be called by the SCO after Terminate has been successfully called. When completed, the resulting returnValue will contain true if the method was successful, false if it was not.

- **Commit** Saves the data that has been sent to the LMS via SetValue calls. If a SCO exits without invoking Commit, none of the learners data is saved to the LMS. Commit is implicitly invoked by the API when Terminate is called. When completed, the resulting returnValue will contain true if the method was successful, false if it was not.

- **GetValue** Retrieves data from the LMS for use in the SCO. The SCO must pass the data element that it is requesting as an argument. When completed, the resulting returnValue will contain the score that is retrieved from the LMS.

- **SetValue** Passes data from the SCO to the LMS. The data is retained and may be retrieved during the user session, but is not saved to the LMS until Commit is invoked. The SCO must indicate the data element and its value that is to be saved as arguments. When completed, the resulting returnValue will contain true if the method was successful, false if it was not.

- **GetDiagnostic** Exists for LMS specific use. Returns a diagnostic text description based on the parameter that is passed as an argument. When completed,
the resulting return_value will contain the text of the diagnostic information.

Using the GetValue and SetValue methods, the SCO is able to send and retrieve all the necessary data for effective tracking to and from the LMS. Some commonly used data elements include learner_id, learner_name, score, completion_status and suspend data. The example that follows will demonstrate proper usage of a few of these elements.

So what we find when looking at the code of a SCO is some process that is executed at launch, usually found in the onLoad event of the body tag, that finds and initializes the API. In some cases the SCO will then request and load basic information using GetValue, such as the learners name and id. Additionally, the SCO may attempt to load previous session data such as score, progress status or learner responses that may be necessary to continue the learners interaction with the SCO. From that point, SCO interactivity and functionality such as navigation and quizzing will likely operate independently of any SCORM-related functions. At certain points though, result data is passed to the LMS (using SetValue and possibly Commit) and upon completion of the SCO, perhaps within the functionality of an Exit button, the Terminate method is called.

### 2.7 CMI Data Model

This section explains storing and retrieving data about learner performance from and to the Learning Management System (LMS). Understanding the types of data that can be communicated via the SCORM data model enables you to discuss with your instructional system designer (ISD) what information you can retrieve from or store in the LMS.

The data model is used whenever you need to store or retrieve data related to the learners session in the LMS. For example, you might want to retrieve the following information from the LMS:

- Score
- Total time spent in a SCO
- Time spent in a single session of a SCO
- Completion status
- Responses to assessment items
- Interactions within a SCO
- Pass/fail status

The cmi data model is accessed through JavaScript calls in the content. In the API wrapper files we are using in this document, there are two calls that are used when working with these elements: doSetValue and doGetValue. Values of the cmi data model are simple strings and numbers.
2.8 Inter-SCO Data Storage

Prior to SCORM 2004 4th Edition, SCOs had no visibility or access into information tracked by other SCOs or even different attempts on a given SCO. With 4th Edition, ADL has provided the adl.data data model element to allow for the sharing of sets of data across SCOs through a collection of data stores. [9]

Data stores are associated with a SCO using the `adlcp:data` extension element in the Manifest File. The data stored in adl.store may be shared across content packages as well. An attribute adlseq:sharedDataGlobalToSystem in the `organization` element of the manifest controls if this data is accessible outside the scope of the content package. The default value is true, so be sure to set the value to false if you wish to restrict access.

bfStatus and Scoring

Both a SCO and an aggregation have a primary objective. This primary objective is an object holding status values related to the progress and success of the learner’s interaction with the activity. The following information is stored in the primary objective:

- Success_status
- Completion_status
- Score (scaled, raw, min, max)
- Progress_measure

When you need to track a numeric score in a SCO, the score object of the primary objective is used. The score is calculated by the content. For example, to set the score to 85%, the following JavaScript code is used in the content:

```javascript
doSetValue("cmi.score.scaled", .85);
```

The other elements of cmi.score are used to provide additional context only. These
include cmi.score.min, cmi.score.max, cmi.score.raw. The min and max values may be used to define a range and the raw element should be a number within the range of min and max.

A score in and of itself does not have any impact on the course. It should be combined with the Success Status and Sequencing. The Success Status represents if an activity is passed or failed. This can be set directly, using the data model API: doSetValue("cmi.success_status","passed");

2.9 Sequencing and Navigation

- **Sequencing:** Developers of SCORM conformant runtime environments must implement sequencing. But developers of SCORM conformant content do not have to implement sequencing. Sequencing is optional. If the content designer is happy to allow learners free rein over the content, no sequencing needs to be specified [4].

  How do you specify sequencing? Some designers believe that unless their content is used according to a particular activity sequence it will not work. Therefore, SCORM 2004 allows you to specify sequencing behavior in a content package. But if the learner already knows something, there is no need to go through a tutorial on the subject. A simple way to support different learning styles, for example, is to enable guided flow without disabling free choice in navigation between the activities. For field-dependent learners, the defined flow provides the guidance they need. For field-independent learners, the ability to choose lets them learn through discovery.

- **Navigation:** SCORM the control resides firmly with the runtime environment, and the SCOs are just resources used in the fulfillment of learning activities. However, sometimes it is useful to be able to embed a navigation control in the SCO itself. For example, a content developer might want to provide a package in which a "Continue" button goes from event to event inside a SCO. When navigation inside the SCO reaches the end of the SCO.

  This feature allows the SCO to display or enable a "Continue" button only if the runtime environment reports that continuing to another SCO will be allowed. It also allows the SCO to communicate to the runtime environment a navigation event that should take place when the SCO calls Terminate to end its communication session. When Terminates is called, the runtime environment commits any remaining data set by the SCO, and then it updates the status of the activity tree, and then is looks at a possible pending navigation signal from the SCO. It is important to note that this signal is only a request. The runtime environment, based on its interpretation of the sequencing rules and on the status of the activity tree, may deny the request.
2.10 Exiting SCOs and Courses

The concept of exiting learning content is an important one. Everything that gets launched must eventually exit. There are many variations of exiting a course, that will appear to be similar from the learner’s perspective, but the results are quite unique.

There are multiple applications of the word ”exit,” so to make it clear, consider the following use cases:

1. The learner has finished a SCO and is ready for the next activity.
2. The learner needs to ”pause” in the course and return later.
3. The learner has completed all the activities in the course.

Each of these represents a way to ”exit” an activity in the course. When a SCO terminates, it is either with a normal or suspend exit state. If a SCO is exited normally, it means that this attempt on the SCO is completed and any re-entry into the SCO will initiate a new separate attempt. A suspended SCO retains state data from the prior session, and upon resumption of the content, the learner may be placed where he/she left off.

To exit normally, which typically indicates some finality (not needing to resume), the following JavaScript code in the SCO may be executed:

```javascript
doSetValue("cmi.exit", "normal");
doTerminate();
```

To exit in a suspend state (the ability to resume later):

```javascript
doSetValue("cmi.exit", "suspend");
doTerminate();
```

If the SCO is exited, as described above, the course session is still active. The learner will have to select a new activity or sequencing will present a new activity to him/her. The course session is not necessarily exited if the SCO exits. A course is usually exited when the learner needs to take an extended break from the course or when the course is completed.

A course is exited through the use of exitAll or suspendAll navigation requests. If a course receives either of these navigation requests, this will cause the course to close and the LMS will regain control.

There are four places that affect when and how content is exited:

1. The LMS Navigation UI
2. The LMS table of contents
3. Programming JavaScript calls inside the SCO
4. Sequencing post-condition rules
2.11 ADL SCORM Overview

Bookmarking
Bookmarking is the common term used for allowing the learner to take a break and then return to where they left off in the content. SCORM 2004 does not use the term Bookmarking, but provides multiple ways to accomplish this. This can be done either by Save the State Or Suspend the Content[4].

Two ways to save content state that can be accessed when the learner resumes. These elements are simply storage areas for text that may be used for bookmarking. No bookmarking occurs automatically.

cmi.location (hold 1000 characters) and cmi.suspend.data (hold 64000 characters) are the two options for bookmarking using Save the State.

Prerequisites
A very common design pattern is that of a prerequisite. This is where the availability of one activity is dependent on some external condition, typically the completion or satisfaction of a prior activity.

The steps to implement this are listed below and are described in detail in this section:

1. Create an aggregation (cluster) for the lessons
2. Add Rollup Rules to aggregation
3. Map aggregation’s primary objective to a shared global
4. Create Pre-Condition rule on the assessment

Assessments
An assessment, otherwise known as a test, is a common type of content. It is implemented as a normal SCO. However, SCORM provides some features that can assist in the tracking and reporting of the learner’s experience in the assessment. The tracking requirements for an assessment will vary based on the design. In some cases, you may only need to mark the assessment as passed or failed. In other cases, you may also need to calculate a numeric score and store this in the LMS as well. In yet other instances, you may have to capture and store in the LMS detailed information about each assessed item.

Most assessments are graded and given a result of passed or failed. This value is calculated by the programming logic in the course and the resulting value is stored in the LMS as follows:

// cmi.success.status is either passed or failed  
doSetValue("cmi.success.status","passed");

Many assessments are also given a numeric score. The calculated numeric score is stored in the LMS using the cmi.score object:

// represents a score of 75%  
doSetValue("cmi.score.scaled","0.75");

The cmi.interactions data model element is an array type object consisting of many interactions. A single interaction describes a tested item. An interaction is represented by dot notation similar to other cmi data model elements. The most common
elements used in the cmi.interaction object are: id, description, type, timestamp, correct_responses, learner_responses, result, latency. The interactions are a collection of information stored in arrays. The cmi data-model uses a zero-based index position (n) in the dot-notation to separate the data.
Chapter 3

Creation and Deployment of SCO

SCORM is just a standard and a SCORM object (or package) is just a collection of information. Creating one is relatively easy. Here follows the steps to create a simple SCO [7]:

1. Create a folder and two sub-folders. One of the subfolder will contain resources and the other will contain SCORM object.

2. Start the Reload editor.

3. Go to "File," select "New" and choose "ADL Scorm" to create a new SCORM object. The "Select Folder for New Content Package" window will appear.

4. Choose the resource folder as content package folder and click OK.

5. Right-click on the "Organizations" item in the main window and select "Add Organization" to add a default organization to your SCORM object. Right-click on the new organization, named by default "Organization," and select "Rename." Change its name to "Main," the conventional name for default organizations.

6. Right-click on "ScormPack," select "Import Resources" and browse to your resource folder. All the files present in this folder will be imported to your SCORM object.

7. Drag the resource files from the resource folder to the "Main" organization to add them to your object.

8. Right-click on "MANIFEST-..." and select "Add Metadata" to add a metadata container. This will create a "Metadata" item. Right-click it, go to "Add Schema" and select "Schema Version."

9. Go to "File" and click "Save" to save your SCORM object. Go to "File" and select "ZIP content package" to start creating a content package from your object. Give a name to your new package and select a location where to save it.
10. Test your new SCORM object by using the Reload Editor’s "Reload Scorm Player" tool.

Once the SCO is created it can be deployed by uploading the zip file in LMS. various available LMSs are: Moodle, Atutor, Meridian, Docebo etc. In case of moodle the following steps are followed to upload a file:

1. Click on "Turn editing on" after logging into moodle.

2. Click on "Add an Activity or Resources" as is done while uploading other resources. It opens a pop up window.

3. Select "SCORM package" in the pop window that was opened in the previous step.

4. Click "Browse" and upload your ZIP file.

5. Choose the various options according to the requirement.

6. Once the SCORM object is uploaded it can now be accessed by the students by simply clicking on the activity. It opens the object in the browser.

Otherwise, one can test the ZIPped SCORM object by using https://cloud.scorm.com/ link. It provides the LMS interface for launching and testing the SCOs.
Chapter 4

Benefits and Drawbacks of SCORM

4.1 Benefits

From a business perspective, standards are beneficial because they are essential to the growth and expansion of any technology-based industry. Whether it’s 802.11 for wireless networking, HTML for the Web, or standardized track gauges for railroad transportation, standards foster efficiencies and synergies that enable markets to grow. These benefits certainly apply to e-learning standards, allowing organizations that adopt SCORM to create efficiencies, lower costs, reduce risk, and increase overall learning effectiveness and return on investment (ROI).

Before SCORM, integrating content with a delivery platform for e-Learning or training used to take days, weeks or sometimes years, unless the content was built specifically for that platform. Often the costs of modifying the content or building special adapters, along with the time to deployment, were simply prohibitive.

1. Reuse content for faster development
   Developing content once, then reusing it for multiple audiences and contexts, reduces development time

2. Share content between systems
   Moving to SCORM makes integration easier between existing and future systems, protecting your infrastructure investments and lowering your cost of ownership. SCORM learning content can be integrated with, and delivered on, past and future SCORM-compliant systems

3. Reduce cost of content maintenance
   By enabling your organization to maintain content in-house using any tool you choose regardless of system or content vendor, SCORM lowers your overall cost of content maintenance

4. Maximize technology investments
   SCORM content can be launched, operated, and tracked by any SCORM-compliant content delivery system, whether content was developed in-house or by a third party, enabling you to get the most mileage from your technology investments

5. Avoid proprietary authoring tools
   The same tools that your development team is using to create Web content can be used to develop SCORM-compliant content, eliminating the need to use or develop proprietary tools
6. Train developers faster: As SCORM adoption continues to grow, the talent pool of content developers experienced in SCORM techniques and technologies grows too. A wide selection of SCORM training materials is available to get less experienced developers up-to-speed fast.

7. Leverage best practices: Take advantage of the collective knowledge and expertise of the growing SCORM community that is continually evolving and enhancing SCORM functionality, tools, methodologies, and best practices with proven results.

### 4.2 Drawbacks

1. The standard is completely based on a Web environment (mostly browsers).

2. The standard puts a great emphasis in the client-side of the learning process.

3. The security of the API can only be guaranteed in a very limited way since it must be accessed from JavaScript and consequently can also be manipulated externally.

4. Inherently insecure.
Chapter 5

Conclusion and Future Work

ADL SCORM constitutes an important first step towards liberating learning content objects from local implementations. It represents currently a leading effort towards the reusability and interoperability of learning resources. Also SCORMs good compatibility with most of popular learning resource specifications ensures its widespread acceptance and bright application prospect in the near future. The tools and techniques available for SCORM packaging are, in themselves, relatively easy to use and friendly enough to be tried by most teachers with just a little training. Adopting or adapting a useful, effective metadata set requires institutional effort and involvement. Learning materials preparation is hard work; creating sharable learning objects requires some additional effort, but reusability makes the task worth.

My future work will focus on exploring more advanced mechanism to control browser-based sequencing and navigating between learning resources based on the SCORM CP Application Profile, and adopting more advanced Data Model in the SCORM RTE to enhance interactive and adaptive capability. Also I plan to develop an engineering educational application to be deployed in android device.
Bibliography


