Converting to SCORM: Lessons Learned

Susan Marshall, Damon Regan
Joint ADL Co-Lab
Orlando, FL
susan.marshall1@us.army.mil, dregan@alionscience.com

Patricia Mulligan
AFAMS/ESS
Orlando, FL
patricia.mulligan@afams.af.mil

ABSTRACT

In this paper, the authors describe the process of converting the Air Force Modeling and Simulation Introductory Course (AFMSIC) to conform to the Sharable Content Object Reference Model (SCORM). SCORM is a collection and harmonization of specifications and standards that define the interrelationship of content objects, data models, and protocols such that objects are sharable across systems that conform to the same model.

The Air Force Agency for Modeling and Simulation (AFAMS) collaborated with the Joint ADL Co-Lab to demonstrate the process of converting web-based courseware to conform to SCORM, while collecting lessons learned, resulting in the production of a SCORM 2004 conformant course.

The objective of this paper is to explain this conversion process, while sharing lessons learned with the education and training communities implementing SCORM conformant courseware, to align with DoD distributed learning policy. The process involved (1) reviewing organizational concerns; (2) ensuring content remained instructionally sound; (3) defining Sharable Content Objects (SCOs); (4) debating the pros and cons of converting to SCORM 1.2 or 2004; (5) adjusting the design and development to include the Sequencing and Navigation specifications of SCORM 2004; (6) using practical tools; and (7) testing the courseware in the SCORM Conformance Test Suite.

ABOUT THE AUTHORS

Susan Marshall is a Principal Investigator for the Joint ADL Co-Lab. She has over 20 years of experience as a project engineer for the Navy and Army. Her areas of concentration at the Co-Lab include: Learning Content/Simulation Integration, and SCORM 2004 Applications. Mrs. Marshall has a BS in Electrical Engineering from Marquette University, and MS in Industrial Engineering in Simulation from the University of Central Florida.

Damon Regan is a Software Engineer for the Joint ADL Co-Lab. He has played active roles in providing SCORM acquisition and technical guidance. Before joining the Joint ADL Co-Lab, Mr. Regan operated air defense simulations in distributed training exercises at U.S. Army training centers. His primary interest is in distributed systems to support diverse learning experiences. Mr. Regan has a BS in Computer Science from the University of Central Florida and is an MBA candidate at the Crummer Graduate School of Business at Rollins College.

Patricia A. Mulligan is the Instructional Systems Designer for Engineering Systems Solutions (ESS). She is responsible for supporting the Chief of the Force Development Division at the Air Force Agency for Modeling and Simulation (AFAMS) in building the Air Force Modeling and Simulation (M&S) community. Ms. Mulligan assists in the management of the Air Force M&S Professional Development Program, contributes course management and content development support for the Advanced Distributed Learning (ADL) introductory course in M&S, provides analytical support to Knowledge Management applications, provides support to the AF M&S Workforce Development Study and leads the Air Force Modeling and Simulation Education and Training Working Group. She received her Bachelor of Arts degree in Speech and Hearing Science at Plattsburgh State University, Plattsburgh, NY and her Master of Arts degree in Instructional Technology – Instructional Systems Design at the University of Central Florida, Orlando, FL.

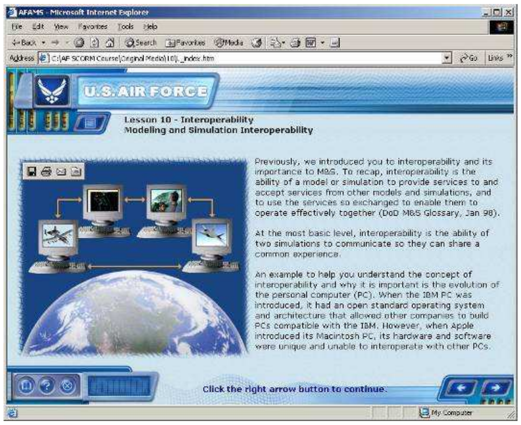
Converting to SCORM: Lessons Learned

Susan Marshall, Damon Regan
Joint ADL Co-Lab
Orlando, FL
susan.marshall1@us.army.mil, dregan@alionscience.com

Patricia Mulligan
AFAMS/ESS
Orlando, FL
patricia.mulligan@afams.af.mil

BACKGROUND

Modeling and Simulation (M&S) has become the key tool in how the Air Force (AF) prepares and trains its workforce to build readiness and save valuable resources. It allows the Air Force to depict events or battlespaces even if large forces are required, when the timeframe is well into the future, and where we cannot physically go regularly, such as space. With shrinking budgets hitting the Air Force, educating the M&S workforce has become a challenging task. The Air Force Agency for Modeling and Simulation (AFAMS) explored innovative approaches to educating the Air Force workforce on M&S using available training opportunities and identifying ways to fill these training shortfalls using cost-effective methods, such as advanced distributed learning (ADL).


AFAMS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links
U.S. AIR FORCE
My Computer

C:\AF SCORM Course\Original Media\10_index.htm

Lesson 10 - Interoperability
Modeling and Simulation Interoperability

Previously, we introduced you to interoperability and its importance to M&S. To recap, interoperability is the ability of a model or simulation to provide services to and accept services from other models and simulations, and to use the services so exchanged to enable them to operate effectively together (DoD M&S Glossary, Jan 98).

At the most basic level, interoperability is the ability of two simulations to communicate so they can share a common experience.

An example to help you understand the concept of interoperability and why it is important is the evolution of the personal computer (PC). When the IBM PC was introduced, it had an open standard operating system and architecture that allowed other companies to build PCs compatible with the IBM. However, when Apple introduced its Macintosh PC, its hardware and software were unique and unable to interoperate with other PCs.

Click the right arrow button to continue.

Figure 1. Original Course

The design and development of the Air Force Modeling and Simulation Introductory Course (AFMIC) (see Figure 1) was created to be delivered over the web. The course has been operational since August 2003. One last challenge remained. The Department of Defense (DoD) created the Sharable Content Object Reference Model (SCORM), which facilitates courseware interoperability. SCORM aims to foster creation of reusable learning content as "instructional objects" within a common technical framework for computer and Web-based learning. SCORM describes that technical framework by providing a harmonized set of guidelines, specifications and standards. Borrowing from work of other specification and standards bodies, ADL developed a model for creating and deploying e-Learning" (http://www.adlnet.org). AFAMS teamed up with the Joint ADL Co-lab in Orlando, FL to convert a non-SCORM AF course to conform to the SCORM 2004 reference model. The following is a description of our journey and our lessons learned.

INTRODUCTION

The SCORM 2004 set of specifications was published last year. Since then, there has been a lot of interest in how best to implement it, especially in light of DoD Instruction 1322.HH, which is expected to be signed in 2005. This Instruction will mandate the use of SCORM for new content development for the DoD.

The AFMIC course includes twelve modules plus pre-test and post-test. This was a collaborative effort between personnel from the Joint ADL Co-Lab and AFAMS. The main goals of our effort were to:

- Be sure the course would run in the Air Force's chosen LMS after the course was deployed
- Be reusable to the other Services
- Capture the lessons learned and document the process that we followed

ORGANIZATIONAL CONCERNS

Our initial plan was to convert the AFMIC to SCORM 1.2, because SCORM 2004 was not yet released. We originally planned to make each lesson a Sharable Content Object (SCO); however, we changed our strategy to make each learning objective a SCO. This decision was based on suggestions described in the Learning Systems Architecture Lab's SCORM Best Practices Guide for Content Developers and guidance from Air Education and

Training Command (AETC). What we discovered was that making the learning objectives as SCOs would not be a straight-forward process due to how the lessons were technically constructed. The course was built using framesets shown in figure 2 below.

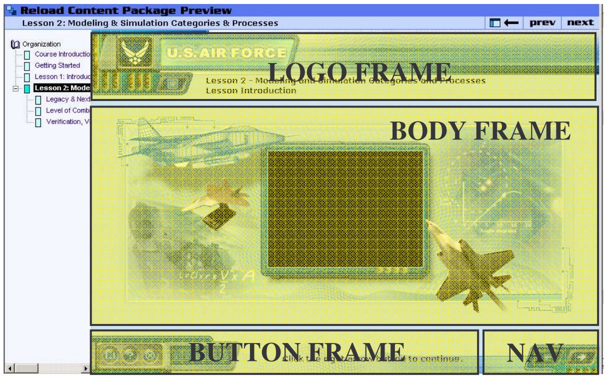


Figure 2. Frameset

Specifically, each lesson created the frameset; therefore, if a SCO (defined as a learning objective) was launched, it would not have the lesson's frameset, so the learner would only see the body frame shown in figure 2.

Leadership at AFAMS reviewed the options of developing SCOs at the learning objective level and concluded that they did not want the course to look or function differently than initially intended. This led us to review the new capabilities offered by SCORM 2004. The updated SCORM reference model offered new navigation capabilities that would allow us to implement SCOs as learning objectives without changing the look or function of the course.

INSTRUCTIONAL DESIGN

The team decided to record the effects on the instructional design of SCOs and converting courseware to SCORM 2004.

Defining Sharable Content Objects (SCOs)

Defining SCO's at the learning objective level required us to make some content modifications. The course was structured to introduce basic concepts first, and gradually build upon these concepts as the learner progressed through the course. For example, in lesson 3 we discuss the term interoperability in modeling and simulation. After providing a brief definition and some examples we reference that the learner will be learning more about the concept in a later lesson. The Instructional Designer had to review the content and make modifications to adapt to the SCORM reference model. This was one of the challenges of converting courseware to SCORM. How do you create learning objectives to stand alone without breaking the flow of instruction? Be creative. We decided that we can either remove the reference or provide a link to more information on the web.

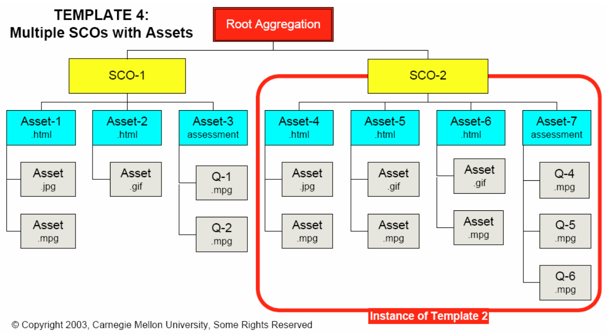
We also had to modify the navigation instructions to be very generic, so that no matter which Learning Management System this course was loaded on, courseware navigation instructions would make sense to the learner. For example, on each summary page the original content would instruct the learner to press the "main menu" button to continue. On the original LMS, that would bring the learner back to the LMS main menu where the learner could choose another lesson. By using the navigation capabilities of SCORM, we were able to override the behavior of the navigation buttons to automatically go to the next SCO.

SCORM Version

We were originally planning to convert the web-based course to SCORM 1.2 because the targeted LMS for deployment was a SCORM 1.2 LMS. However, we decided to use SCORM 2004 because it offered advanced sequencing & navigation capabilities we could use to seamlessly move between learning objectives, which was acceptable since the targeted LMS for the Air Force is in the process of being updated to be SCORM 2004 conformant.

SEQUENCING AND NAVIGATION

SCORM 2004 allows for interoperable sequencing rules to be included in the manifest file, to affect the order that the SCOs are delivered at run-time by the Learning Management System (LMS). The sequencing design for this course was quite simple. The design was intended to allow a learner to launch any SCO at any time. Once a SCO had been viewed (satisfied), the next SCO should be launched automatically by the sequencing engine. This corresponded to the behavior outlined in Template 4 (Figure 3) of the SCORM Simple Sequencing Templates and Models provided by the Learning Systems Architecture Lab at Carnegie Mellon University.



© Copyright 2003, Carnegie Mellon University, Some Rights Reserved

Figure 3. Template

There are ten templates available in the SCORM Simple Sequencing Templates and Models document. Some address pre and post test sequencing, and also remediation. For each item in the Content Package's organization, we applied a controlMode element with all attributes set to true, with the exception of the forwardOnly attribute to ensure the learner could choose their path through the course.

We also associated a sequencing rule with each item. The sequencing rule was implemented to instruct the sequencing engine to exit the SCO when it was "satisfied." This allowed the sequencing engine to flow to the next SCO when a SCO is satisfied, since the flow attribute in the controlMode element was set to true.

As we stated, we used Template 4 for our approach, and we obtained an example content package from the ADL Co-Lab in Alexandria, VA. We used this content package as our structure and replaced the content and extended the manifest to represent the AFMSIC. We used Application B of the Template 4 Rules. This template helps communication between the instructional designer and programmer. For example, our instructional strategy called for the learner to view the SCOs in any order, then the template describes what corresponding SCORM function is needed. If we had used more complicated instructional strategies like remediation using objectives with multiple SCOs, there would have been additional rollup rules.

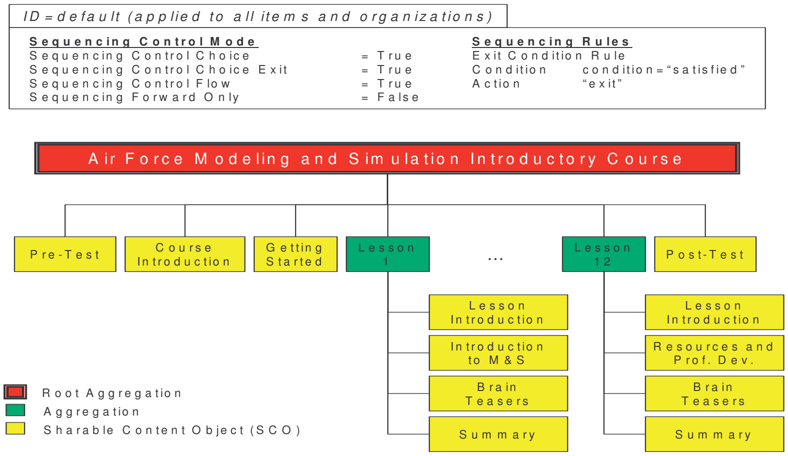
Figure 4 is a summary of our Content Structure Design and corresponding SCORM sequencing rules. It shows our course structure, or activity tree; root aggregation at the top, smaller aggregations in green, SCOs in yellow.

By applying a controlMode element with all attributes set to true, with the exception of the forwardOnly attribute, the learner could choose their path through the course, as was intended with the original design.

Some sequencing rules that were NOT chosen for this project include:

- Remediation based on assessment
- Pre- and Post- Test Sequencing (CMU Template 6)
- Skipping lessons based on assessment

The LMS that the AFMSIC currently operates in captures a score from the pre-test and the posttest, but not the individual choices for each question the learner selects. It is the intent that once the AF LMS becomes SCORM 2004, we will use SCORM 2004's ability to capture the learners' selection of each question and report back to the LMS.


ID=default (applied to all items and organizations)

Sequencing Control Mode		Sequencing Rules	
Sequencing Control Choice	= True	Exit Condition Rule	
Sequencing Control Choice Exit	= True	Condition	condition="satisfied"
Sequencing Control Flow	= True	Action	"exit"
Sequencing Forward Only	= False		

Figure 4. Activity Tree

Navigation Design

The SCORM conformant version of the course was intended to offer a similar user interface to the original course. Because it included navigation buttons as part of that interface, we decided to use those buttons to interact with the sequencing engine. This required us to modify some of the underlying JavaScript navigation handling to make calls to the SCORM Run-Time Navigation Data Model.

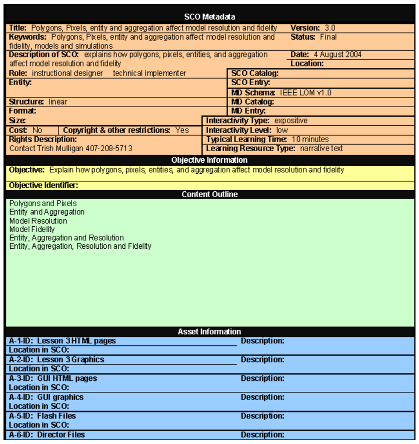
Another navigation consideration involved the Lesson introduction, brain teasers and summary pages. These were added as SCO's rather than assets to allow the SCORM Run-time Navigation Data Model to control the flow between them.

PRACTICAL TOOLS

We started out by using a Word document to help us structure the content of the course. For example, we outlined lesson 4, "Policy and Guidance," as an aggregation:

- LO4: Lesson 4: Policy and Guidance (Aggregation)
    - LO4-C-1: Lesson Introduction (SCO)
    - LO4-C-2: Modeling and Simulation Country (SCO)
    - LO4-C-3: DoD Policy and Guidance (SCO)
    - LO4-C-4: Brain Teasers (SCO)
    - LO4-C-5: Summary (SCO)

We then used the LSAL Aggregation and SCO Specification Templates to fully describe each aggregation and SCO.



SCO Metadata

Title: Polygons, Pixels, entity and aggregation affect model resolution and fidelity
Version: 3.0
Keywords: Polygons, Pixels, entity and aggregation affect model resolution and fidelity, models and simulations
Status: Final
Description of SCO: explains how polygons, pixels, entities, and aggregation affect model resolution and fidelity
Date: 4 August 2004
Location:
Role: instructional designer technical implementer
SCO Catalog:
Entity:
SCO Entry:
MD Schema: IEEE LOM v1.0
Structure: linear
MD Catalog:
Format:
MD Entry:
Size:
Interactivity Type: expositive
Cost: No
Copyright & other restrictions: Yes
Interactivity Level: low
Rights Description: Contact Trish Mulligan 407-208-5713
Typical Learning Time: 10 minutes
Learning Resource Type: narrative text

Objective Information

Objective: Explain how polygons, pixels, entities, and aggregation affect model resolution and fidelity
Objective Identifier:

Content Outline

Polygons and Pixels
Entity and Aggregation
Model Resolution
Model Fidelity
Entity, Aggregation and Resolution
Entity, Aggregation, Resolution and Fidelity

Asset Information

A-1-ID: Lesson 3 HTML pages **Description:**
Location in SCO:
A-2-ID: Lesson 3 Graphics **Description:**
Location in SCO:
A-3-ID: GUI HTML pages **Description:**
Location in SCO:
A-4-ID: GUI graphics **Description:**
Location in SCO:
A-5-ID: Flash Files **Description:**
Location in SCO:
A-6-ID: Director Files **Description:**

Figure 5. LSAL SCO Specification Template

Figure 5 represents a completed LSAL SCO Specification Template. The orange section at the top represents metadata, the yellow section represents learning objectives, the green section represents the content outline (layout), and the blue section represents the assets the SCO requires.

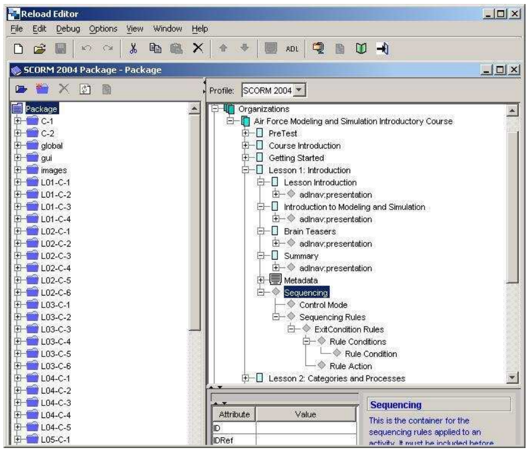
Content Modifications

The creation of this SCORM course did not involve the creation of new content since this was a conversion effort. We therefore did not need to use a content development tool to make graphics or web page assets. However, we did need to reorganize the content to streamline it for web delivery. This effort involved eliminating duplication of content by making some content global in scope such that it could be reused by various components of the resulting content package.

We used the Microsoft Windows file system to move files around to accomplish our reorganization. We then had to make changes to some of the files to reflect the new locations of content. For example, in the original content organization, a particular web page might have referred to a graphic in the same directory. If this graphic was common to many other web pages, it was moved into a common location that other web pages could reference. We therefore needed to update the web pages to reflect the new "common" location. We used a simple text editor to make such changes to web pages. One of the text editor's features that proved useful in this effort was the ability to perform search and replace operations across a collection of files. This helped expedite our content modifications.

Reload Editor

Once the content had been reorganized for web delivery, we used the Reload Editor to build our SCORM conformant content package manifest file. We had organized our content package using directories as described in our Word document content structure. For example, using the previous example, the media files representing the second SCO of lesson 4 aggregation were placed in a L04-C-2 directory. We could then drag and drop each of our SCOs from this content package to create resources in the imsmanifest file. Once a resource was created, we simply dragged that resource into the correct location in the content organization. Reload provided these capabilities, while maintaining the imsmanifest file integrity.



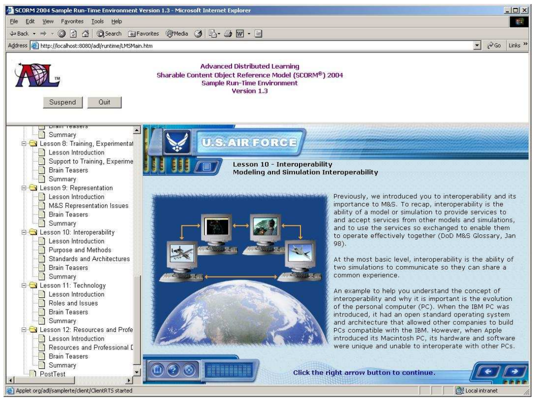
This is the container for the sequencing rules applied to an activity. It must be included before

Figure 6. The Reload Editor

Once we had the imsmanifest file content organization developed, we used Reload (see figure 6) to implement the sequencing rules previously discussed. Using Reload, we right clicked on one of our aggregations to pull up a context menu and chose to edit sequencing rules. The resulting form allowed us to set our control mode properties and sequencing rules.

SCORM Sample Run-Time Environment (RTE)

When we were done making changes to our imsmanifest file using Reload, we clicked a button in Reload to automatically export our content package as a package interchange file (PIF). We were then able to register this file with the SCORM Run-Time Environment (RTE) (see Figure 7) to see how our course might behave on a deployed Learning Management Systems (LMS).



Address: http://localhost:8080/adl/runtime/LMSMain.htm

Advanced Distributed Learning
Sharable Content Object Reference Model (SCORM®) 2004
Sample Run-Time Environment
Version 1.3

Lesson 10 - Interoperability
Modeling and Simulation Interoperability

Previously, we introduced you to interoperability and its importance to M&S. To recap, interoperability is the ability of a model or simulation to provide services to and accept services from other models and simulations, and to use the services so exchanged to enable them to operate effectively together (DoD M&S Glossary, Jan 98).

At the most basic level, interoperability is the ability of two simulations to communicate so they can share a common experience.

An example to help you understand the concept of interoperability and why it is important is the evolution of the personal computer (PC). When the IBM PC was introduced, it had an open standard operating system and architecture that allowed other companies to build PCs compatible with the IBM. However, when Apple introduced its Macintosh PC, its hardware and software were unique and unable to interoperate with other PCs.

Click the right arrow button to continue.

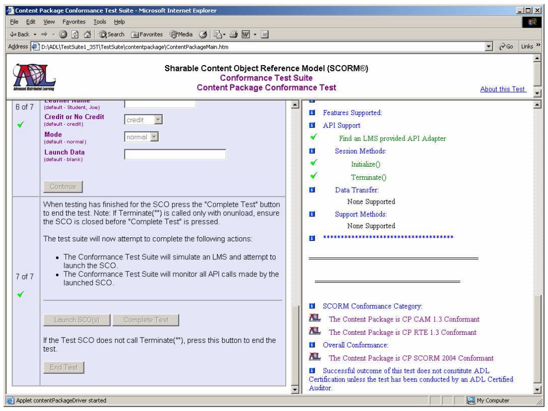
Applet org/adl/samplerte/client/ClientRTS started

Figure 7. Sample Run-Time Environment

After observing the behaviors in the RTE we would return to Reload to modify the imsmanifest file or to the media files themselves to modify the content. This was a back and forth process we repeated until the content behaved as we intended.

SCORM Conformance Test Suite

Once the content was behaving as expected in the RTE, we ran the content in the SCORM Conformance Test Suite (see Figure 8) to ensure our package interchange file adhered to the SCORM specifications and standards.


Content Package Conformance Test Suite - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Links
About this Test
Continue
Launch SCO(s)
Complete Test
End Test
My Computer

Address: D:\ADL\TestSuite1_3ST\TestSuite\contentpackage\ContentPackageMain.htm

Sharable Content Object Reference Model (SCORM®)
Conformance Test Suite
Content Package Conformance Test

6 of 7 ✓

Learner Name (default - Student, Joe)

Credit or No Credit (default - credit): credit

Mode (default - normal): normal

Launch Data (default - blank):

7 of 7 ✓

When testing has finished for the SCO press the "Complete Test" button to end the test. Note: If Terminate("") is called only with onunload, ensure the SCO is closed before "Complete Test" is pressed.

The test suite will now attempt to complete the following actions:

- The Conformance Test Suite will simulate an LMS and attempt to launch the SCO.
- The Conformance Test Suite will monitor all API calls made by the launched SCO.

If the Test SCO does not call Terminate(""), press this button to end the test.

Features Supported:
API Support
✓ Find an LMS provided API Adapter
Session Methods:
✓ Initialize()
✓ Terminate()
Data Transfer:
None Supported
Support Methods:
None Supported
****

SCORM Conformance Category:
The Content Package is CP CAM 1.3 Conformant
The Content Package is CP RTE 1.3 Conformant
Overall Conformance:
The Content Package is CP SCORM 2004 Conformant
Successful outcome of this test does not constitute ADL Certification unless the test has been conducted by an ADL Certified Auditor.

Applet contentPackageDriver started

Figure 8. Conformance Test Suite

If the Test Suite told us we had errors in our content package, we returned to Reload to make necessary modifications. Once the content package properly adhered to the SCORM specifications and standards, it passed the Test Suite, which reported that the content package was “CP SCORM 2004 Conformant.”

With the knowledge that our course was SCORM conformant, we were ready for deployment to a SCORM conformant Learning Management System. The target LMS is scheduled to be SCORM 2004 conformant in late August.

CONCLUSION

In conclusion, we would like to highlight a few points on this conversion effort. The roles of instructional designers in creating SCORM content will change. Developing or converting content may pose as a challenge at first, but being able to use more creativity in the design process should be viewed positively not negatively. Having a good team with open communication proved invaluable in working on this effort. Much like team communication, the available tools for developing or converting SCORM content is an important pre-requisite for success. The power of sequencing and navigation of SCORM 2004 proved to make it possible for our SCORM course to behave in the same way as the original course, while offering interoperability. The team is excited to test out our conversion effort once the AF LMS becomes 2004 conformant; stay tuned for the results.

REFERENCES

Air Force Agency for Modeling and Simulation Website. Retrieved June, 2005, from http://www.afams.af.mil/.

Joint ADL Co-Lab Website. Retrieved June, 2005, from http://www.jointadlcolab.orgl/.

Learning Systems Architecture Laboratory at Carnegie Mellon. SCORM Simple Sequencing Templates & Models. Retrieved June, 2005, from http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/.

Learning Systems Architecture Laboratory at Carnegie Mellon. SCO Specification Template. Retrieved June, 2005, from http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/.

Reload Editor. Retrieved June, 2005, from http://www.lsal.cmu.edu/adl/scorm/tools/reload/.

SCORM® Conformance Test Suite. Retrieved June, 2005, from http://www.adlnet.org/.

SCORM® Sample Run-Time Environment. Retrieved June, 2005, from http://www.adlnet.org/.