Sharable Content Object Reference Model (SCORM)

Laura Farinetti
Dipartimento di Automatica e Informatica
Politecnico di Torino
laura.farinetti@polito.it

The problem

- Creation and deployment of high-quality eLearning content
- Different Learning Management Systems have very different delivery environments and tracking tools
- Need for interoperable content
  - Durable, portable between systems and reusable in a modular fashion
Sharable Content Object Reference Model (SCORM)

- A collection of specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reusability of Web-based learning content
- Developed by the Advanced Distributed Learning initiative (ADL)
  - US: White House Office of Technology, Department of Defence, Department of Labor

SCORM concepts

<table>
<thead>
<tr>
<th>SCORM Concept</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusable</td>
<td>Content is independent of learning context. It can be used in numerous training situations or for many different learners with any number of development tools or delivery platforms.</td>
<td>Content developed by a refinery to train its employees to respond to a petroleum spill could be reused by the fire department as part of a hazardous materials training program.</td>
</tr>
<tr>
<td>Interoperable</td>
<td>Content will function in multiple applications, environments, and hardware and software configurations regardless of the tools used to create it and the platform on which it is delivered.</td>
<td>Content developed in one authoring system where the delivery platform is a CD on a non-networked Macintosh will also operate over the Web on a PC using both Internet Explorer and Netscape equally well.</td>
</tr>
<tr>
<td>Durable</td>
<td>Content does not require modification to operate as software systems and platforms are changed or upgraded.</td>
<td>Upgrading an operating system from Windows NT to Windows 2000 has no impact on the delivery of content to the learner.</td>
</tr>
<tr>
<td>Accessible</td>
<td>Content can be identified and located when it is needed and as it is needed to meet training and education requirements.</td>
<td>A manager can conduct an online search for training on sexual harassment and identify appropriate materials for her specific organizational needs based on information provided in the content metadata.</td>
</tr>
</tbody>
</table>
SCORM versions

- **SCORM 1.0**
  - The first version: proof of concept only
  - Introduced the notion of Shareable Content Object (SCO) and the API model in which the burden of managing communication latency across the Internet is handled by the runtime environment, not by the content objects

- **SCORM 1.1**
  - The first release: a “trial balloon”, not fully functional
  - Used a Course Structure Format XML file to describe content structure, but lacked a robust packaging manifest and support for metadata
  - Quickly abandoned in favor of SCORM 1.2

- **SCORM 1.2**
  - The first “real” release, with a real conformance test in the form of a test suite
  - Uses IMS Content Packaging specification with full content manifest and support for metadata describing the course
  - Lacks sequencing
  - No longer maintained or supported by ADL

- **SCORM 2004 (formerly SCORM 1.3)**
  - The current version
  - Includes ability to specify adaptive sequencing of activities that use the content objects
  - Includes ability to share and use information about success status for multiple learning objectives or competencies across content objects and across courses for the same learner
  - A more robust test suite
Sharable Content Object (SCO)

- Asset
  - Electronic representation of media, text, images, sounds, Web pages, assessment objects, and other pieces of data that can be delivered to a Web client
  - highly reusable: described using metadata
- Sharable Content Object
  - Collection of assets that becomes an independent, defined piece of educational material

Sharable Content Object (SCO)

- The smallest logical unit of instruction that can be delivered and tracked via a Learning Management System (LMS)
- A SCO should be able to stand alone
- SCOs cannot directly access other SCOs
  - All inter-SCO sequencing is directed by the LMS through hard-coded sequencing behavior
Examples of SCOs

- Depends on the level on which a learner should be tracked
  - A learning objective in a lesson
  - A segment on a lesson
  - A lesson in a module
  - A module in a course
  - A lesson in a course
  - A unit in a course

What SCORM specifies

- The aggregations of content objects for portability
  - How reusable web-based content objects can be aggregated into a portable package that includes a manifest to form a larger self-contained content object
  - A SCORM manifest provides a detailed description of the content of the SCORM package
What SCORM specifies

- Launching and tracking of content objects in a package
  - A runtime environment (RTE) must be used to launch the individual content objects in a SCORM conformant package, typically provided by a LMS
  - The runtime environment is completely independent of the content
  - The runtime environment launches the SCOs one at a time, according to a particular activity prescription included in the package

- Unless the activity prescription forbids it, the user can also navigate from SCO to SCO through controls provided in the runtime environment’s user interface

- The SCO must establish a communication session with the runtime environment, with a standard set of data elements

- This includes tracking data that allows the SCO to report success and progress, as well as other information about the status of content objectives, results of interactions, …
What SCORM specifies

• Content aggregation package vs. content resource package

• Content aggregation package
  ◦ Contains a manifest element named organization, that is a special section which describes how the content objects are organized for delivery
  ◦ An organization element defines a tree of activities and sub-activities that use the content objects

What SCORM specifies

• Content resource package
  ◦ Does not include any organization information
  ◦ Not intended for delivery to a learner
  ◦ Used to move amorphous collections of content objects from one system to another, or to archive a collection of content objects

• Adaptive sequencing behaviors for activities
  ◦ By default, no sequencing information associated with the activity tree
  ◦ The runtime environment shows all the activities and let the learner choose what to do
What SCORM specifies

- The creator of a package may add sequencing rules to the activity tree to prescribe guided flows through the content, adaptive sequencing and other navigation options
- SCORM specifies how to add and implement sequencing rules
- It also specifies how the tracking data reported by content objects when they are used can affect adaptive sequencing

What SCORM does not specify

- How to design learning content
  - The SCORM is neutral when it comes to pedagogy
- Look and feel
  - The SCORM does not specify what content should look like, what a runtime environment looks like, and in particular what the user interface for navigation between SCOs looks like
  - It does however assume that certain navigation facilities will be available
What SCORM does not specify

- What to do with tracking data
  - The SCORM does not specify how a LMS uses and reports tracking data collected while running SCORM content
- Granularity of SCOs and other content objects
  - The SCORM does not specify a particular granularity, size or duration for SCOs and other content objects

SCORM 2004 specification books

- Overview
  - Introduces SCORM and describes how the other books relate
- Content Aggregation Model
  - Describes packaging of content and learning object
- Run-Time Environment
  - Describes runtime API and data model used for communication between content objects and learning management systems
**SCORM 2004 specification books**

- **Sequencing and Navigation**
  - Describes how sequencing between learning activities is defined and interpreted
- **Conformance Requirements**
  - Detailed list of the conformance requirements that are verified by the ADL SCORM conformance test suite

**Content Aggregation Model**

- **Content Model**
  - Describes the content being delivered
  - If the content contains more than one module, the content model describes any relationships between those modules (called Aggregations)
  - Describes also the physical structure of the content (files needed, etc).
- **Meta-data specification**
  - Provides a mechanism to describe the content using the IEEE LOM metadata standard
Content Aggregation Model

- Content Packaging specification
  - Defines how the Content Model and Meta-data are implemented
  - Requires all content to be transferred in a folder or a ZIP file called a PIF
  - At the root of this folder there must be an XML file called “imsmanifest.xml” which contains information from the Content Model and Meta-data specifications in a well-defined format

A SCORM package

- A collection of files …
- … with a manifest that describes how the files fit together and how to deliver them …
- … and with metadata that can be used to describe the package in a catalog
A SCORM package

Directory structure

- The files in a SCORM package must be in a directory structure under a single root directory
  - The directory may be arbitrarily deep
- When the content is deployed for delivery on a web site, the files are placed in a physical or virtual directory structure that exactly mirrors the original package
  - Any links to other files within the package must be relative
The manifest

• An XML file that contains
  ◦ Metadata about the package
  ◦ Organization structures that describe the structure of the content
  ◦ An inventory of the content resources in the package
• The name of the manifest file is always “imsmanifest.xml”

Metadata about the package

• Include
  ◦ Metadata that identify the manifest as SCORM 2004 compliant
  ◦ Descriptive and administrative metadata according to the IEEE LOM
• The importing system (LMS) can inspect the manifest and mine the metadata for the needed cataloguing information
  ◦ The required elements are specified in details in the SCORM Content Aggregation Model book
Organization structures

• The manifest must contain some prescriptive information about how to deliver the content for active use by a user
  ◦ The manifest must contain at least one organization element
• The organization contains one or more activities that can be nested to any depth as sub-activities
  ◦ This tree of activities represents the structure of the content, as the package author intends it to be delivered

Organization structures

• Each activity has a title
  ◦ Is used if the package structure is shown in a table of content, or in other situations, like reports that show the status of the activity
• Each activity in the tree is either the “parent activity” in a cluster of sub-activities, or a leaf activity with no children
Organization structures

• Leaf activities reference a content object which is used when the activity is started
  ◦ A leaf activity may reference only a single content object
  ◦ When the package is delivered to a user and the user chooses to “run” a leaf activity, the corresponding content object is launched

Resources

• A manifest contains a list of resource elements
• Each resource element describes a content object
  ◦ Typically, a resource element contains a list of one or more files required to deliver the resource
• A SCO is always represented by a resource element
Resources

- Resources are either “launchable” or not
  - A launchable resource has an attribute named “href” whose value is a URL and may also contain some launch parameters
  - A “non-launchable” resource is just a container for a list of shared files used by one or more other resources
- The leaf activities in an organization may only reference launchable resources

Resources

- SCORM defines two types of launchable resources: SCO or asset
  - A SCO is a content object that will use the SCORM API to interact with the runtime environment when it is launched and while it is running
  - An asset is a content object that will not use the SCORM API but that can still be used for an activity (e.g. a text document or an image)
Resources

- A resource is launched when an activity that references that resource starts.
- Multiple activities can reference the same resource.
- Different activities that reference the same SCO may have different parameters:
  - For example, one activity might use a SCO, telling it to use level one of difficulty, while another activity might use the same SCO specifying a different level of difficulty.

Shareable Content Object (SCO)

- A special kind of content object that knows how to communicate with the runtime environment in which it is launched.
- A web content:
  - It can be launched in a web browser by using a URL.
  - It may consist of a single HTML page, or it may be a large collection of web pages and include simulations, Flash assets, or other media rich content.
Shareable Content Object (SCO)

- Basically, a small portable web site that can be copied from place to place by gathering all its files and capturing them in a SCORM package.
- To be portable, a SCO must be compatible with any generic web server:
  - It cannot depend on special services that might exist on one web server but not on another.

From traditional course structure...
... to ...

Types of Fruits and Vegetables

Fruits
- Evergreen Tree
- Tomato
- Avocado
- Banana
- Pineapple
- Papaya

Vegetables
- Cucumber
- Submarine
- Logurinious
- Lettuce
- Tomato
- Lima Beans
- Pepper
- Spinach
- Eggplant
- Assessment

© Copyright 2003, Carnegie Mellon University. Some Rights Reserved.

... SCORM

- More reusable structure: different enabling objectives

Avocados
- Avocado
- Growing Avocado Trees
- Pests that infest Avocado Trees
- Slicing Avocados
- Harvesting Avocados
- Nutritional Values of Avocados

© Carnegie Mellon University
Example of SCOs reusability

Example of SCO metadata

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title:</td>
<td>Growing Avocado Trees</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Keyword:</td>
<td>growing avocado, avocado trees</td>
</tr>
<tr>
<td>Status:</td>
<td>Draft</td>
</tr>
<tr>
<td>SCO Description:</td>
<td>This SCO provides information about varieties of avocado trees as well as techniques for raising avocados in different climates. The content is ideal for farmers and gardening enthusiasts.</td>
</tr>
<tr>
<td>Date:</td>
<td>2003-02-07</td>
</tr>
<tr>
<td>SCO Catalog:</td>
<td>xtu-avoaccess.rdf</td>
</tr>
<tr>
<td>SCO Entry:</td>
<td>1.6.avocado.xtu-avoaccess.rdf</td>
</tr>
<tr>
<td>Classification Description:</td>
<td>Avocado Trees</td>
</tr>
<tr>
<td>Format:</td>
<td>Portable document format (pdf)</td>
</tr>
<tr>
<td>Size:</td>
<td>120000 bytes</td>
</tr>
<tr>
<td>Purpose:</td>
<td>Classification Category:</td>
</tr>
<tr>
<td>Classification Category:</td>
<td>growing avocados, avocado trees</td>
</tr>
<tr>
<td>Contribution:</td>
<td>Learning Systems Architecture Lab</td>
</tr>
<tr>
<td>Role:</td>
<td>Author</td>
</tr>
<tr>
<td>MD Catalog:</td>
<td>xtu-avoaccess.rdf</td>
</tr>
<tr>
<td>MD Entry:</td>
<td>1.6.avocado.xtu-avoaccess.rdf</td>
</tr>
<tr>
<td>Learning Resource Type:</td>
<td>Narrative Text</td>
</tr>
<tr>
<td>Interactivity Type:</td>
<td>Expository</td>
</tr>
<tr>
<td>Cost:</td>
<td>No</td>
</tr>
<tr>
<td>Copyright &amp; other restrictions:</td>
<td>Yes</td>
</tr>
<tr>
<td>Interactivity Level:</td>
<td>Low</td>
</tr>
<tr>
<td>Rights Description:</td>
<td>Redistribute or reprint in full only</td>
</tr>
<tr>
<td>Credit:</td>
<td>Carnegie Mellon University, 581, etc.</td>
</tr>
<tr>
<td>Typical Learning Time:</td>
<td>0.5 hours</td>
</tr>
<tr>
<td>Location:</td>
<td><a href="http://www.lsi.cmu.edu/avo.pdf">http://www.lsi.cmu.edu/avo.pdf</a></td>
</tr>
</tbody>
</table>

© Carnegie Mellon University
Recommended metadata fields

  - from IEEE LOM

SCORM navigation

- User interface depends on the LMS
- No direct SCO to SCO navigation
- In SCORM sequencing describes and prescribes the manner in which learners receive individual SCOs from the LMS
  - this allows a set of SCOs to be sequenced in many different ways depending upon the designers who structures the content
Sequencing rules

- The author of a package may add sequencing rules to the activity organization
  - This is entirely optional
  - If no rules are specified, there is a default sequencing behavior, which is that the user gets to choose any activity at will
- If sequencing rules are defined, the package should only be delivered in a runtime environment that supports SCORM sequencing and navigation

Sequencing rules

- When a SCO is launched, it typically provides tracking data
  - The tracking data can in turn influence the result of sequencing rules
  - For example, a passing score for a SCO may result in skipping some other activity
- The rules are associated with individual activities, at any level of the activity tree
Sequencing templates

- Simple Sequencing (the IMS specification on which SCORM 2004 sequencing is based) is all but simple
  - Mastery of this very intricate model requires a deep understanding of the behaviors, how they interact or modify each other and how a runtime environment is supposed to implement them
- Useful behavior patterns can be predefined in templates
- Loss in flexibility, gain in ease of use

Components of a template

- A tree of activity nodes (items)
- Placeholders for the learning resources used to implement the activities
  - Each placeholder typically has an associated specific purpose
  - Example: a template may have placeholders for a pre-assessment, a tutorial, a remediation and a post-assessment; another template may have placeholders for learning resources that can be explored in a discovery approach
- Rules that govern the use and instantiation of the template
Simple templates - examples

- Simple sequential guided flow, which traverses an activity tree in a linear manner
- Learning activity followed by optional practice followed by assessment with remediation including optional practice if the assessment fails

Simple templates - examples

- Set of learning activities, each with a specific objective, followed by an optional practice for each objective and by an assessment that includes each objective, with remediation including optional practice for the failed objectives only
- Simple exploratory learning module, flowing from an introduction into an activity which affords the learner unlimited choice and exploration of sub-activities, followed by an optional practice and an optional assessment
A learning activity “cluster”

Unit X
  - Pre-test
  - Tutorial
    - Topic 1
    - Topic 2
    - Topic 3
  - Post-test

Unit X+1

Association with competences

Unit X
  - Pre-test
  - Tutorial
    - Topic 1
    - Topic 2
    - Topic 3
  - Post-test

Unit X+1

Relevant for: SKILLSET1
SKILL1, SKILL2, SKILL3

Relevant for: SKILLSET1
SKILL1, SKILL2, SKILL3

Relevant for: SKILL1

Relevant for: SKILL2

Relevant for: SKILL3

Relevant for: SKILLSET1
SKILL1, SKILL2, SKILL3
### Navigation: choice

- The learner can choose any activity, in any order.
- This is the only navigation mode that can be assumed in SCORM 1.2.

### Navigation: guided flow

- The designer enables the “flow” mode.
- This guides the learner through each activity in a predictable sequence.
Navigation: choice + guided flow

- The learner can choose any activity, in any order
- The learner can also follow the guided flow
  - For example, just clicking a “Continue” button will go to the next activity in the flow

Navigation: rules

- Use pre-test to determine which topics to suggest to the user in the guided flow
- Can be taken only once
  - If passed, skip to next unit
- Skip topics already mastered
  - Skip all if pre-test passed
- Once post-test taken, learner can no longer take the pre-test
- Retry until successful
Learner tries, fails, tries again

- Pre-test
- Tutorial
- Topic 1
- Topic 2
- Topic 3
- Post-test
- Unit X
- Retry until successful
- Use pre-test to determine which topics to suggest to the user in the guided flow
  - Can be taken only once
  - If passed, skip to next unit
- Skip topics already mastered
- Skip all if pre-test passed
- Once post-test taken, learner can no longer take the pretest

Scenario 1

- User masters every objective in the pre-test

- Pre-test results
  - Objective SKILL1
  - Objective SKILL2
  - Objective SKILL3
  - Objective SKILLSET1
Scenario 2

- User masters 2 of 3 objectives in the pre-test

Pre-test results
- Objective SKILL1
- Objective SKILL2
- Objective SKILL3
- Objective SKILLSET1

Scenario 2

- User masters none of the 3 objectives in the pre-test

Pre-test results
- Objective SKILL1
- Objective SKILL2
- Objective SKILL3
- Objective SKILLSET1
Sequencing templates

<table>
<thead>
<tr>
<th>Template or Model</th>
<th>Description</th>
<th>Rule Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template 1</td>
<td>Single SCO with a Single Asset</td>
<td>1</td>
</tr>
<tr>
<td>Template 2</td>
<td>Single SCO with Multiple Assets</td>
<td>1</td>
</tr>
<tr>
<td>Template 3</td>
<td>The Black Box, single SCO with multiple assets and complex internal structure</td>
<td>1</td>
</tr>
<tr>
<td>Template 4</td>
<td>Multiple SCOs with Assets</td>
<td>2</td>
</tr>
<tr>
<td>Template 5</td>
<td>Remediation Using Objectives</td>
<td>2</td>
</tr>
<tr>
<td>Template 6</td>
<td>Pre- and Post-Test Sequencing</td>
<td>1</td>
</tr>
<tr>
<td>Template 7</td>
<td>Pre- and Post-Test Sequencing (2)</td>
<td>1</td>
</tr>
<tr>
<td>Template 8</td>
<td>Remediation Using Objectives (2)</td>
<td>1</td>
</tr>
<tr>
<td>Template 9</td>
<td>Basic Branching</td>
<td>2</td>
</tr>
<tr>
<td>Template 10</td>
<td>Pre- and Post-Test Remediating with Test Options for Remediating</td>
<td>1</td>
</tr>
<tr>
<td>Model 1</td>
<td>Remediation Multiple Aggregations</td>
<td>2</td>
</tr>
<tr>
<td>Model 2</td>
<td>Mastery Testing Multiple Aggregations</td>
<td>1</td>
</tr>
<tr>
<td>Model 3</td>
<td>Pre- and Post-Test Sequencing with Aggregations</td>
<td>1</td>
</tr>
<tr>
<td>Model 4</td>
<td>Traditional CBT Branching with Multiple Decisions</td>
<td>1</td>
</tr>
</tbody>
</table>

© Carnegie Mellon University

Example of sequencing template
Example of sequencing template

**Template 4 Rules (Application A):**

<table>
<thead>
<tr>
<th>Behavior</th>
<th>SCORM Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. To complete the Root Aggregation, the learner must complete SCO-1 and SCO-2.</td>
<td>Rollup: All satisfied, completed.</td>
</tr>
<tr>
<td>2. To complete each SCO, the learner must complete the assessments within the SCOs.</td>
<td>No SCORM function</td>
</tr>
<tr>
<td>3. The learner cannot start SCO-2 until SCO-1 is complete.</td>
<td>SCO-1: If Not complete, Deny Forward Progress</td>
</tr>
<tr>
<td>4. The learner can return to SCO-1 from SCO-2 at any time.</td>
<td>Root Aggregation: Forward Only-false</td>
</tr>
</tbody>
</table>

**Template 4 Rules (Application B):**

<table>
<thead>
<tr>
<th>Behavior</th>
<th>SCORM Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. To complete the Root Aggregation, the learner must complete SCO-1 and SCO-2.</td>
<td>Rollup: All satisfied, completed.</td>
</tr>
<tr>
<td>2. To complete each SCO, the learner must complete the assessments within the SCOs.</td>
<td>No SCORM function</td>
</tr>
<tr>
<td>3. The learner can view the SCOs in any order.</td>
<td>Root Aggregation: Flow and Choice</td>
</tr>
</tbody>
</table>

© Carnegie Mellon University

---

**SCORM sequencing example**

- Photoshop course example by Advanced Distributed Learning (ADL)
  - http://www.adl.org
- Same content, sequenced several ways using different instructional strategies
- Content by the Institute for Interactive Technologies (Bloomsburg University)
**Sequencing rules**

- No sequencing rules
- Linear
  - Linear Controls
- Linear Choice
- Constrained Choice
- Knowledge Paced
- Remediation
- Competency Assessment

**No Sequencing Rules**
No Sequencing

- Basic SCORM 2004 Content Package – no explicit sequencing information specified in the manifest
- When no sequencing information is defined for a content aggregation the learner is free to choose learning activities at will, in any order, without any restrictions on number of attempts
- LMS shall provide some user interface devices that enable selection of learning activities

Linear

Photoshop Example – Linear

Introduction

Module 1: Basics
  - Lesson 1: Interface
  - Lesson 2: Toolbox
  - Lesson 3: Palettes
  - Lesson 4: Layers

Module 2: Enhancing Images
  - Lesson 5: Color Balance
  - Lesson 6: Brightness/Contrast
  - Lesson 7: Hue/Saturation

Module 3: Blending Images
  - Lesson 8: Selection Tools
  - Lesson 9: Transforms

Exam (Assessment)
  - Question 1
  - Question 2
  - Question 3
  - Question 4
  - Question 5
  - Question 6
  - Question 7
  - Question 8
  - Question 9
Linear

- The learner must progress through the content aggregation in a pre-determined order.
- The learner will encounter the introduction first, then all modules and lessons in a linear order, directed by the LMS.
- There is always the option to “go back” to view previous lessons, but the learner cannot proceed forward until completing the current lesson.
- Each module is complete when all of its lessons are complete. There are no module exams, only a comprehensive exam after all of the modules are completed.

Linear w/ Sequencing Information

Sequencing Control Mode: Flow = true, Choice = false.
Rollup Controls: Rollup Objective Satisfied if all completed; Satisfied if all satisfied; Not Satisfied if any not satisfied.
Exit Rules: Exit if completed.

Photoshop Basics - Module

Module 1: Basics
Lesson 1: Interface
Lesson 2: Toolbox
Lesson 3: Palettes
Lesson 4: Layers
Lesson 5: Color Balance
Lesson 6: Brightness/Contrast
Lesson 7: Hue/Saturation
Lesson 8: Selection Tools
Lesson 9: Transform

Module 2: Enhancing Images
Lesson 10: Enhancing Images

Module 3: Blending Images
Lesson 11: Blending Images

Module 4: Advanced
Lesson 12: Advanced

Exam (Assessment)

Rollup Controls: Rollup Objective Satisfied = false.
Terminology

- Rollup
  - How the state of child Activities and Clusters affects the state of Parent Clusters
  - E.g.: if a learner masters three out of four child Activities for a Cluster, should that Cluster be considered as mastered by this specific learner?

- Rollup rules
  - Govern how rollup works
  - Used to determine when a learner has completed the entire Activity Tree

Linear with Controls

Photoshop Example — Linear Controls

- Introduction
- Module 1: Basics
- Module 2: Enhancing Images
- Module 3: Blending Images
- Exam (Assessment)

Questions:
- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Question 6
- Question 7
- Question 8
- Question 9
Linear with Controls

- Same sequencing strategy as Linear, but content is structured differently
  - Individual lessons are not part of the content aggregation, they are embedded in various Module SCOs
  - The LMS does not provide navigation controls, the Module SCOs provide them
  - Lesson to Lesson (intra-SCO) navigation is handled by the Module SCOs
  - Module to Module (inter-SCO) navigation is handled by the LMS through the navigation data model
Linear Choice

- The learner must experience and complete the introduction first, and then is presented with a menu of module choices — the learner must select a module
- After selecting a module, the module’s lessons are presented in a predefined order; the learner cannot ‘jump’ (select) individual lessons, specific modules, or specific module exams
Linear Choice

- After selecting or ‘flowing’ into a module exam, the learner must attempt each question in order. While experiencing a module exam, the learner cannot choose to exit the module exam before completing it.
- Before completing the module he must attempt and complete the module’s exam.
- The learner is not permitted to attempt any module more than once.
- The course is complete after the learner attempts all of the modules.
The learner must experience and complete the introduction first, and then is presented with a menu of module and lesson choices – the learner must select a module or a lesson.

The learner can ‘jump’ (select) individual lessons and specific modules within the range of the constrained choice – only one module before and after the current module.
Constrained Choice

- After selecting a module or a lesson, the module’s lessons are presented in a predefined order; however, the learner can also ‘jump’ to (select) other lessons in any order.
- The module exams cannot be selected by the learner.

Constrained Choice

- After selecting or ‘flowing’ into a module exam, the learner must attempt each question in order. While experiencing a module exam, the learner cannot choose to exit the module exam before completing it.
- Before completing the module the learner must attempt the module’s exam.
- The course is complete after the learner attempts all of the modules.
Knowledge Paced

- The learner must experience and complete the introduction and then may proceed to the module 1 pre-test, select another module pre-test, or select a lesson.
- The learner may ‘jump’ between modules, selecting pre-tests or lessons in any order.
- Module post tests are not selectable by the learner; they are only encountered after ‘flowing’ through the modules lessons.

Knowledge Paced

- After selecting or ‘flowing’ into an exam (pre or post), the learner must attempt each question in order. While experiencing a module exam, the learner cannot choose to exit the module exam before completing it.
- If the learner passes an exam (pre or post), the module’s learning objective has been satisfied and the module’s post-test becomes disabled.
- The learner may continue to select individual lessons for the duration of the course, even after a module’s objective has been satisfied.
Knowledge Paced

- If the learner does not pass the exam, the learner is directed to that module’s instructional content, and once completed, must retake the module exam (post-test)
- After all of the modules have been attempted, a summary of the learner’s results is presented
- The course is completed after the learner reviews the course summary
Remediation

Module 1: Basics
Lesson 1: Toolbox
Lesson 2: Layers
Lesson 3: Palettes
Lesson 4: Layers

Module 2: Enhancing Images
Lesson 1: Color Balance
Lesson 2: Brightness/Contrast
Lesson 3: Multi-Selection

Module 3: Blending Images
Lesson 1: Color Balance
Lesson 2: Brightness/Contrast
Lesson 3: Transform

Exam (Assessment)

Introduction
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9

Part 1
Part 2
Part 3

Continued on next diagram
Remediation

- The learner must experience and complete the introduction, and then follows a ‘linear’ approach (described previously in the Linear Example)
- If the learner passes (meets all of the module objectives) the comprehensive exam, the course is completed
- For each section of the exam (module objective) that is “not satisfied”, the learner is directed to that associated Module(s) of instructional content, and once completed, must retake the Module Exam(s)

Remediation w/ Sequencing

Additional Activities

Photoshop Example -- Remediation
Primary Objective = obj_module_1
Primary Objective = obj_module_2

Module 1:
Basics

Lesson 1:
Interface

Lesson 2:
Toolbox

Lesson 3:
Palettes

Lesson 4:
Lesson 5:
Color Balance

Lesson 6:
Brightness/Contrast

Lesson 7:
Hue/Saturation

Lesson 8:
Selection Tools

Lesson 9:
Transform

Module 2:
Enhancing Images

Module 3: Blending Images

Exam (Assessment)

Introduction

Question 1

Question 2

Question 3

Question 4

Question 5

Question 6

Question 7

Question 8

Question 9

Part 1

Part 2

Part 3

Continued on next diagram
Remediation w/ Sequencing

Photoshop Example -- Competency

Introduction

Module 1: Enhancing Images
- Lesson 1: Interface
- Lesson 2: Toolbox
- Lesson 3: Palettes
- Lesson 4: Layers

Module 2: Blending Images
- Lesson 5: Color Balance
- Lesson 6: Brightness/Contrast
- Lesson 7: Hue/Saturation

Module 3: Module 2
- Lesson 8: Selection Tools
- Lesson 9: Transform

Exam (Assessment)
- Part 1
  - Question 1
  - Question 2
  - Question 3
  - Question 4
- Part 2
  - Question 5
  - Question 6
  - Question 7
- Part 3
  - Question 8
  - Question 9
  - Question 10
Competency Assessment

- This learning strategy is a variation of remediation
- The learner is first presented with the introduction
- The learner is then presented with an assessment SCO that internally evaluates the learner’s mastery of each of the Photoshop module objectives. The assessment SCO reports satisfaction status of the module objectives through the SCORM Run-time Data Model

Competency Assessment

- The learner is presented with the instructional material (modules) related to unsatisfied objectives
- After the learner has completed all of the required instructional material, if any, an exam is presented that re-tests the objectives the learner has not satisfied
Competency Assessment w/ Sequencing

Sequencing Control Mode: Flow = true; Choice = false; Forward Only = true; Objective Satisfied = true; Precondition Rule: Skip if satisfied; Rollup Controls: Rollup Objective Satisfied = false; Rollup Considerations: Required for Completed = if not skipped.

Photoshop Example -- Competency

Sequencing Control Mode: Flow = true; Choice = false; Forward Only = true; Objective Satisfied = true; Precondition Rule: Skip if satisfied; Rollup Controls: Rollup Objective Satisfied = false; Rollup Considerations: Required for Completed = if not skipped.

Primary Objective = obj_module_1

Module 1:
Basics
Lesson 1:
Interface
Lesson 2:
Toolbox
Lesson 3:
Palettes
Lesson 4:
Layers
Lesson 5:
Color Balance
Lesson 6:
Brightness/Contrast
Lesson 7:
Hue/Saturation
Lesson 8:
Selection Tools
Lesson 9:
Transform

Primary Objective = obj_module_2

Module 2:
Enhancing Images
Lesson 1:

Primary Objective = obj_module_3

Module 3:
Blending Images
Lesson 1:

Navigation rules

Control Modes  | Learner Choice (default)  | Linear (Auto Advance)  | Conditional Flow  | Forward Only
---|---|---|---|---
“Comes From”  | SS Condition
Lesson_status  | Passed
Lesson_status  | Failed
[LWS]  | Mastery
Lesson_status  | Unknown
Completed
Incomplete
[LWS]  | Progress
Unknown
[LWS]  | Attempted
MaxAttemptsMet
MaxAttemptDur
MaxActivityTime
AvailableTime, BeginEnd
Always

Condition  | Action
Passed  | Set to Skip
Failed  | Exit Parent
Mastery
Unknown
Completed
Incomplete
Progress
Unknown
Attempted
MaxAttemptsMet
Timed Out
Outside Avail.
Time Range
Always

Part 1
Part 2
Part 3
Final Exam

Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Part 1
Part 2
Part 3

Navigation rules
Terminology

- **Content package**
  - A self-contained collection of learning content that is structured according to an imsmanifest.xml file

- **Organization**
  - A specific, structured path through learning content in a content package
  - The path through learning content is designed by an instructional designer to achieve specific training outcomes
  - In most cases, an organization is a hierarchical tree structure of learning content

---

Terminology

- **Activity**
  - A unique leaf item in an organization that is delivered and tracked by the LMS
  - Either a SCO or an Asset

- **Cluster**
  - A unique, composite item containing child Activities and/or Clusters
  - Represents a related sub-set of content in an organization
  - The state of a Cluster depends on the state of its child Activities and Clusters
Terminology

• Activity Tree
  ◦ A hierarchical tree data structure that holds and manages a learner’s current progress through a collection of learning Activities and Clusters
  ◦ Used by the LMS to determine how to sequence Activities for a learner

Terminology

• Rollup
  ◦ How the state of child Activities and Clusters affect the state of Parent Clusters
  ◦ E.g.: if a learner masters three out of four child Activities for a Cluster, should that Cluster be considered as mastered by this specific learner?

• Rollup rules
  ◦ Govern how rollup works
  ◦ Used to determine when a learner has completed the entire Activity Tree
**Terminology**

- **Objective**
  - A desired outcome in online learning
  - Has a completion status (not attempted, incomplete, completed), a success status (passed or failed), a score, and a description

- **Prerequisite**
  - Some condition that must be met before a learner can access a specific Activity
  - Two forms of prerequisites:
    1) prerequisites that depend on the state of an Activity or Cluster
    2) prerequisites that depend on the state of an objective

---

**Terminology**

- **Flow Navigation**
  - A common navigation scheme where the learner uses next and back buttons to navigate through a content package

- **Choice Navigation**
  - A common navigation scheme where the learner is allowed to choose Activities of interest from a content package
Terminology

- **Control Mode**
  - Determines how the user views content in an Activity
  - A Control Mode group of variables describes:
    - Can you choose this activity’s children? (default: true)
    - Can you exit the current activity to choose another activity? (default: true)
    - Does this activity allow a valid flow sub-process to be carried through its children? (default: false)
    - When doing flow, can users go forward? (default: true)
    - Do you use the Objective Progress Information activities other than the current activity? (default: true)

---

Terminology

- **Sequencing rules:**
  - Pre-conditions
  - Post-conditions
  - Exit-conditions

- **Structure:** If [Condition(s)]. . . Then [Action]
Terminology

• Pre-condition rules
  ◦ Sequencing rules that are evaluated during a traversal request to get the correct activity for delivery

<table>
<thead>
<tr>
<th>Precondition Actions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip</td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>Hidden From Choice</td>
<td></td>
</tr>
<tr>
<td>Stop Forward Traversal</td>
<td></td>
</tr>
</tbody>
</table>

• Post-condition rules
  ◦ Sequencing rules that are evaluated after termination of an activity

<table>
<thead>
<tr>
<th>Postcondition Actions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit Parent</td>
<td></td>
</tr>
<tr>
<td>Exit All</td>
<td></td>
</tr>
<tr>
<td>Retry</td>
<td></td>
</tr>
<tr>
<td>Retry All</td>
<td></td>
</tr>
<tr>
<td>Continue</td>
<td></td>
</tr>
<tr>
<td>Previous</td>
<td></td>
</tr>
</tbody>
</table>

• Exit-condition rules
  ◦ Control termination of the activity