Best Practices Guide for the

Design and Development of SCORM Assessments

Carnegie Mellon

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*
1870/8DEA9259B1124C6AA592A9DA8246DE46

Contact Information and Redistribution Guidelines

The **Best Practices Guide for the Design and Development of SCORM Assessments** is a publication of Carnegie Mellon University.

This guide was sponsored by the Advanced Distributed Learning (ADL) Initiative Joint Co-Laboratory in Orlando, Florida. It was created to provide SCORM content developers with guidelines and parameters for their use of assessments in the Sharable Content Object Reference Model (SCORM).

The Carnegie Mellon (licensor) permits others (licensees) to copy, distribute, display, and create hyperlinks to the **Best Practices Guide for the Design and Development of SCORM Assessments** (the work). In return, credit must be given to the licensor. Licensees may not use the work for commercial purposes, without the permission of the licensor. The licensor permits others to distribute derivative works under a license identical to the one that governs the licensor's work.

Attribution

Noncommercial

Except where otherwise noted, this work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/1.0 or send a letter to:

Creative Commons
559 Nathan Abbott Way
Stanford, California 94305
USA

The **Best Practices Guide for the Design and Development of SCORM Assessments** will be available on the Joint ADL Co-Laboratory website.

NEED JADL LINK

If you are redistributing or hyper-linking to this document, please notify the Joint ADL Co-Laboratory so that we may keep you informed of the latest revisions to the guide. For more information, please contact:

NEED JADL LINK

Table of Contents

1. Overview of SCORM Assessments 1

2. General Assessment Guidelines 3
2.1 Preparing to Write a Test 3
2.2 Writing Effective Test Items 4
2.3 Common Test Items Types 4
2.4 Test Integrity 7
2.5 Test Validation 8

3. Learning Strategies with SCORM Assessments 9
3.1 Pre-Tests 9
3.2 Post-Tests 10
3.3 Knowledge Checks or Practical Exercises 11
3.4 Mastery Tests 11
3.5 Remediation 12

4. Structuring SCORM Assessments 14
4.1 Assessments as a Single SCO 14
4.2 Assessments as Multiple SCOs 14

5. Creating Test Banks 15

6. Identifying Data Collection Requirements 16
6.1 Common Data Collection Needs 16
6.2 Data Evaluation Periods 17

7. Understanding cmi.interactions 18

8. Sequencing and Rollup Rules 19
8.1 LSAL Templates 19
8.2 LSAL Templates with SLAG Assessments 19

9. Securing Your Assessments 20

About the Authors 21

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*
1870/8DEA9259B1124C6AA592A9DA8246DE46

1. Overview of SCORM Assessments

SCORM has never directly addressed how to create assessments nor when and how a SCO should be considered an assessment.

This has created confusion in the instructional design and content development communities. Developers wonder if assessments can be delivered in SCORM-conformant content and, if they can, then how they should be structured to collect the assessment data and metrics. Designers require this data for the validity and reliability reporting that is common to the summative and formative evaluation processes, especially in the US Department of Defense (DoD).

Without guidance on how to create assessments within SCORM content, many organizations have resorted to delivering assessments using proprietary assessment solutions provided by their learning management systems (LMSs). This is not the ideal solution because assessments and assessment metrics defined within a given LMS solution are not interoperable or reusable in other LMSs or other learning experiences.

Prior to the release of SCORM 2004, the amount of data that could be collected and stored, in an interoperable manner, about a learner's progress within individual SCOs was limited. LMSs were only required to support a subset of the defined CMI data elements. For example, one of the data elements that LMSs were not required to support was `cmi.interactions`. The `cmi.interactions` data element provides a detailed model for designers and developers to collect metrics about learner responses or performance within a SCO, particularly data related to performance on assessments such as correct response, learner's response, duration taken to respond, and weight of the particular item relative to the overall assessment score.

SCORM 2004 now requires any SCORM-conformant LMS to support all of the CMI data elements defined in the SCORM Run-Time Environment (RTE). Most data elements require little explanation to allow a content developer or instructional designer to use them in SCORM content; for example, the description of a typical CMI data element is usually about one half to one page in the SCORM RTE book. The `cmi.interactions` element is exceedingly complex; its description is approximately 25% of the SCORM RTE book because of the detailed data collection this data element enables.

By properly using `cmi.interactions`, instructional designers can now collect metrics for formative and summative evaluation reporting, and link `cmi.interactions` and other CMI data elements to sequencing rules to create remediation or adaptive learning strategies that provide further customized learning experiences. This may significantly improve learner performance by allowing designers to better match performance on individual learning objectives with remediation and feedback strategies. With the guidance provided in this document on when and how to use `cmi.interactions` in assessments, designers will be able to create robust assessments using SCORM 2004 that meet both the delivery and data collection requirements of their clients and ensure learner mastery of instructional material.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*
1870/8DEA9259B1124C6AA592A9DA8246DE46

The primary objectives of these guidelines are to understand and document the requirements for creating assessments within SCORM-conformant content, and document best practices for the design and development of assessments within SCORM-conformant content. A set of sample content packages with assessment SCOs was also created. They can be used as assessment templates to provide guidance to instructional designers and content developers in the ADL SCORM community.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

2. General Assessment Guidelines

The most important thing to remember before you write test questions is that your test items should be structured to measure learners' understanding or knowledge of a **learning objective**, not of the actual learning content.

2.1 Preparing to Write a Test

Before you begin writing test items for any application, you should have written the learning objectives and had them approved. Always write the test items before scripting and storyboarding the content, or you may end up with tests that address the learning materials, often focusing on some secondary information or details, rather than ensuring that learners have achieved the desired outcomes.

Table 2.1 outlines some key points you need to keep in mind when preparing test items. The left-hand column lists the goals for your test items; the right-hand column lists some tips on how to develop test items that meet these goals.

Table 2.1 Test item development goals and guidelines

Your test items should	So you need to
Use actual performance situations or simulations to realistically measure learning objectives whenever possible.	• Establish a mastery standard for each objective. • Write test questions that enable learners to demonstrate what they know.
Refer to an established measurement criterion.	• Define criterion for learners to measure their performance according to some published standard, such as a training manual.
Require actual performance or be based on performance.	• Write questions that correlate a learner's responses with what learners actually do on the job rather than asking them to recall memorized facts. • Use realistic mission or job scenarios that learners will encounter. • Present cues and distractors as close to those experienced in job performance as possible.
Test the particular skill or knowledge appropriately.	• Avoid writing questions that exceed the learning requirements for the content.
Match the practical exercises in both the level and type of performance required.	• Avoid practical exercises that are watered-down versions of test questions or vice versa. • Test items should be interchangeable with practical exercises.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

2.2 Writing Effective Test Items

Ideally, each learning objective in your content should be tested twice: once within a section of instructional materials as a knowledge check or practical exercise, and once at the completion of the instruction to assess mastery.

Knowledge checks or practical exercises are used to help learners ensure they comprehend the content as they proceed through it and before they are expected to master it.

You should generate at least three to five test items per learning objective. Strive to write test items that are clear, complete and concise. Learners should always understand what is being asked of them, even if they don't know the correct response to an item. Never try to trick learners with clever distractors, or use humorous or implausible distractors just to have more distractors in a multiple choice question.

Remember to:

- Use standardized punctuation and capitalization.
- Use language, abbreviations, acronyms and grammar consistently.
- Use only the most common and familiar abbreviations and acronyms.

2.3 Common Test Item Types

There is a wide range of item types available for e-learning applications, including multiple choice, true/false, matching, drag-and-drop/interactive, and fill-in-the-blank. Multiple choice items are the most common item type used in assessing e-learning content, but they are also the most difficult to write well. Be aware that each type of test item has its advantages and disadvantages.

Multiple Choice	True/False	Matching	Drag-and-Drop or Interactive	Fill-in-the-Blank
• Tests many learning levels • Easy to program and score in e-learning • Difficult to write	• Allows learners to easily guess correctly • Easy to program and score in e-learning	• Allow learners to assess relationships between items • Easy to program and score in e-learning with templates	• Allows learners to interact directly with content • Difficult to program in e-learning	• Forces learners to understand content • Difficult to program and score in e-learning

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

2.3.1 Multiple Choice (MC) Items

Punctuate all responses for an item the same way. The punctuation you use for a set of responses depends on the form of the stem. Responses for stems that are questions are punctuated differently from responses for stems that are incomplete sentences. For stems that are constructed as incomplete sentences, do not use any punctuation at the end of the stem and do not include an underlined, blank space. Simply leave the sentence unfinished.

Write stems that:	Avoid stems that:
• Contain as much information related to the item as possible. • Clearly and simply state the question (even for highly technical content). • Include the portion requiring completion at the end of the stem **without** an underlined, blank space and **without** end-of-line punctuation.	• "Trick" the learner into answering incorrectly. • Force the learner to struggle to understand the item or a complex sentence structure in the item. • Use negatively worded stems like "which of these is **not** correct." • Contain unnecessary or irrelevant information that does not directly pertain to the item.

Multiple choice items typically have four or five response options. However, if you cannot write plausible distractors (incorrect responses that seem to be likely answers to the learner), then use fewer, well-written response options. Items on a given test do not have to use a consistent number of distractors.

Ensure there is one, and only one, correct answer for the item. If you believe that a response could be considered correct under certain circumstances, then rewrite either the stem or the response options.

Make item response structures **parallel** within a given item so that each response looks and sounds as likely as the next option.

- Lead with the same part of speech (all verbs, all nouns, or all gerunds).
- Follow the same sentence structure (all could complete the sentence formed in the stem, all have the same punctuation, etc.).

Responses should be put in a **random order** to help make tests unique for each learner. Never randomize items that contain ordered numerical or alphabetic values. Place responses in a **logical order** if they are:

- Sequential – the responses involve a process
- Alphabetical – the responses are single letters
- Numerical – the responses contain single numeric values
    - Ascending or descending order is acceptable

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

Below is a summary of “dos” and “don’ts” for writing responses for multiple choice test items.

Write responses that:	Avoid responses that:
• Are relatively succinct, but are all approximately the same length. • Provide plausible answers to the item. • Contain one, and only one, correct answer. • Are randomized EXCEPT when – Numbers are used. – Single letters are included. – A logical order of the responses is obvious.	• Are much shorter or much longer than other responses and give away the correct answer. • Can be easily discarded as a possible answer. • Use options that are familiar to the learner or are true, but do not correctly answer the item. • Are negatively worded. • Use phrases like “all of the above”, or “none of the above,” etc. • Contain concrete determiners like “always”, “never”, “all”, and “none”.

2.3.2 True/False Items

True/false items are easy to write and easy to score in e-learning; however, most good test writers avoid using true/false items, particularly during mastery testing. Since there are only two options, learners have an excellent chance of responding correctly. If you must use true/false items, then focus your use of them on knowledge checks or practical exercises where learners’ scores are not tracked.

2.3.3 Matching Items

Matching items work particularly well for assessing learners’ knowledge of vocabulary terminology and relationships between items. If your authoring tool provides templates, then matching items are easy to program and develop for e-learning. To ensure learners aren’t using the process of elimination to respond correctly, try to use an uneven number of options on matching items.

2.3.4 Drag-and-Drop / Interactive Items

E-learning presents its greatest advantage over other forms of assessment when drag-and-drop or interactive items are used properly. They allow learners to interact with content as they might in a real-world situation. Drag-and-drops are most effective as simulations for processes where learners have to perform one or more steps to receive credit for correctly responding to an item. They are also useful for visually rich information where learners have to recognize or move pieces of information.

Design drag-and-drop items that do the following:	Avoid drag-and-drop items that do the following:
• Are as close to real-life situations or applications as possible • Mimic the conditions under which learners would typically respond	• Require learners to interact solely for the sake of interacting

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

2.3.5 Fill-in-the-Blank Items

Fill-in-the-blank or short answer items can be one of the best ways to determine if a learner knows the content because the learner is not likely to guess the correct response. Construct this type of item with a **blank space** for which the learner will supply a brief response. Use only one blank per item. For clarity, place the underlined, blank space as near to the end of the item as possible. Specify the **units** required for the response in a computational item, such as “pounds” or “tons,” except when the units are also important for achieving the learning objective.

While fill-in-the-blank items have the advantage of preventing learners from merely guessing, these items are difficult to grade in e-learning because a machine and not a human must grade learners responses to the items. This means that the test writer must either account for all possible correct answers to ensure that the learner receives the proper credit for the item, or very clearly specify to learners the constraints for the response, such as “one word”, “whole number”, or “no commas”.

If it is unreasonable to account for all possible answers a learner might provide (variations in capitalization, spelling, punctuation, and possible answers), then avoid using the fill-in-the-blank item type. Because of the difficulties associated with anticipating all the correct variations that learners may input as their responses, the assessment templates described below allow for fill-in-the-blank items with **numeric responses only**.

Write fill-in-the-blank items that:	Avoid fill-in-the-blank items that:
• Place the underlined, blank space as near to the end of the item as possible. • Allow the correct response to be as brief as possible. • Specify the units required for the response in a computational item, except when the units are also important for achieving the learning objective.	• Put the underlined, blank space at the beginning of the item. • Use more than two underlined, blank spaces in an item.

2.4 Test Integrity

Since you put so much effort into writing effective test items, you want to ensure that your learners take both their learning experience and their testing experience seriously. One of the ways you can do this is by denying access to learning materials except in the case of references that are provided directly via a link on the test item screen or by an image on the test item screen.

Randomization of the items and the responses within a particular item is another way to ensure the integrity of your assessment. Randomization helps to ensure that no two learners receive the same combination of items, making it more difficult for them to share answers or create cheat sheets. Most authoring systems provide a mechanism by which you can quickly and easily randomize both the items within an assessment and the distractors for each assessment. Refer to Section 2.3.1 for additional considerations when randomizing multiple choice items.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

2.5 Test Validation

Before tests can be used to assess learners' knowledge of course content, they must be validated to ensure that they actually assess the learning objectives and that all of the test items are well-written, clear, and error-free. The instructional designer should review and approve all tests before they are finalized. Tests should also be reviewed by the subject matter expert (SME) to ensure they are technically accurate.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*
1870/8DEA9259B1124C6AA592A9DA8246DE46

3. Learning Strategies With SCORM Assessments

With the addition of sequencing, SCORM 2004 enables numerous learning strategies. This section describes the most common learning strategies in general terms to acquaint you with some of the possible roles for your assessments.

3.1 Pre-Tests

Pre-tests are most commonly used to identify learners' knowledge before beginning the instruction and then measure the change in learning that occurs as a result of the instruction. Pre-tests are also useful for diagnosing learners' current knowledge, skills and abilities. Learners who score high on a pre-test may be allowed to test out of the instruction about which they already have the requisite knowledge, skills and abilities.

When pre-tests are used to measure learners' knowledge, skills or abilities before completing the instruction, all learners may be required to complete all sections of the content even if they have demonstrated mastery of that instruction during the pre-test. When this strategy is used, the instructional designer typically compares learners' pre-test and post-test (mastery) scores to see how much learners actually gained as a direct result of the instruction. There are no real values associated with this type of strategy for learners; however, you may find it helpful for evaluating and revising your instruction in the future.

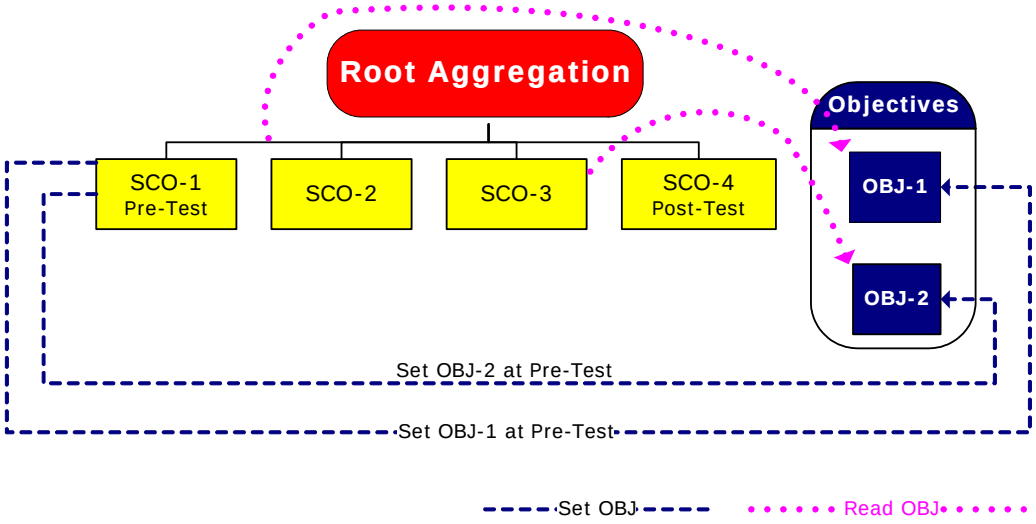
Pre-tests that diagnose learners' existing knowledge, skills or abilities map each test item to specific learning objectives within the planned instruction. As learners pass items on the pre-test, your SCO can trigger variables in the LMS that enable learners to bypass the sections of content for which they have demonstrated mastery. This strategy is called test-out.

With test-out, learners with mastery of existing instruction can complete the instruction more quickly by moving ahead to those areas that are new or more advanced. This also creates what is known as customized learning – giving each individual learner just what they "need to know" instead of extraneous materials they already know.

The following diagram shows Carnegie Mellon's Sequencing Template 6. This is one way of structuring a content package containing both a pre- and post-test and using sequencing objectives to pass information between the SCO and the LMS. In this diagram, the pre-test SCO is programmed to set sequencing objectives in the LMS as learners take the test. Before SCO-2 and SCO-3 launch, the LMS will read the sequencing objectives that were set by the pre-test to determine whether or not to show SCO-2 and SCO-3.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

TEMPLATE 6: Pre- and Post-Test Sequencing (1)



© Copyright 2004, Learning Systems Architecture Lab, Some Rights Reserved (CC: by-nc-sa/2.0)

Typically, if you are allowing learners to pre-test out of an entire SCORM content package, the passing criterion — the score required to test out — should be set relatively high. Learners should only be given **one attempt** to respond correctly. Many organizations won't allow learners to test-out of instruction unless they score 95% or higher on a comprehensive pre-test. Other organizations set the passing criterion somewhat lower.

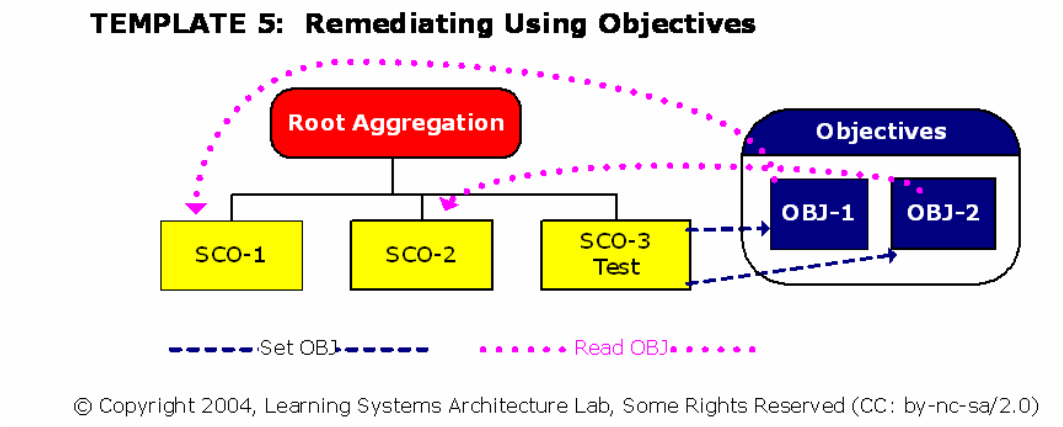
You should base the passing criterion on the criticality of the content and the contract requirements for your project. If you are instructing emergency or critical procedures, safety information or any other area where loss of life of limb could result from improper instruction, then your passing criterion should be set very high. In those cases, you may even want to prohibit learners from testing out of certain topics that are absolutely critical. If your instruction involves less critical content, then you may consider setting a lower test-out score to allow more learners to complete the content more quickly.

3.2 Post-Tests

After learners have responded to all the items on a post-test, they should be given **one or two attempts** to respond correctly to any missed items. You will have to determine the passing criterion for each post-test depending on the requirements of your contract or statement of work.

During both pre-test and post-test exams, you need to ensure that the answers are not accessible to learners until they have completed the learning experience. If you decide to make the pre-test and post-test exams identical, then do not reveal any correct answers to missed items to the learners after they complete the pre-test.

The diagram below shows Carnegie Mellon's Sequencing Template 5. It shows how a content package can be structured with multiple content SCOs and a post-test. (It also shows a remediation strategy; see Section 3.5 for more about using this template for Remediation.)



3.3 Knowledge Checks or Practical Exercises

Knowledge checks, also called practical exercises, help learners ensure they are comprehending the content as they proceed through it (within a given SCO) and before they are expected to master it. A good rule of thumb is to present a knowledge check after learners have seen 3 to 5 screens of content.

Knowledge checks are **not scored** and the score **is not reported** to the LMS. After submitting their response, learners receive **immediate feedback**. Typically learners should be given **two attempts** to respond correctly. The feedback messages that learners receive should be polite and brief; never insulting, cute or funny.

3.4 Mastery Tests

Mastery testing is a testing strategy used by instructional designers to ensure that learners fully comprehend the content. Mastery testing involves using several test items (three to five questions is considered acceptable) to validate a learner's understanding of a learning objective. However, the more test items used to validate a particular learning objective, the greater the possibility of ascertaining the learner's mastery of the content.

Mastery testing is typically conducted at the conclusion of a course or curriculum. Mastery tests are **scored** and the score **is reported** to the LMS. Learners generally receive **delayed feedback**; that is, feedback on which items were answered correctly and incorrectly is conveyed only after a learner completes and submits an exam.

Based on the score received on the mastery test, learners can either pass or fail it. If the assessment is failed, then learners may be assigned to remediation (see discussion below). In Carnegie Mellon's Sequencing Templates 5 and 6 shown above, the post-test data is not only reported back to the LMS, it is also used to create remediation for learners who do not pass the mastery test.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

3.5 Remediation

Remediation is an instructional strategy used to help learners comprehend instruction that they may be struggling with. There are several forms of testing remediation. In the event that learners do not answer a test item correctly, or fail an entire test, you can remediate them in one of several ways.

Learners could be remediated to content they have already viewed and forced to review it since they may not have grasped everything on their first attempt. This method is somewhat effective, particularly if learners are not anticipating a test and fail to pay close attention to the instructional materials in the hope that they won't need to learn them. This is one of the least costly remediation strategies since you are reusing previously viewed content.

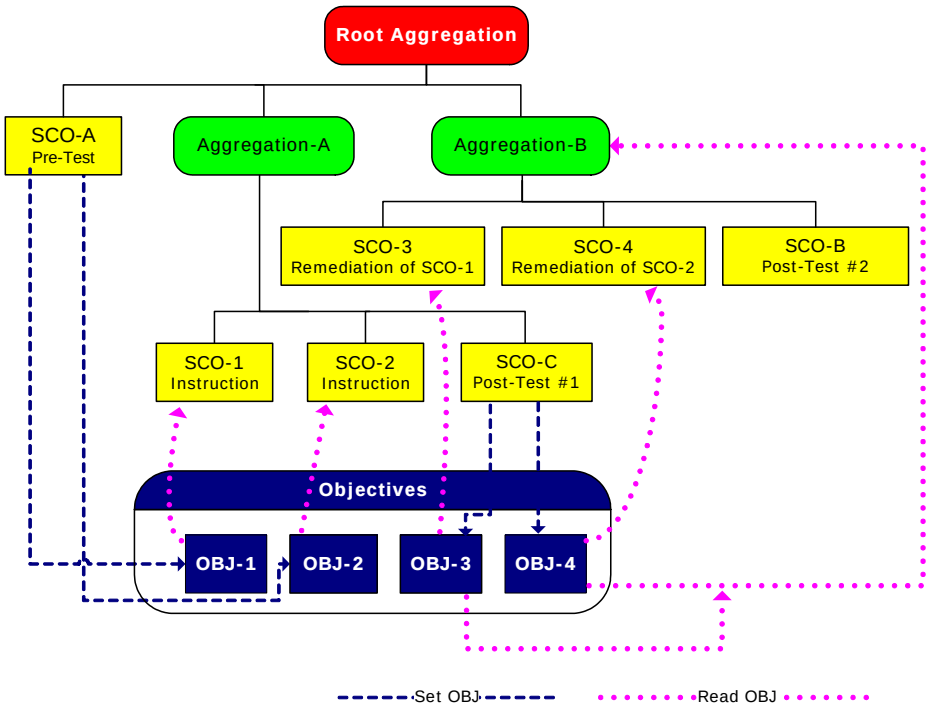
In Carnegie Mellon's Sequencing Template 5 shown above, the post-test results remediate learners to content they have already viewed based on an objective by objective basis – that is, learners only view the content related to learning objectives that they failed; they are not forced to review all of the content.

Another way of remediating learners is to present the same instruction in a new or different manner. This is the most effective remediation strategy since everyone learns differently. Learners who may have trouble understanding the content the first way it is presented, but when presented with new examples or diagrams, they may understand the content immediately. While being very effective, this strategy can be very costly, since it involves creating twice as much content. However, since many learners need to see things expressed in several ways before they are able to fully comprehend them, the cost expended in creating the materials could be recovered during training.

In Carnegie Mellon's Sequencing Template 10, on the following page, learners' results on the post-test remediate them to new content, in a distinct aggregation, on an objective-by-objective basis – that is, learners only view the new content related to the learning objectives that they failed; they are not forced to complete all of the new content.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*
1870/8DEA9259B1124C6AA592A9DA8246DE46

TEMPLATE 10: Pre- and Post-Test Sequencing With New Content for Remediation



© Copyright 2004, Learning Systems Architecture Lab, Some Rights Reserved (CC: by-nc-sa/2.0)

If sending learners back to the same content to review it or creating two sets of content for the same instruction is too time consuming for the type of content you are creating, then consider providing feedback via each test item as a form of remediation. By doing this, learners can see what the correct responses are as well as seeing a clarification of why they are correct, so learners actually can “learn from their mistakes”.

4. Structuring SCORM Assessments

SCORM assessments can be structured in two ways: as a single SCO containing multiple test items or as multiple SCOs, each of which has a single test item. Each assessment SCO must report a score and completion status, regardless of the way in which each type is structured.

4.1 Assessments as a Single SCO

When an assessment is presented to learners as a single SCO, the LMS receives one comprehensive score and a completion status is set. Using `cmi.interactions`, you can collect detailed response information for each test item in an ***interoperable*** way. This facilitates the formative and summative evaluation process. It also preserves the integrity of learners' testing experiences, since within a single SCO test items can be presented one at a time or on a single, scrolling screen. You can also control randomization of the test items *inside* the SCO, and you don't have to write as many sequencing rules to control the presentation and remediation behaviors of each test item.

This level of functionality within a SCO does require more time to program, but again, there are fewer sequencing rules required. Should you need to change a single item within the assessment, you'll have to update the entire SCO.

4.2 Assessments as Multiple SCOs

When an assessment is presented to learners as multiple SCOs – a series of SCOs, each containing one test item – the LMS can receive a score and a completion status for each SCO. To compile a comprehensive score, you can use the rollup function in sequencing, but this is a complicated function to program. Using `cmi.interactions`, you can collect detailed response information for each test item in an ***interoperable*** way. This facilitates the formative and summative evaluation process. To move between test items, you can use sequencing rules.

It is relatively easy to update an assessment created as multiple SCOs if you only need to change a single test item since you can remove one SCO and update the sequencing rules accordingly. Unfortunately, delivering an assessment as multiple SCOs can create an interruption in the "test flow" for learners. Since the LMS must evaluate each SCO individually, and some LMSs may require learners to manually submit each SCO, the processing time may create a slight delay after each test item as learners navigate between the LMS and the assessment SCOs. This can result in a loss of concentration by learners.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

5. Creating Test Banks

Test banks are one of the easiest ways to ensure the security of your tests. Ideally, you should provide a bank of questions from which a given number of items are pulled randomly to test each learning objective.

For a standard assessment, you should write at least three to five test items per learning objective to assess learners' mastery. If you want to either ensure that the likelihood of two learners receiving exactly the same assessment is limited or if you want to ensure that on multiple attempts the same learner does not receive exactly the same test, then test banks are one way of controlling your assessments.

Test banks randomly select a given number of test items from a pool of related test items. The related test items should be grouped by learning objective, so if, for example, you had an assessment with seven learning objectives, you would create seven distinct groups of test items. The test bank should randomly pull three to five test items, the same number you would use in a standard assessment. This means that you'll actually need to generate at least seven to nine test items per learning objective to ensure you achieve sufficient randomization. While it may be difficult to think about writing seven to nine valid test items for a single learning objective, remember that you can vary the item type (multiple choice, true/false, drag-and-drop, and completion) for similar item content. You can also vary the responses on a multiple choice item or simply reword the stem.

Section 2.4 of this guide addressed randomization of test items. Not only should the responses within a multiple choice item be randomized, but the delivery order of the test items should also be randomized. With a test bank, you will also be randomizing the delivery of a set number of test items, which further ensures the integrity of your assessments.

When you create test banks for SCORM assessments, you can either create a single SCO as an item bank for a given learning objective (in which case you would require multiple SCOs for a complete assessment) or you can create a single SCO made up of multiple test item banks. The most effective way to structure your assessment would be with a single SCO comprised of multiple test item banks. Using `cmi.interactions,` you will still be able to collect all of the data you require for each item, and you will preserve the flow of the learners' experience as they proceed through the assessment.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

6. Identifying Data Collection Requirements

The amount of data you can collect from an assessment can be significant, but there are certain types of data that will help you to ensure that your learners are receiving the best possible assessments and achieving their best possible performance.

Balancing the amount of programming required to collect the data against the way you plan to use the data is essential in SCORM assessments. Deciding to collect all of the data you possibly can about learners' performance on an assessment can add countless hours of programming to your project, so before you begin there are a number of questions you should ask your team and/or your client about the data collection requirements for a given project. Some of those questions include:

- Who will have access to the data?
    - Will the data be aggregated or reported individually?
    - What security considerations are there for the data if it is reported individually?
- How will the data you collect be used?
    - Will the data be used to improve the content and the assessment?
    - Will the data be used to identify individual learners who excel or perform poorly?
- What stage of the evaluation process are you in, formative or summative?
    - What are the contractual reporting requirements for the project in this evaluation stage?
    - How must the reported data be delivered, as aggregated or individual data?
- What type of reporting is provided by your LMS to facilitate data reporting?
    - How many report options are provided?
    - Are the reports easy to run and automatically formatted?

6.1 Common Data Collection Needs

The most common data collected from assessments is addressed by `cmi.interactions`. The following table shows how the data collection elements in `cmi.interactions` can be used to evaluate your content.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

Data Collection Element	Common Usage
ID	An identifier that is unique to the individual test item – useful for randomized assessments so you always associate the same information with this item
Type	The type of test item (multiple choice, true/false, matching, etc.)
Objectives ID	An identifier that can be used to associate the test item with a database of learning objectives
Timestamp	The exact time at which the item was attempted
Latency	The difference in time between when an item was attempted and completed – useful for determining how long learners took to respond to the item
Correct Response Pattern	The correct response to the item
Learner Response	The response made by the learner
Weighting	The relative value of the item compared to other items in the assessment
Result	Whether the learner's response was correct or incorrect
Description	A textual explanation about the test item – may be used for any purpose

6.2 Data Evaluation Periods

6.2.1 Formative

Formative evaluation allows you to assess the effectiveness of the materials throughout the development process. This evaluation stage may include small-group tryouts of your assessments to check that they are both functioning properly from a technical perspective and validating the intended learning outcomes. The goal of formative evaluation is to be able to revise your materials before they have been finalized to minimize the cost to the project.

6.2.2 Summative

After the initial project release, summative evaluation allows you to collect data about how real learners are performing in the first few learning sessions. Based on the data you collect, you can revise your learning materials accordingly to ensure they are as effective as possible. For example, if you find that half of your learners are responding to a single multiple choice item in one of your assessments incorrectly, then you may decide to look more deeply at both the test item itself and the content it assesses to make sure that you are not confusing or misleading learners.

6.2.3 Operational

Once your project is stable and in constant operation, it is important to periodically review the assessment data you are collecting to ensure that your learners continue to perform effectively. A review at this stage will allow you to further improve the materials and guarantee the ongoing instructional effectiveness of your assessments.

NOTE: Sections 7 – 9 are designed for programmers responsible for implementing SCORM assessments.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

7. Understanding cmi.interactions

The SCORM Data Model's `cmi.interactions` element allows a SCO to store data on the LMS about a learner's performance on an assessment.

A single interaction describes a test item. For a single test item, a single interaction can store:

- question **identifier** – used to associate the question to a database or master list of questions
- **type** of question – multiple-choice, true/false, matching, etc.
- **order of the responses** as they were presented to the learner – if the order of the responses was randomized
- **correct answer** – what the learner should have answered
- learner's **response** – what the learner actually answered
- whether the learner's response was **correct** or not

An assessment can report one interaction for every test item in the assessment. SCORM requires an LMS to be able to store 250 interactions per SCO.

An LMS can only store the information that the SCO is programmed to send to it. The information stored in `cmi.interactions` is not used by the LMS to compute a score for each assessment SCO, so you must program the SCO to compute the score and store it using the `cmi.score.scaled` element.

The instructional designer must convey the real reporting needs to the programmer. It may be sufficient to just store the details of every test item on the LMS for new assessments (to ensure that they were programmed correctly and that the assessment is properly testing the learners) or for "final exams." Collecting too much data about assessments:

- Increases programming time (which will increase the cost to the project)
- Increases the amount of communication between the SCO and the LMS (which may slow LMS response time)
- Increases the storage requirements of the LMS (which may increase the amount of total storage needed)
- May violate the rights of learners in some cases (particularly in situations where learners are union employees)

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

8. Sequencing and Rollup Rules

8.1 LSAL Templates

The LSAL Templates that ship with the Reload Editor have been updated based on increased understanding and experience in working with the SCORM Sequencing specification. The changes to the templates include:

- Use of complete URNs for objective identifiers – for example, the old templates had objective identifiers such as “global-obj-1”; the new templates have URNs such as “urn:lsal.org/template8/global-obj-1”. The SCORM documentation recommends that the identifiers for objectives be fully qualified URNs.
- Discontinued use of primary objectives for mapping to and from global objectives – primary objectives were causing some problems with sequencing and remediation loops because the primary objective could be automatically set to “complete” and “satisfied” when the SCO was finished, which could overwrite values set by the SCO. Using non-primary objectives solves this problem.

8.2 LSAL Templates with SLAG Assessments

The LSAL Templates that formerly shipped with Reload Editor 2004 used a very simple assessment for setting the values of objectives. These simple assessments have been replaced with prototype assessments generated using SLAG (SCORM Learner Assessment Generator) as another way to illustrate how to create assessments using the SLAG toolset.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

9. Securing Your Assessments

Many content developers are afraid that learners could look at the source code for assessments to determine the correct answers to test items. This quick tip provides a simple way to provide a measure of security for your assessments when they are delivered through a SCORM-conformant LMS.

A SCO is typically composed of a set of static HTML and other files stored on a web server and delivered to a learner. A savvy learner may view the source for the HTML file and reverse-engineer your assessments to discover the correct answers.

A simple way to secure your assessments is to put each assessment into another HTML file and store the name of the file as `SCO launch_data`. The name of the file will only be retrieved during the running of the SCO, and a learner will not be able to discover the name of the file by viewing the HTML source files.

In your content package, you specify the name of the file using the dataFromLMS field:

```
<item identifier="ITEM-1" identifierref="RES-1">
  <title>Sample Assessment</title>
  <adlcp:dataFromLMS>hidden_assessment.html</adlcp:dataFromLMS>
</item>
```

Then, when the learner is ready to take the test, the SCO gets the name of the file using the cmi.launch_data element:

```
<frameset rows="*">
  <script language="javascript">
    var assessment_file = doGetValue( "cmi.launch_data" );

    document.write( "<frame src=\""
      + assessment_file
      + "\" name=\"mainFrame\" />" );
  </script>
</frameset>
```

Use this simple strategy to make your assessment SCOs secure so you will not have to re-create new assessments just because learners gained access to the answers.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

About the Authors

Nina Pasini Deibler, MA, MAS
ndeibler@acceleraromar.com

Nina Deibler is a Principal Consultant and Instructional Designer at AcceleraRomar where she works to improve the performance of sales representatives in the pharmaceutical industry. Ms. Deibler has over ten years experience designing and developing learning solutions for clients in academia, industry, and government. Ms. Deibler has worked extensively with learning standards and learning standards organizations including the Advanced Distributed Learning Initiative, IMS Global Learning Consortium, and MedBiquitous. Much of her work includes teaching other instructional designers and content developers how to design and develop SCORM-conformant learning materials.

Prior to joining AcceleraRomar, Ms. Deibler was a senior instructional systems designer with the Learning Systems Architecture Lab at Carnegie Mellon University and The Boeing Company. Her work includes the “Defense Ammunition Center (DAC) e-Learning Style Guide”, “SCORM Best Practices Guide for Content Developers” and the “IRS Content Developers’ Guide to SCORM,” as well as workshops to assist content developers and programmers implementing SCORM-compliant instructional materials.

Ms. Deibler earned her MA in Human Communication from Abilene Christian University and her MAS in Operations from Embry-Riddle Aeronautical University. She received her BA from Grove City College.

William H. Blackmon, PhD
bill.blackmon@adlnet.gov

William H. Blackmon, PhD, is the Chief Technical Officer for the Advanced Distributed Learning Initiative. He has been involved in the successful adoption of SCORM and the development of the ADL-Registry and the CORDRA architecture. As the Systems Architect for the Learning Systems Architecture Lab, Dr. Blackmon was the co-author of the “SCORM Best Practices Guide for Content Developers,” and co- leader of a successful series of workshops on the basics of SCORM for instructional designers. These activities have given Dr. Blackmon extensive experience in helping instructional designers who are new to SCORM address issues related to the size of SCOs and ways to reconcile the needs of individual learners with the use of reusable content.

Dr. Blackmon earned his PhD in Civil Engineering from Carnegie Mellon University and his BA from Duke University.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*

This page intentionally left blank.

Carnegie Mellon University *© Copyright 2006, Carnegie Mellon University, Some Rights Reserved*