Enhanced Tracking and Shared Data for SCORM 2.0

By John Campbell and Don Holmes

Some of the biggest challenges we at Imedia.it have faced since starting to develop SCORM content have been related to run-time data.   SCORM has not been designed to focus on the retention and sharing of learner interaction and result data. In fact, the current SCORM design has indirectly discouraged designs that require this type of data use. We are taught to design content in isolated chunks that have no need of outside data and leave little trace of its existence after its run-time life-cycle is completed. However, in reality, there are many reasons for additional tracking mechanisms and the use of shared data in courseware.

Learner Interactions

Both SCORM 1.2 and 2004 have access to the cmi.interactions data model. This data model is mainly used for tracking how the learner responds to given questions and/or situations. The data model can be used for standard true/false, fill in the blank, or multiple choice types of questions, as well as more complicated task-based or custom situations. The data model does have some drawbacks however. First, a SCO is limited to 250 interactions. A system may allow more than 250 interactions, but it's not guaranteed to support more than this limited amount. Second, interactions are not guaranteed to be persistent beyond the life-cycle of a SCO. Third, the cmi.interactions data model is not scoped beyond a single SCO. Finally, there is little support for extracting and reporting cmi.interaction data in LMSes today.   All these items make it difficult to the detailed level of tracking that we desire today.

In SCORM 1.2, we typically had to create each course as one monolithic SCO. We routinely were required to keep track of more than 250 items. In order to track these learner interactions, we used an external database. This database allowed us great flexibility in tracking, sharing the data across the entire course, and reporting; however, it lacked in portability. If the LMS did not run ASP and have access to a Microsoft SQL Server database, the course would not function.

Another important function of learner interaction data is for use in courseware validation. Some clients, like the US Army, require the use of cmi.interactions for this purpose. However, by design, this data may not be available when validation occurs. As multiple attempts are made on SCOs, prior interaction data may be deleted by the LMS. Some major LMSes do not even retain this data after the SCO exits normally. This interaction data is critical for use in validation. One may argue that this process of retaining interaction data should be a reporting feature of the LMS and not a responsibility of SCORM. This may be; however, in order to be truly interoperable, a very well-defined, consistent approach to data persistence is needed. We feel that there are enough use cases to warrant a stronger and more persistent data model.

Shared Data

Shared data is effectively non-existent today with SCORM 2004 3rd Edition. The only shared object is a global objective which can carry a passed/failed value and a numeric value between -1 and 1. IMS Shared State Persistence (SSP) was once thought to be a solution in this area, but it doesn't seem to fit with the direction of SCORM at this time.

So, the question arises "Why do we need shared data?" It would seem the concept of shared data goes against some of the ideals that originally went into the design of SCORM. We have been told that content should be created in re-usable, independent chunks that can be shared across multiple courseware environments. This content should also be blind to the environment and the context in which it is presented to the learner.   This approach may work well in many cases, but in others, more context within the content might be more beneficial. The more context a designer or developer adds, the more relevant and personalized content can be for the learner. If designed correctly, one need not even sacrifice re-usability in the process. In fact, the use of shared data, in many cases, will encourage the use of more granular learning objects. For example, with the current SCORM design, two objects that could potentially make use of shared data currently would have to be combined into a single SCO to share that data. However, with a new model, these same objects could easily be separated into what may be a more natural organization and SCORM Sequencing could be used to move between them. This allows us to develop objects in parallel and test the sequencing early on through the use of SCORM skeletons. We find that in addition to facilitating reuse, this granularity in our learning objects reduces the time and cost to produce content.

There are many use-cases that require shared data:

- **Learner Journal or Notebook** - We often have the desire (and sometimes the requirement) to give the learner a scratchpad area to take notes on as they work their way through courses. For example, one course we have developed has multiple scenarios. Each scenario consists of multiple lengthy interviews, followed by a test. In the test, you are required to use information gathered during the interview process to answer questions. Having notes stored on the LMS allows the learner to always have them nearby and not needing to worry about losing papers, notes, etc. However, currently, the entire scenario has to be a SCO in order to share this data (using cmi.suspend_data). Shared data in SCORM would allow the course to be broken up into a more natural set of objects, allowing the learner to have access to this journal in any SCO that it was needed.

- **Build on Prior Work** - Many times we have had clients as us if we can produce exercises that build upon work done in prior sections of a course. This allows the learner to see the impacts of prior decisions and provide opportunities to problem solve and to find workarounds and alternate solutions. If the exercise is broken up into multiple SCOs, this becomes impossible. SCORM needs a way to store and retrieve large amounts of data that is shared across SCOs.

- **Personalization** - With shared data, course-ware can easily be personalized to refer to the learner's name, rank, nickname, locale, etc. SCORM currently provides access to only the learner's name. We have found that learners enjoy hearing personalized messages like "Yes, Ma'am" or "No, Captain". Simple personalization can be done using overloaded global objectives, but in order to provide a richer set of customizations, shared data would be desirable.

- **Real-time remediation** - It would be great if the SCO could have access to attempt information (current and historical) from other SCOs. This would allow for immediate, specific, feedback for tasks that are continually being failed. For example, content could give feedback like "You missed this in the pre-assessment as well. Let's take a closer look at the following information." Or "You have failed to successfully perform this task three times in a row. Review the following material before proceeding". Or the remediation could be more subtle, allowing the entire scenario to adapt based on the learner's prior success (or lack thereof). The possibilities are endless.

- **Peer to Peer Communication** - As SCORM provides support for team training (hopefully a topic of other papers for SCORM 2.0), sharing data between teammates will be a critical feature that must be supported. A rudimentary, but effective use case would be a simple system to allow messages to be passed between learners or teammates.   Shared data would allow content to add this and other richer sets of features to training.

Summary

We have identified the following use cases and needs.

- More robust and persistent way to track learner interactions
- Better reporting requirements and mechanisms for extracting interaction data
- An interoperable way to share data between learning objects

The proposed ideas expressed above may seem better suited as modifications to the current SCORM 2004 architecture, versus revolutionary steps for the SCORM 2.0. However, these features will be needed in the future and don't seem to be currently on the radar for SCORM 2004. We hope that SCORM 2.0 is a revolution, but that it also continues to support the ongoing needs of today's customers.