Exploring SCORM and the national flexible learning toolboxes

Ralph Wirski and Grame Brownfield
eworks

Ron Oliver
Edith Cowan University

This paper discusses a project which used investigated use of the SCORM model within the design and development process of high quality learning materials to explore its capability for supporting the reusability and interoperability of digital resources within the Australian VET sector. The paper describes SCORM and its potential advantages and discusses its retrospective implementation to the redevelopment of learning materials in the National Flexible Toolbox Project. The paper describes the process of applying SCORM to the development of an inquiry based set of learning resources and discusses implications for the design of reusable quality e-learning resources and learning objects using the SCORM model.

Keywords: learning objects, learning designs, reusable digital resources, interoperability

Introduction

Developers of elearning resources and materials are aware of the potential of learning objects. The idea is that if online resources and materials are designed carefully they can be reused in other settings in cost effective ways. But barriers can present themselves when people reuse learning materials, so not much reuse ever tends to occur. Problems limiting reuse include:

- the materials not being developed in ways which facilitate separation into smaller parts;
- the original learning context being too strong to support use in other settings;
- other teachers being unaware of the materials;
- other teachers being unaware of the prospect of reusing the materials; and
- difficulties associated with combining materials developed for use in one setting with materials developed elsewhere.

The prospect of the reusability of digital resources is very enticing for many stakeholders in education and despite the difficulties, interest and enthusiasm still abounds for this. In fact there has been a huge amount of work undertaken by a number of large organisations and groups to facilitate the reusability and interoperability of digital learning resources and this work has appeared to have removed many of the barriers which previously limited reuse of learning resources. The work being done to develop the Sharable Content Object Reference Model (SCORM) is a strong case in point. SCORM has been developed by the Advanced Distributed Learning (ADL) initiative and provides a design and development model for learning resources which strongly supports reusability and interoperability.

What is SCORM?

SCORM is a model that describes a standardised way to design and develop learning materials so that learners' pathways and successes in the learning setting can be tracked and monitored. At the same time SCORM supports an approach which facilitates the reuse and interoperability of compliant resources. The learning materials themselves (learning objects) are managed and coordinated within a compliant learning management system (LMS) such as WebCT or Blackboard. And SCORM also supports ways for the learning materials to be discovered and accessed for re-use. All in all it proposes a powerful solution to the problems facing those wanting to reuse learning resources.

How does SCORM support these capabilities? The answer is in the way it has been developed. SCORM (the model) describes two main elements: a content aggregation model (CAM) and a run time environment. The CAM describes the ways in which SCORM materials are organised, described, and packaged so that they can be seamlessly exchanged between different learning systems. The run time environment provides the means for the learning materials to communicate with the LMS and for the collection of data to track and monitor learners.

SCORM content aggregation model

The SCORM CAM describes 4 main elements.

1. It uses **learning object metadata descriptors** as a means of describing the contents of the materials. Metadata describes each learning object by indicating the content, ownership, costs for use (if any), the technical requirements, and educational purposes, etc.
2. The **metadata XML binding** defines how the metadata tags can be expressed in XML to make them readable by the technology and people.
3. CAM also describes **content packaging**, how the materials will be packaged and described. Materials are linked together with an XML manifest that defines the contents in the materials and how they relate to one another.
4. The fourth element in CAM describes **content structure**, a description that indicates the structure of the learning setting. The content structure defines the intended relationships of the content. It is read and used by the LMS when a package is imported to determine the organisation of the materials. This element of the CAM is where the learning design is recorded to enable reuse of the materials to follow what was intended by the original designer.

SCORM run time environment

Within SCORM, a standardised way is presented for the learning materials to send and receive information between the learner and the LMS. The run time environment consists of an **Application Program Interface** (API) which provides a consistent and standard way of communications between the materials and the LMS, irrespective of the ways in which the content was developed. Within the run time environment, there is a set of data elements, **a data model**, which can be transferred through the API. This dataset comprises records of the learner's action and progress, for example, scores in tests, times spent in sections and levels of mastery achieved. This data model enables the LMS to track learners' progress.

SCORM technical implementation

SCORM packaging adheres strictly to the IMS Content Packaging Specification but provides additional explicit implementation guidance for packaging digital learning resources. The Content Packaging specification component deals with Assets, Sharable Content Objects (SCOs) and Content Aggregation Packages. While SCOs are collections of Assets, Assets may be single individual objects such as media or HTML pages, or collections of such assets.

Assets and SCOs are intended for reuse and are the building blocks of a Content Aggregation Package. They should be independent of learning context and intended to be subjectively small units, such that potential reuse across multiple learning objectives is feasible. The SCORM does not impose any particular constraints on the exact size of a SCO.

Given its characteristics, the term SCO is often used to mean learning object. There is not a universally accepted definition for learning objects, however there are a number of common themes.

- Learning objects should be of a size of that can be reused within different learning contexts.
- They should be in a format that can be deliverable, such as digital entities.
- Learning objects can provide an educational experience for some pedagogical purpose.

Within the confines of this paper we refer to a learning object as "any digital resource that can be reused to support learning." (Wiley, 2000). This definition includes anything that can be delivered to a learner, be it large or small.

The adoption of the SCORM standard across the VET sector appears to offer the possibility to create, store, and disseminate reusable LOs in the form of SCOs. These SCOs could provide content developers and teachers access to existing learning content via digital repositories.

In this context Content Aggregation Packages comprise one or more SCOs or assets, that is one or more learning objects. They need to be structured in such a way that they are ready for delivery to a learner.

This implies that Content Packages are not necessarily as reusable as SCOs, because they are associated with specific learning context information. They can still, however, be delivered via a SCORM compliant Learning Management System.

National flexible learning toolboxes

We have previously demonstrated at ASCILITE a number of the products and processes associated with the design and development of the National Flexible Toolboxes. In 2003 we demonstrated a digital repository containing resources from these products and we discussed some of the limitations that we experienced building a repository using learning materials developed using conventional HTML Web linking (Brownfield & Oliver, 2003). These limitations included:

- access problems resulting from the limited scope of the metadata for many resources;
- variations in the accuracy and integrity of the metadata for many resources;
- the prevalence of similar and equivalent metadata across many resources failing to distinguish between them adequately;
- having discovered useful resources, the system didn't provide the means to discover and access related resources
- difficulties associated with weighting and ordering of search results to return the best resources for any search;
- difficulties in reuse as a consequence of the contextual nature of many resources; and
- interoperability of the repository with other repositories.

Having discovered these limitations, the team gave serious thought to exploring the prospect of increasing the reusability and interoperability of the future Toolbox products by using SCORM. The team was aware that while SCORM promised great opportunities for reusability and interoperability, the model had been developed primarily for learning environments based on linear and didactic learning designs.

One of the proponents and designers of SCORM, Dan Rehak, had commented earlier about this and indicated,

> SCORM is essentially about a single learner, self paced and self directed. It has a limited pedagogical model unsuited for some environments." This is mainly a consequence of the needs of the main initiators of SCORM: the US Federal Government in general, and the Department of Defence in particular. Their needs are mainly in the area of training for specific systems and situations by people who are not generally in full time education. This need is addressed very well by the spec, but "SCORM has nothing in it about collaboration. This makes it inappropriate for use in HE and K-12. (Kraan & Wilson, 2002).

The Toolboxes have all tended to use contemporary learning designs based on knowledge construction (Oliver & Blanksby, 2003) and the comments by Dan Rehak were suggestive that these resources might be difficult to build with SCORM. As a trial activity to judge how SCORM might be applied in the design and development of Toolboxes, an activity was undertaken to produce a SCORM conformant Content Package using an existing Toolbox, The Grange Care Services. The purpose of this activity was to test the processes by which Toolboxes might be designed and developed using SCORM

Content packaging an existing toolbox

A sample of the Grange Care Services Toolbox is available via the Internet at http://flexiblelearning.net.au/toolbox/series6/602.htm. The first noticeable aspect of the Grange Care Services Toolbox is the front page which is a flash animation (Figure 1). The flash animation allows the learner to enter either the home care or residential services components of the toolbox. Here, we immediately discovered the impact of the SCORM rule that disallows navigation between SCOs. To adhere to this SCORM rule and keep the flash animation, one would have to treat the entire Toolbox as one SCO. This is not only inelegant but defeats the purpose of SCORM: creating reusable learning objects. To produce a SCORM conformant Toolbox, we could remove the navigation contained within the Flash object and transfer the navigation to a manifest file that accesses separate SCOs.

Of the two available areas in the toolbox, we chose the Grange Home Care area to convert to a SCORM compliant Content Package. The Grange Home Care area consists of duty statements that cover each day of the week. Each day can have more than one duty statement, while each duty statement can have more than one task. This area also contains a resource library that contains: tests, links to manuals, policies and procedures in the form of word documents, and areas for discussion.

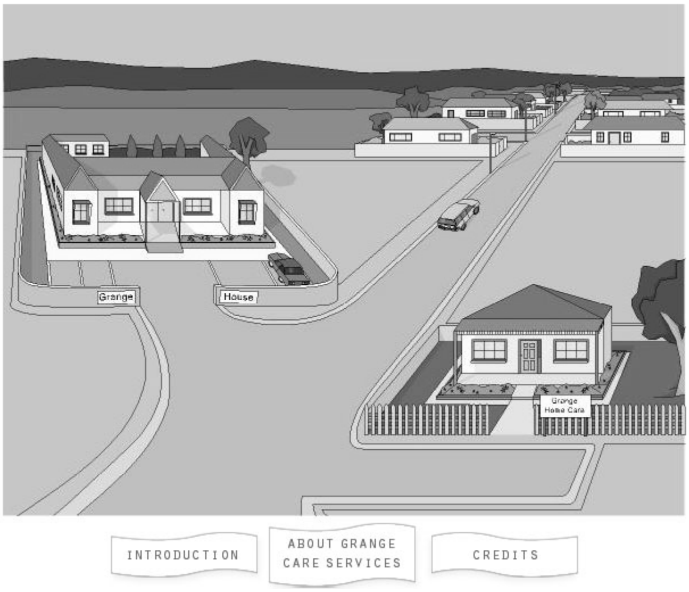


Figure 1: Grange Care Services front page

Before building the Content Package, we had to decide which html pages would be SCOs and which html pages would be assets. This meant choosing SCOs (i.e. learning objects) of a size and content that would make them optimally reusable and meaningful. According to the SCORM 1.2 standard, a SCO can be of any size and scope, as long as it is described by a resource element in a Content Package manifest file. Given the flexibility of SCORM, the onus of choosing the correct SCO size falls on the content developer.

The SCORM documentation does not provide any mandatory methods of deciding which learning resources are SCOs or assets. However, according to the SCORM, SCOs are intended to be subjectively small units, such that potential reuse across multiple learning objectives is feasible. When determining whether to make a resource a SCO or an asset, the content developer should consider the amount of material required to achieve the learning outcome, and the level of reuse to be obtained.

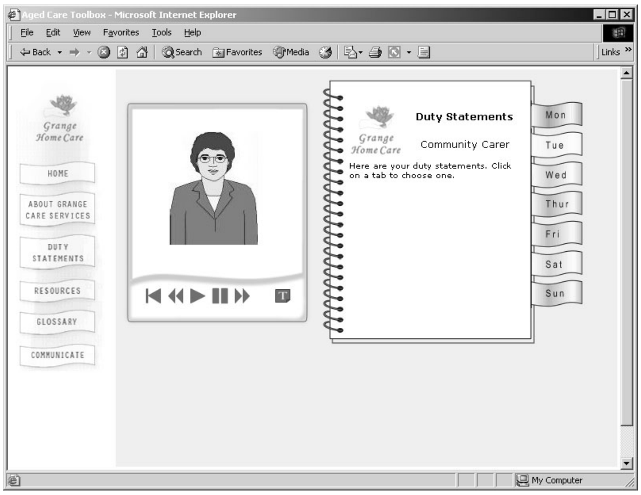
Any learning resource that is deemed a SCO must contain the minimum SCORM API calls to locate an LMS's API Adapter. A Content Aggregation provides content structure used to aggregate learning resources into a cohesive unit of instruction. The Content Aggregation provides the sequence in which learning resources are to be presented to the user. A Content Aggregation is often interchanged with the term Content Package. Technically, a Content Package is SCORM's mechanism for binding a Content Aggregation and metadata. A Resource Package is a Content Package without an organisation, in other words a Content Package that does not specify a sequence for the learning resources.

SCORM created SCOs must communicate with the LMS run time service using the LMSInitiliase and LMSFinish functions. This is mandatory SCORM behaviour. It is optional for SCOs to communicate with the LMS while being displayed using functions that get and set values. Any SCORM compliant LMS must handle these functions.

Display and navigation

Toolbox developers using the SCORM standard cannot have any control on how learning resources are launched. This means that while the content itself will be controlled by the Toolbox developer, there will be no control over the size of the window that displays that content. Further, content developers can not make any assumptions about the color scheme or layout of the surrounding windows (or frames).

Figure 2 shows the navigation layout of Grange Home Care before conversion to the SCORM standard. We can see the navigation buttons on the left hand side. When moving to the SCORM standard, all hard coded navigation will have to be removed from our HTML pages. This ensures we do not include any links between SCOs.


Aged Care Toolbox - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Links
Grange Home Care
HOME
ABOUT GRANGE CARE SERVICES
DUTY STATEMENTS
RESOURCES
GLOSSARY
COMMUNICATE
T
Mon
Tue
Wed
Thur
Fri
Sat
Sun
My Computer

Duty Statements

Community Carer

Here are your duty statements. Click on a tab to choose one.

Figure 2: Original Grange Home Care layout and navigation

A SCO may not assume that it will run in a top level window, or attempt to force itself to run in a top level window. This will have implications for many of the already produced toolboxes as they rely on having control of all surrounding frames for navigation. The idea behind this thinking is that a SCO can be launched without any prior knowledge of its surroundings.

Although the use of pop up windows is allowed within SCORM compliant SCOs, this behavior is discouraged by SCORM best practice (Ostyn 2001). SCOs are responsible for closing any pop up windows that are opened, before a SCO is unloaded. A SCO is not allowed to leave any trace of itself in the user’s environment window after it has been unloaded. This implies that SCO developers using pop windows would have to write extra code to close all pop ups. This requirement, which would most likely be implemented in the form of client side scripts, could create compatibility problems.

According to SCORM, the general for creating manifests is that a package always contains a single top level manifest. This implies that a manifest file should be created for a stand alone SCO. However, when that SCO is part of a Content Package, a manifest file for the SCO is not necessary, since the SCO will be described within the top level manifest. There are still advantages to providing a manifest file for every SCO within a SCORM Content Package:

- Aids in easy aggregation.
- Provides clarity in recognising the overall structure of a Content Package without having to view the manifest.
- When separated from a content package, the SCO will still maintain its context.
- Removes the requirement to derive a manifest when a SCO is disaggregated from a Content Package.

SCO options for Grange Home Care

Figure 3 shows a high level summary of the Grange Care Services proof of concept. Note that the term “tasks” used in the diagram does not imply any meaning – just refers to tasks in each duty statement.

Option 1: All encompassing SCO

The layout shows that there are a number of ways to create SCORM conformant content. For example, it is possible to determine that the Grange Home Care area is one big learning object, and all other resources are assets. This way, the manifest file will contain the index html page of Grange Home Care as the only SCO. While this approach may pass the SCORM conformance test (and it does - we checked), it does not provide a learning object of a suitable size for reusability.

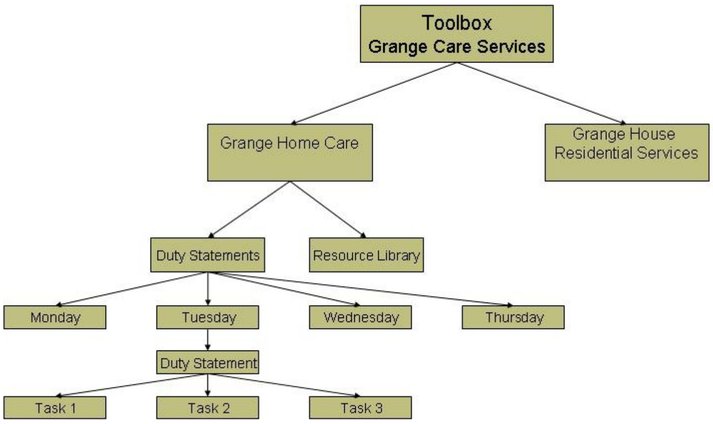


Figure 3: Grange Care Services toolbox layout

Option 2: Sub manifests

Another approach to building SCORM conformant content can be to produce small content packages and combine those to produce a larger content package. Within the Grange Care example each duty statement, as shown in Figure 3, can be considered a content package containing individual tasks as SCOs. Each day of the week could also be a content package containing sub-manifests that define duty statements. This idea could carry on recursively, incorporating duty statements as a content package, until eventually we encompass all learning content.

From a theoretical point of view, this is an ideal solution. By creating content packages at a low level, we have learning objects that are readily reusable, and because they are a content package they should be also be meaningful. Using the smaller content packages we create a larger content package that can be disaggregated by future users. Unfortunately, under the current definition of SCORM (1.2), the implementation for sub-manifest aggregation is awkward at best. While the problems are not explicitly documented in SCORM documentation, some include:

- No clear consensus, or support, by IMS or SCORM on how to reference external manifests.
- Identifier conflicts. As you join sub-manifests into another manifest, you are likely to run into identifier conflicts.
- Package component file name conflicts.
- Aggregation and disaggregation operations of manifests are outside the scope of the SCORM specification. SCORM only specifies the syntax of the final manifest.

The efficient and obvious method for joining manifests would be if the top level manifest referred to a sub-manifest, in the same manner as a top level manifest refers to SCOs or assets. Unfortunately, under SCORM 1.2 and its underlying IMS Content Packaging Specification, one has to copy all sub-manifest information to the parent manifest. This requirement is counter productive as we have to produce a manifest file more than once. To get the same effect as aggregating sub-manifests, why not produce only one manifest that contains a number of nested SCOs? This has the same effect as combining several sub-manifests.

Option 3: Multiple SCOs

The solution we chose for Grange Care was to create multiple SCOs and join them using the organisation section of the manifest. Figure 4 shows the division of SCOs for our Content Package. At the lowest level we created a SCO for duty statements. This SCO contains assets that define how to complete a particular duty comprising sequential tasks. This represents the smallest reusable object. We used an individual duty statement, together with others, to form different sets of duty statements, each for a specific set of circumstances.

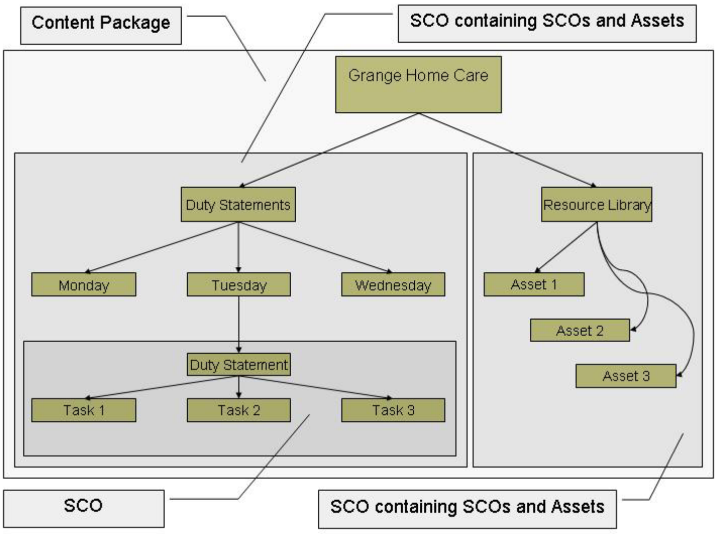


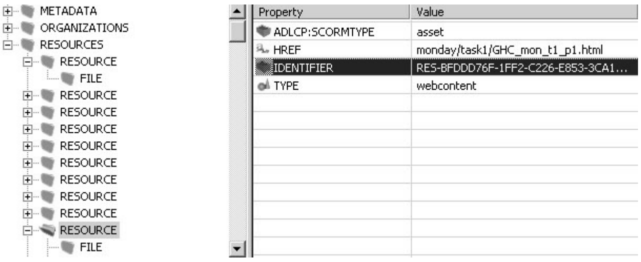
Figure 4: Multiple SCO layout

We had a choice to make, in that, should each day of the week be SCO? Given that a learning object should be reusable; would a SCO for each day of the week be reusable? For example, a collection of duty statements for Monday is only relevant to the set of circumstances for Monday. Given this, you would not expect an SCO for a particular day to be meaningful within a different set of circumstances and therefore reusable. Therefore we have not created SCOs for each day of the week.

The top level SCO is at the duty statements level. At this level, we know the circumstances for choosing duty statements and the context of those duty statements. For example, if certain tasks are chosen for Monday, at the top level SCO we can also see the tasks chosen for other days, which provides context.

To describe our content package we needed to create the top level IMSmanifest.XML file. There are three sections to this manifest: Metadata; Organisation; and Resource. Within the resources section we described SCOs and assets that comprise our content package. The navigation is defined in the organisation section.

Figure 5 shows the organisation section of our manifest within an XML editor. You can see that the SCO layout from Figure 4 matches the organisation section in the manifest. Figure 5 shows a subset of the resources sections within an XML editor.



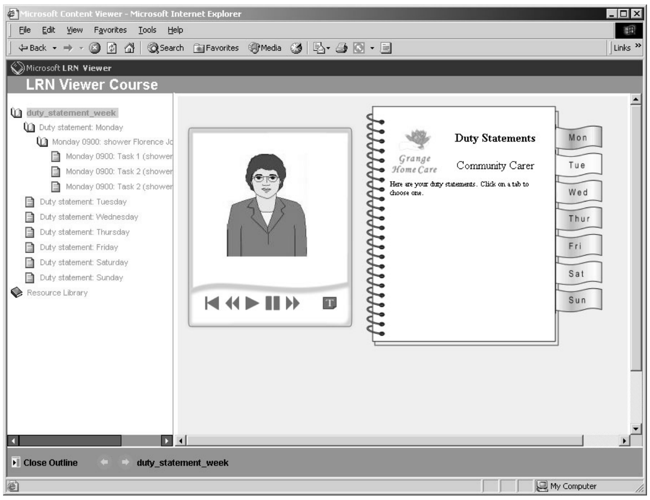
```
⊞ METADATA
⊞ ORGANIZATIONS
⊟ RESOURCES
    ⊟ RESOURCE
        FILE
    ⊞ RESOURCE
    ⊞ RESOURCE
    ⊞ RESOURCE
    ⊞ RESOURCE
    ⊞ RESOURCE
    ⊞ RESOURCE
    ⊞ RESOURCE
    ⊞ RESOURCE
    ⊟ RESOURCE
        FILE
```

Property	Value
ADLCP:SCORMTYPE	asset
HREF	monday/task1/GHC_mon_t1_p1.html
IDENTIFIER	RES-BFDDD76F-1FF2-C226-E853-3CA1...
TYPE	webcontent

Figure 5: Manifest resources section

Viewing SCORM content

SCORM content can be viewed within a SCORM LMS system. However, companies such as Microsoft or Reload provide applications that allow the user to view SCORM content. These applications are referred to by the following common (and interchangeable) names: LMS wrapper; LMS viewer; or LMS player. Figure 6 shows the SCORM version of Grange Home Care and how it looks within Microsoft’s LRN Viewer.



Microsoft Content Viewer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links
Microsoft LRN Viewer
LRN Viewer Course
Mon
Tue
Wed
Thur
Fri
Sat
Sun
T
Close Outline
My Computer

duty_statement_week
Duty statement: Monday
Monday 0900: shower Florence Jc
Monday 0900: Task 1 (shower
Monday 0900: Task 2 (shower
Monday 0900: Task 2 (shower
Duty statement: Tuesday
Duty statement: Wednesday
Duty statement: Thursday
Duty statement: Friday
Duty statement: Saturday
Duty statement: Sunday
Resource Library

Grange Home Care

Duty Statements

Community Carer

Here are your duty statements. Click on a tab to choose one.

duty_statement_week

Figure 6: Grange Home Care within a SCORM viewer

Sequencing content

SCORM compliant resources by themselves are inarticulate, in that without an LMS they will be simply a collection of resources, and not a course in a format easily traversable by the learner. Practically, this induces the need for an LMS or an LMS viewer that will produce a visual tree, menu or a set of nested menu, which allows the user to traverse between SCOs and Assets.

Technically, SCORM SCOs are joined together by defining items in the organisation section of the Content Package manifest file. The SCORM restricts sequencing embedded in content to exist only within SCOs – that is content from one SCO may not refer to content in another SCO. Sequencing between SCOs must he handled by an LMS and defined in the manifest file.

The LMS will control all SCO to SCO navigation. The order in which SCOs are presented to the learner is provided by the organisation section of the Content Package manifest. Typically an LMS will provide both a tree structure, and forward and next buttons for navigation. This means that a sequence of content will be presented and available, but the learner will be able to choose which learning objects to view and in what order.

While restrictive, there are several advantages to this approach:

- The order of resources can be easily changed because there is no hard coding of sequencing. The order can be changed using a package such as reload, or manually within the manifest file.
- Content can be easily disaggregated.
- Reduction of problems arising from proprietary and often innovative sequencing implementations utilising frame sets and java scripts.

The major disadvantage of content sequencing under SCORM 1.2 is what is referred to as the "*glossary problem*". The *glossary problem* describes the occurrence where a common resource is used throughout a Toolbox. Frequently, this common resource is a glossary.

In current Toolbox development, if a glossary is required, it is created only once, and all content within a Toolbox that refers to information from a glossary, will refer to a single copy of the glossary. However, under SCORM 1.2, you will need a copy of the glossary for each SCO you produce. This could get quite annoying for Toolboxes that lend themselves to reusability and foster large numbers of SCOs. The *glossary problem* creates two further problems:

- Content size increase in terms of storage space.
- Maintenance: the need to update each existing copy of the glossary.

SCORM and the dependency element

Within SCORM 1.2 there is no clear accepted solution to the *glossary problem*. We can however come close to providing a satisfactory method of creating multiple SCOs referring to the same glossary. We elucidate the problem by using the dependency element that is allowed within SCORM 1.2.

The SCORM documentation defines the dependency element as one that "contains a reference to a single resource that can act as a container for multiple files that resources may be dependent on". Translated into technical terms, this means that you can:

- Create an asset, such as a glossary, containing content that other SCOs depend on.
- Every SCO that wants to refer to content from the glossary, must declare the glossary asset as a dependency.

We created an example Content Package where many SCOs refer to content within a single glossary asset. This Content Package has been tested and passed the SCORM ADLCP-PIF1 conformance test.

Figure 7 illustrates the layout of a Content Package where several different SCOs can access content from a single glossary asset.

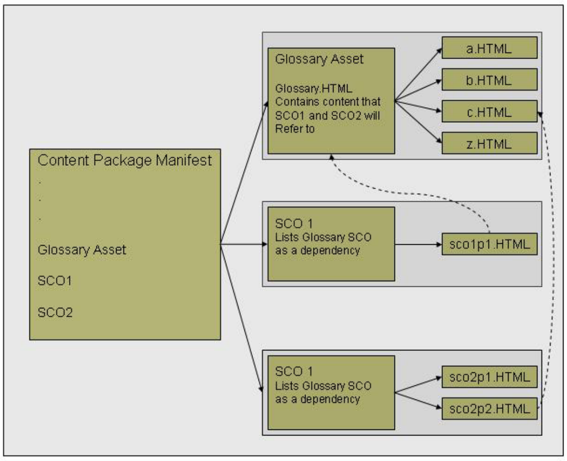


Figure 7: Content package containing a glossary SCO

The first step was to create the glossary. Due to the technical implementation of SCORM we did not define the glossary as a SCO, but as an asset that contains a number of resources, namely html pages containing definitions. The glossary asset can be used by other SCOs. Unless we utilised the more elaborate elements of SCORM there was no disadvantage to treating the glossary as an Asset. The telling difference between a SCO and asset is that a LMS is not able to track an asset.

Conclusions

The activity to establish whether Toolboxes could be made SCORM conformant appeared to be successful. We tested our Content Package using the ADL Test Suite, Reload and the LRN Viewer. The end product was a conformant product which still retained the student centred learning design and original functionality. The outcome added weight to the general thinking that the planned Series 7 of the Toolboxes could be developed to be SCORM conformant. The recommendations for this plan coming from this project were:

- Content developers should produce SCORM conformant Learning Objects but not necessarily SCORM conformant Toolboxes. The current version of SCORM has some limitations and short comings: for example, the technical implementation of the aggregation and disaggregation of packages is not yet ideal. The current limitations on SCO sequencing and navigations are enhanced in SCORM 2004.
- SCORM conformance is not difficult to reach. The major components required are: standard and readily available API functions applied to SCOs; metadata used to describe the learning resources;

and a well formed manifest file. Learning objects should reach the minimum conformance levels SCO-RTE1: minimum conformance for SCOs; ADLCP-PIF1: default for conformant Content Packages.
- Gaining a higher SCORM conformance will not increase the reusability and interoperability of a resource.
- Using the ADL SCORM test suite should be adopted by toolbox developers who intend to produce reusable and SCORM conformant content. To date, the ADL test suite is the only automated and common method developers have of determining if their content is SCORM conformant
- The glossary problem can be alleviated within a content package by using the dependency element

The products from the Series 7 National Flexible Toolbox project are all nearing completion and when this paper is presented at the Conference, there will be some examples of the SCORM compliant resources to show to delegates to further demonstrate the success of the project.

References

Advanced Distributed Learning (ADL) (2002). Sharable Content Object Reference Model (SCORM®) 1.2 Overview. [verified 28 Oct 2004, menu] http://www.adlnet.org/index.cfm?fuseaction=rcdetails&libid=40&filterid=24&page=1&keywords=&applyto=libTitle,libAuthor,contentText

Advanced Distributed Learning (ADL) (2004). Sharable Content Object Reference Model (SCORM®) 2004 Overview. [verified 28 Oct 2004, menu] http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false

Brownfield, G. & Oliver, R. (2003). Factors influencing the discovery and reusability of digital resources for teaching and learning. *Interact, Integrate, Impact: Proceedings 20th ASCILITE Conference* (pp 74-83). Adelaide. http://www.ascilite.org.au/conferences/adelaide03/docs/pdf/74.pdf

Carnegie Mellon (2003). Best Practices Guide for Developers. [verified 28 Oct 2004] http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/index.html

IMS (2003). *IMS Content Packaging Best Practice Guide*, Version 1.1.3 Final Specification, June. http://www.imsglobal.org/content/packaging/index.cfm

Kraan, W, and Wilson, S. (2002). Dan Rehak: "SCORM is not for everyone". CETIS Discussion Article, CETIS, October. [verified 28 Oct 2004] http://www.cetis.ac.uk/content/20021002000737

Oliver, R. & Blanksby, V. (2003). Online learning designs in the training sector. *Interact, Integrate, Impact: Proceedings 20th ASCILITE Conference* (pp 364-374). Adelaide. http://www.ascilite.org.au/conferences/adelaide03/docs/pdf/364.pdf

Ostyn, C. (2001). Cooking up a Scorm. http://home.click2learn.com/standardswork/ [not found 28 Oct 2004; see http://academiaelearning.com/contenido/scorm/cooking/i_cookbook.htm]

Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects: Online Version*. [verified 28 Oct 2004] http://reusability.org/read/chapters/wiley.doc

Zhen Yin, Zhengfang Xu and Abdulmotaleb El Saddik (2003). Study of metadata for advanced multimedia learning objects. *CCGEI, 2003 Proceedings*.

Ralph Wirski can be contacted on ralph.wirski@eworks.edu.au
Grame Brownfield can be contacted on graeme.brownfield@eworks.edu.au
Ron Oliver an be contacted at r.oliver@ecu.edu.au

Please cite as: Wirski, R., Brownfield, G. & Oliver, R. (2004). Exploring SCORM and the national flexible learning toolboxes. In R. Atkinson, C. McBeath, D. Jonas-Dwyer & R. Phillips (Eds), *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference* (pp.938-947). Perth, 5-8 December. http://www.ascilite.org.au/conferences/perth04/procs/wirski.html