DYNAMIC AUGMENTATION OF SCORM PRE-AUTHORED COURSE MATERIALS WITH ADAPTIVE LINKS TO SUPPLEMENTARY RESOURCES

Nor Aniza Abdullah
University of Malaya
Faculty of Computer Science and Information Technology
noraniza@um.edu.my

Hugh C. Davis
University of Southampton
School of Electronics and Computer Science
hcd@ecs.soton.ac.uk

ABSTRACT

In a SCORM compliant courseware, content and the pedagogic approach to be taken are predefined by the course author. As a consequence, users are unable to learn according to their preferences, and implicitly they will all encounter the same learning experience. Recent attempts to personalise learning in SCORM often resulted in either modifying or substituting SCORM specification elements. Nonetheless, SCORM is a widely-used solution to interoperability problems. For this reason, this work focuses on supplementing SCORM rather than redefining it. This is accomplished by dynamically associating each pre-authored learning material in a SCORM package with adaptive links to relevant supplementary resources upon delivery. As a result, we have developed a Personalised Link Service (PLS) to deliver these links into user's SCORM-compliant learning environment. In this paper, we present the design of a PLS authoring architecture which enables the automatic generation of a concept map from a SCORM package and consequently links to alternative learning resources, and a PLS run-time service oriented architecture which delivers these alternative resources, alongside the SCORM defined resources, according to a user model. We demonstrate the feasibility of our architecture by implementing a service to support a simple notion of a user model (Preferred Learning style).

KEYWORDS

SCORM, Adaptive links, Personalised Link Service, Auld Linky, e-learning

1. INTRODUCTION

As the result of recent advances in the business of e-learning there has been a growing interest in e-learning standards, particularly SCORM (Sharable Content Object Reference Model) (SCORM 2004). SCORM is a reference model that integrates a collection of e-learning resource standards and specifications. In a SCORM compliant courseware, course content and the pedagogic approach to be taken are predefined by the course author. As a consequence, users are unable to learn according to their preferences, and implicitly they will all encounter the same learning experience.

Recent attempts to personalise learning in SCORM often resulted in either modifying SCORM or substituting its course sequencing mechanism with their own adaptive techniques in order to sequence course content adaptively, as demonstrated in (Modritscher et al. 2004; Power et al. 2005). As a result, a consistent sequencing behaviour across different learning management systems would not be accomplished. Nonetheless, SCORM is a widely-used solution to interoperability problems with e-learning resources and can realise not only content sharing and reusability but also a consistent sequencing of course content across different systems and tools. For these reasons, rather then redefining SCORM, this work presents an alternative way to accommodate students' differences without proposing any changes to the existing SCORM learning environment. Therefore in its approach, the existing SCORM's course sequencing mechanism is retained, whilst, the SCORM Run-Time Environment (SCORM RTE) (SCORM RTE 2004) is augmented with personalised links to on-demand additional learning resources based on a student model. As a result, each pre-authored learning material in the SCORM courseware will be delivered accompanied by a list of adaptive links to related supplementary materials. Links to these supplementary materials are pre-selected to suit some aspects of students' models in order to assist their understanding of the primary materials contained in the courseware package (which are predefined by the course author).

This paper presents the design of an authoring architecture which enables the automatic generation of a concept map from a SCORM package and consequently links to alternative learning resources, and a run-time service oriented architecture which delivers these alternative resources, alongside the SCORM defined resources, according to a user model.

The feasibility of both architectures is then demonstrated by the successful implementation of a prototype for a personalisation service based on students' preferred Learning Styles (Abdullah 2006). The choice of Learning Styles as the user factor on which to select alternative materials is largely arbitrary, and it would have been possible to demonstrate other personalisation sources such as accessibility needs (e.g. for learners with learning impaired vision or hearing).

In Section 2 we bring some research work related to ours, followed by the description of our system architecture that constitutes both the authoring and run-time environments. Section 4 reports on the results of our system's evaluation. Finally, section 5 concludes our research work.

2. RELATED WORK

There have been several efforts to provide personalised learning within SCORM learning environment. The commonality feature among these attempts is that they either proposed some changes to the SCORM Content Aggregation Model or disregarded its course sequencing mechanisms which is based on the IMS Simple Sequencing specification (IMS SS 2002). In this section, some of these attempts are reviewed to highlight the differences between their approaches and ours.

In (Mödritscher et al., 2004), changes have been proposed to the concept of the SCORM Content Aggregation Model (CAM). Their proposal suggested that a SCO (Shareable Content Object) should also be permitted to contain other SCOs besides a collection of Assets, and an Asset should be able to contain more than one resource. However, according to the SCORM CAM, a SCO is built from a collection of Assets, and there should be no context-dependent material among SCOs in order to permit reusability and interoperability of the SCOs, and an Asset should not contain another Asset. When compared to this work, PLS does not propose changes to the core elements of SCORM which may disturb SCORM's integrity and its purpose of existence. For instance, allowing Assets to contain more than one resource may result in the inability for the Asset to be reused. Also, by suggesting that SCO should be allowed to contain another SCO may introduce conflicts in the existing communication process between SCOs and the SCORM compliant learning management system.

The work by (Power et al., 2005) disregarded the implementation of SCORM sequencing mechanism which is based on IMS Simple Sequencing specification (IMS SS 2002) by applying curriculum sequencing technique to the problem of generating personalised SCORM-based courses. On the contrary, PLS incorporates personalisation features by offering an independent adaptive link support service into SCORM Run-Time Environment alongside its existing course sequencing mechanism.

3. SYSTEM ARCHITECTURE

This section provides the architecture design of both PLS Authoring Environment and PLS Run-Time Environment. The PLS Authoring Environment is responsible for preparing the input components for the PLS Run-Time Environment. These components are a concept map, representing the domain being taught, and "service linkbase", containing links to alternative resources teaching the same concepts. The innovation of the design is the automation of the process of creating both concept map and linkbase so that system dependency on subject expert can be alleviated. The PLS Run-Time Environment, on the other hand, is responsible to deliver relevant alternative resources into the existing SCORM learning environment, alongside the SCORM defined resources, according to a user model.

3.1 PLS Authoring Architecture

The objective of the PLS Authoring Environment is to automatically deduce a concept map from a SCORM package (SCORM 2004), as illustrated in Figure 2.0. The generated concept names are later used as the reference point to create a “service linkbase” of links to alternative resources.

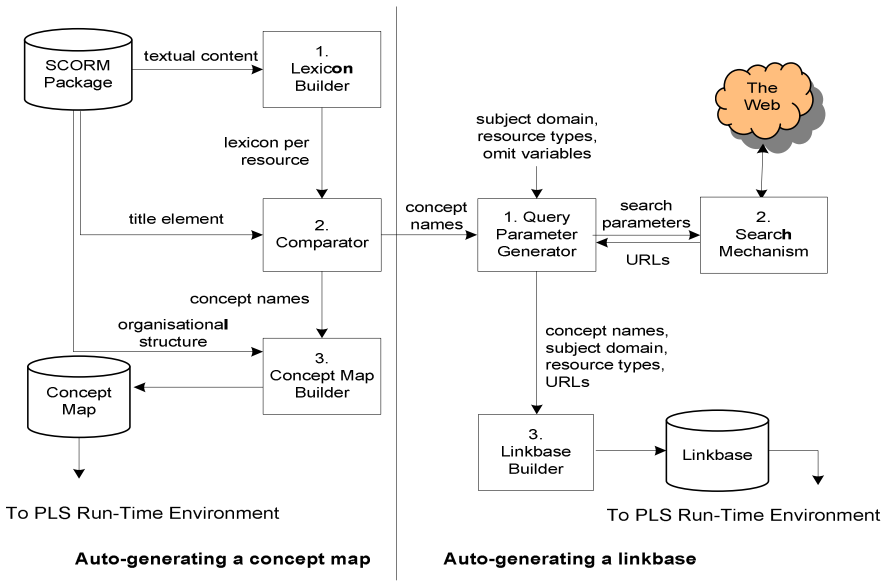
Our concept map is generated automatically based on the information obtained from the SCORM package, particularly the manifest file. In our work we used as an example a SCORM courseware package prepared by the ADL technical team on the subject of “Adobe Photoshop” (ADL Technical Team 2004). The example package (.zip file) contains a collection of learning resources about Adobe Photoshop, a manifest file (named as “imsmanifest.xml”) and other relevant files. The manifest is an XML file that describes both the resources to be used for a particular unit of learning material along with one or more organizations which by default determine the order in which these resources will be presented or sequenced. Technically, the organization contains a collection of activity nodes, each given the title of a particular learning topic, and the leaf nodes are attached to the corresponding resources (SCORM 2004) as shown in Figure 1.0.

Figure 1.0. Fragments on item and resource element found in a manifest file

```
<item identifier="MODULE2">
  <title>Module 2 -- Enhancing Images</title>
     <item identifier="LESSON5" identifierref="RESOURCE_LESSON5">
         <title>Lesson 5 - Color Balance</title>
     </item>
  …
</item>

<resource identifier="RESOURCE_LESSON5" adlcp:scormType="asset"
type="webcontent" href="Lesson5.htm">
  <file href="Lesson5.htm" />
</resource>
```

Figure 2.0. The Architecture of PLS Run-Time Authoring Environment



Our concept map encompasses a list of concepts each with the following attributes: concept name, its associated node id, its parent id and weight of knowledge this concept makes to its parent's concept. The name for each concept is derived by mapping each word in the title element (from the manifest) against a set of lexicons of important terms (learning concepts) within the domain. The lexicons are automatically derived by applying text processing algorithm on every learning resource (html pages) in the SCORM package as shown in the left side of a diagram in Figure 2.

Table 1.0: An auto-generated lexicon obtained from each learning resource in a SCORM package

Resource Id	**Activity Title**	**Lexicon Per Resource**
Lesson 4.htm	Lesson 4 – Layers	layer, image, circle
Lesson 5.htm	Lesson 5 – Color Balance	color, balance, image, layer, original, adjustment
Lesson 6.htm	Lesson 6 – Brightness and Contrast	contrast, brightness, image, layer, adjust, adjustment
Lesson 7.htm	Lesson 7 – Hue and Saturation	hue, saturation, image, layer, color, adjustment
Lesson 8.htm	Lesson 8 – Selection Tools	tool, selection, lasso, select, option

In a package which aims to teach about 'Colour Balance' concept in Photoshop, the lexicon entry might be 'color', 'balance', 'image' etc (as shown in Table 1.0). Prior to determining the concept name for the resource (e.g. Lesson 5.htm), the Comparator will compare this lexicon entry with the title element (e.g. "Lesson 5 – Color Balance") of the associated learning node inside the manifest file. Any matches found will represent the concept name (e.g. Color Balance) for that resource. After the concept names have been confirmed, the relationships among the concept names must be established prior to constructing a concept map. The tree structure of the organisational element of the manifest file permits 'has-a' and 'is-part-of' relationship to be established between the activity nodes. From this information, the following elements can be inferred for the concept map: the activity node id, and its parent node id. When the relationships among the concept names have been identified and the relevant concept map's elements and attributes have been deduced, a concept map can now be generated.

The rest of this section describes our approach in locating alternative resources from the Web and generating the linkbase (refer right side of a diagram in Figure 2). Our linkbase is an XML-file that is composed of associations of links expressed as hyper-structures in FOHM (Fundamental Open Hypermedia Model) (Millard, 2000), and marked up with the concepts that they respond to, and the resource type (e.g. example, experiment, exercise, homework) they represent. Thus the identified links can be semantically organized according to these metadata.

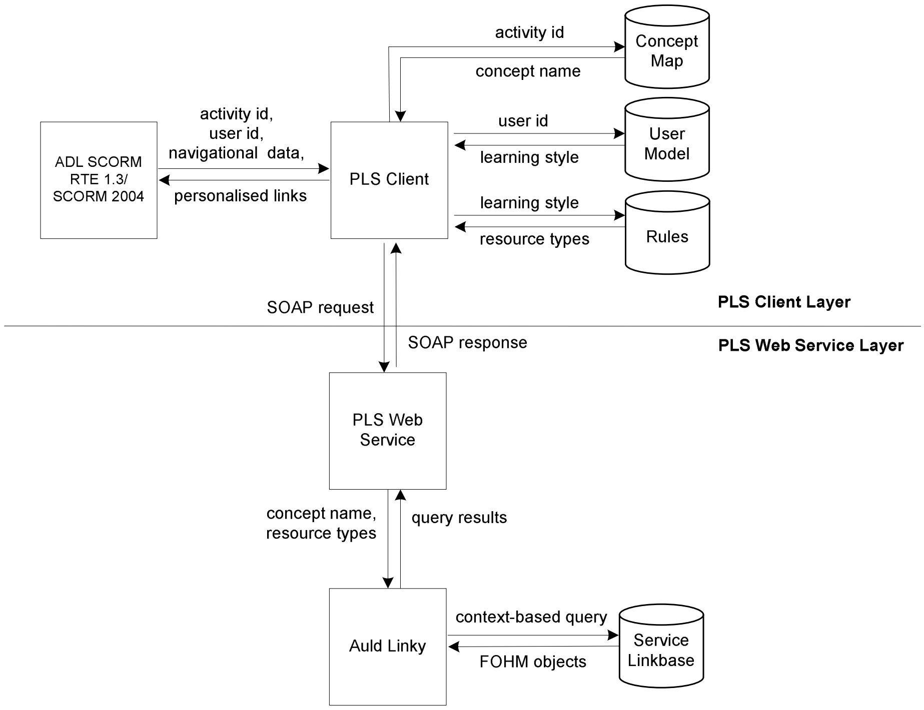
Our first step in this stage is to formulate a query object for Web search engine. The query object consists of the following elements: subject domain (e.g. adobe photoshop), concept name (e.g. color balance), a list of resource types (e.g. example, assignment), and omit-variables (e.g. price, bookstore). The omit_variable allows us to instruct a Web search engine (e.g. Google) to exclude certain words from the search result. A successful search will return a list of corresponding links. These links along with other related information will be processed by the Linkbase Builder prior to constructing a FOHM-structured linkbase.

The choice of FOHM as the model for storing links permits the links in a linkbase to be organised into meaningful groups (Millard et al 2000). For instance, links to resources that deliver the concept 'layer' in Photoshop can be collectively put under the same 'association' object. FOHM also allows several user-defined context objects to be attached to a link so that different contextual values can be used to pull out only the links relevant to the current users. This can be established with Auld Linky (Michaelides et al. 2001), a contextual link server for FOHM as it can query links based on several context values. For example, it can dynamically query for additional links that match the current taught concept delivered by the SCORM RTE (e.g. 'layer') and fulfil a user's preference for learning resources of type 'exercise', 'experiment' and 'case study'.

3.2 PLS Run-Time Environment

PLS Run-Time Environment generates adaptive links dynamically into SCORM by using the concept map and linkbase obtained from the PLS authoring environment, SCORM run-time sequencing data (activity id for the next learning resource to be launched, user id and navigational control data), and a user model. This section describes the process of designing and developing the environment). The processes involved are primarily described according to the following PLS implementation layers (as illustrated in Figure 3.0): PLS Web Service and PLS Client. The PLS Web Service is primarily composed of a contextual link server i.e. Auld Linky (Michaelides et al., 2001) that is responsible for performing adaptive queries to linkbases, based on a requested context object from the PLS Client.

Figure 3.0: Component-based conceptual diagram of the PLS Run-Time Environment



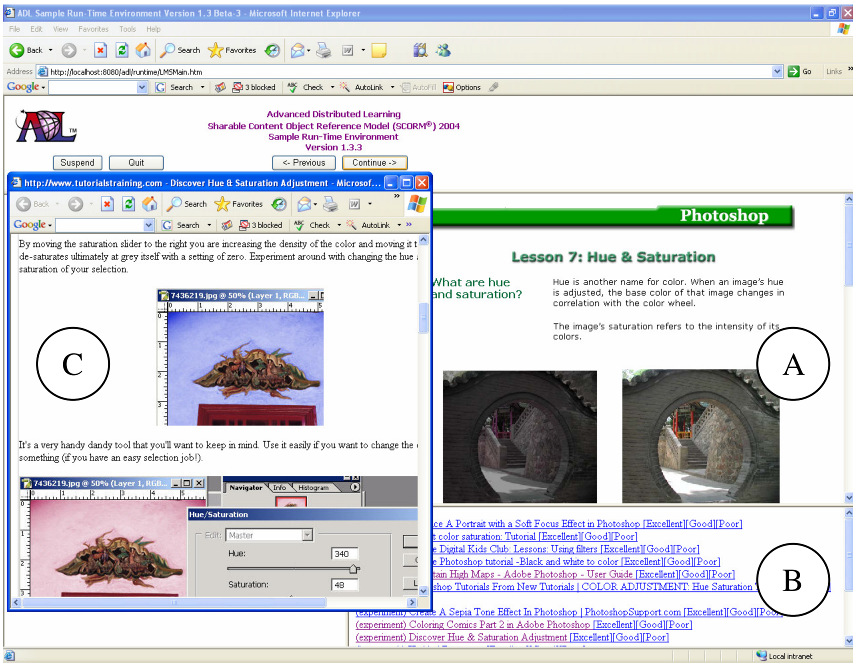
The objective of the PLS Client, therefore, is to built the right query object to the PLS Web Service and rendered the query result in the augmented SCORM learning environment. The query object must contain the following context values: concept name of a page to be delivered by the SCORM RTE and the type of learning resource or resource type (e.g. example, experiment, assignment) preferred by a user.

The availability of the concept map from the authoring environment permits the PLS Client to associate the resource currently delivered by the SCORM RTE with a concept name. This is possible because the concept map contains the association between a concept name and a specific activity id, therefore, as the value of the activity id can be obtained from the SCORM RTE, the corresponding concept name can also be deduced.

In order for the PLS Client to identify users' preferred resource type, it must obtain information regarding users' preferred learning styles, as this work holds an assumption that a user with a specific learning style prefers a specific set of resource types. Figure 3 shows that when students activate the PLS system via the SCORM RTE, PLS Client can derive information about its user id from the SCORM environment. Based on this user id, PLS Client can acquire the corresponding user model that contain information regarding user's preferred learning style. The 'Rules' component will then provide the data-mapping information between learning style and resource type. Finally, when both the learning concept and resource types have been obtained, PLS Client will build a query object and deliver it to the PLS Web Service in a form of a SOAP message.

Upon receiving the SOAP message, the PLS Service will interpret the message, and build a FOHM query object for Auld Linky in order to query the designated linkbases for the corresponding links to alternative resources. Once the query result (i.e a list of URLs together with their relevant properties) is obtained, it will be delivered to the PLS Client using a SOAP response. At the client, the SOAP message will be interpreted and the appropriate URLs elements will be retrieved and processed accordingly. Finally, the PLS Client will render an adaptively sorted list of links together with the relevant links' properties (e.g. resource type) to the SCORM Run-Time Environment on a user's browser along with other information such as resource type of the content referred by each links, as displayed in Figure 4.

Figure 4.0. The interface of the augmented SCORM RTE with the recommended links to alternative resources



The list of links to supplementary resources is provided in frame (B) matching the concept being displayed in the primary resources in frame (A). Once a link from frame (B) is selected a pop up window displaying the content of the URL appears in (C), so that the additional resources does not interfere in any way with a course author's intended resources and his/her instructional sequence. Frame (B) is the component that has been integrated with the existing SCORM environment.

While a student is interacting with the PLS, their selections of the presented links will be monitored. Based on a set of predefined rules and user's links selection behaviour, the system will decide whether the learning style currently deployed should be sustained or changed to the next recommended learning style.

4. EVALUATION AND RESULT

The architecture is implemented using Java. The run-time environment is built on a service-oriented architecture using Apache AXIS (Apache Extensible Interaction System) (Axis Guide 2005). An experiment was carried out to determine that the augmented SCORM environment contains adaptive links that point to alternative learning resources. These resources should match the learning concept of the learning resource being delivered by SCORM RTE, and their resource type should match users' Preferred Learning Style. For example, if SCORM RTE is displaying a primary learning material about the concept of layer in Photoshop, the PLS system should generate a list of links that contained alternative resources that are also about layer in Photoshop and with resource type that is preferable by a particular user according to his/her learning style.

In this experiment, four learning concepts (layer, selection tool, contrast brightness, and hue saturation) in Adobe Photoshop were selected and five resource types (lecture notes, example, homework, experiment, and case study) were considered for each selected learning concept. For every resource type of each learning concept, ten potential links were derived from the existing Web resource, hence, there were two hundreds links involved in this experiment.

The content page referred by each link returned by the system was manually evaluated by a content expert according to its relevancy to the named learning concept and resource type. The results were then categorised into five categories. The first category which is 'All Relevant' signifies that 99% of the content of the referred page represents the named learning concept/resource type. The second category which is 'Mostly Relevant' denotes that 80% of the content of the referred page represents the named learning concept/resource type while another 20% is for another learning concept/resource type. The third category which is 'Some Relevant' conveys that only 50% of the referred content page is of the correct learning concept/resource type, while the rest is some other learning concept/resource type. The forth category which is 'A Bit Relevant' denotes that the intended learning concept/resource type constitutes more than 10% but less than 50% of the content page. Finally, 'Not Relevant' represents irrelevant links for learning concept/resource type. Note that this experiment assumes that relevant links are represented by the first four categories (from 'All Relevant' to 'A Bit Relevant') whereas only the last category represents the irrelevant links. Based on this assumption, experiments results were recorded in the following tables: Table 2.0 and Table 3.0.

Table 2.0: The percentage of relevant and irrelevant links for each intended learning concept

Learning concept	**Relevant (%)**	**Irrelevant (%)**
Layer	80	20
Selection tool	78	22
Contrast brightness	58	42
Hue saturation	78	22
Average (%)	73.5	26.5

Table 3.0: The percentage of relevant and irrelevant links for each intended resource type

Resource type	**Relevant (%)**	**Irrelevant (%)**
Lecture notes	60	40
Example	90	10
Homework	55	45
Experiment	95	5
Case study	47.5	52.5
Average (%)	69.5	30.5

Results shown in Table 2 and Table 3 indicate that about 70% of the content pages referred by the links returned by our link generation algorithm were indeed relevant to the intended learning concepts and resource types. This means that the adaptive links generated by the PLS into the existing SCORM learning environment were appropriate to the taught concept delivered by the existing SCORM Run-Time Environment and users' preferred resource type. As the system holds to the assumption that a user with a particular learning style preferred a specific types of learning resources, the relevancy of the links to the requested resource types can also indicate that users with different learning style will receive an appropriate set of learning resources. Hence, we can conclude that the PLS system is capable of delivering a relevant set of alternative links to individual users with different user model attributes (e.g. Preferred Learning Style). It has also confirmed that the auto-generated linkbase does indeed contain many significant links. This result is encouraging after considering the fact that the linkbase is built without any human intervention.

5. CONCLUSION

This work presents an architecture that can generate adaptive links dynamically into SCORM open learning environment by using the information deduced from the SCORM package, its run-time sequencing data, and a user model. It provides a way to integrate an adaptive support service into the existing SCORM learning environment without proposing any modification to the SCORM specification. As a result, the existing pre-authored learning materials inside the SCORM package can be dynamically augmented with a set of personalised links to relevant supplementary resources. The successful implementation of the PLS prototype has demonstrated how the SCORM learning environment can be augmented with links to resources that are about the same topic as the user is studying, and which appear to support the learner's preferred learning style. However, the architecture is extensible to support other personalisation.

The advantages of this approach to e-learning users are as follows. First is the augmented SCORM learning environment can deliver learning materials that are preferable to both teachers (primary materials of their choice) and students (supplementary materials that suit their preferences). Secondly, it can personalise students' experiences within the SCORM open learning environment without subverting the teacher's narrative (as SCORM course sequencing mechanism is sustained). Lastly, with the augmented SCORM learning environment, students is now able to explore the primary taught concept using their preferred learning materials without adding extra load on the teacher to prepare variations of materials in order to satisfy individual needs.

Despite the automation process, the experimental results have demonstrated that around 70 percent of an automatically generated linkbase contains links relevant to the intended taught concepts and resource types. This encouraging result also shows that PLS approach can leverage the existing Web as a source for alternatives contents. However, to further improve the relevancy of the retrieved resources, the mechanism used for locating and analysing potential resources should be enhanced by integrating research outcomes from other research fields such as Natural Language Processing, data mining and semantic web.

REFERENCES

Abdullah, 2006. An Architecture for Augmenting the SCORM Run-Time Architecture with a Personalised Link Service, PhD Thesis, University of Southampton.

ADL Technical Team, 2004. SCORM Photoshop Examples Version 1.0. http://www.adlnet.gov/downloads/index.cfm

Axis Guide, 2005. Axis User Guide. http://ws.apache.org/axis/ java/user-guide.html

Google API, 2002. Google Search API, Beta Version. Retrieved February 20, 2006 at http://www.google.com/apis/ reference.html#2_2

IMS LIP, 2000. IMS Learner Information Packaging Best Practice and Implementation Guide v1.0. http://www.imsglobal.org/profiles/lipbind01.html

IMS SS, 2002. IMS Simple Sequencing Information and Behavior Model, Retrieved 28 May 2003 at http://www.imsproject.org/simplesequencing/v1p0pd/ imsss_infov1p0pd.html

Michaelides, D.T. et al, 2001. Auld Leaky: A Contextual Open Hypermedia Link Server. *In the Proceedings of the 7th Workshop on Open Hypermedia Systems, ACM Hypertext 2001 Conference*. Aarhus, Denmark. http://eprints.ecs.soton.ac.uk/6141/

Millard, D.E. et al, 2000. FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability between Hypertext Domains. *In Proceedings of the 11th on the ACM Conference on Hypertext and Hypermedia*, pp. 93-102.

Mödritscher, F. et al, 2004. Enhancement of SCORM to Support Adaptive E-Learning within the Scope of the Research Project AdeLE, *In Proceedings of ICL*, Villach, Austria, September 30, 2004, http://www2.iicm.edu/cguetl/papers/adaptiveelarningstandards/adaptivestandard.pdf

Power, G. et al, 2005. Goal Oriented Personalization with SCORM. *In Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005) (in press),* Kaohsiung, Taiwan, July 5-8 2005, pp. 467-471.

SCORM, 2004. SCORM 2004 Application Profile. http://www.adlnet.gov/scorm/index.cfm

SCORM RTE, 2004. Advanced Distributed Learning, SCORM Runtime Environment Version 1.3.1. http://www.adlnet.org/downloads/ 197.cfm