

EXTENDING EXISTING E-LEARNING PLATFORMS TO SUPPORT AUTOMATIC CONVERSION OF LEARNING CONTENT TO SCORM

Polyxeni Arapi, Nektarios Moumoutzis, Yiannis Maragoudakis, Stavros Christodoulakis

Laboratory of Distributed Multimedia Information Systems and Applications

Technical University of Crete (MUSIC/TUC)

Chania, Greece

{xenia, nektar, imarag, stavros}@ced.tuc.gr

Abstract

A critical issue in eLearning domain today is the preservation of existing learning material and user groups and learning communities already set up in the context of operational eLearning environments as the eLearning environments are being transformed to support the new interoperability specifications. This issue has tremendous economic impact. In this paper we address this issue in the context of a pre-existing eLearning platforms implemented on top of relational database management systems. The assumed architecture is based on two assumptions: (1) the pre-existing eLearning platform is implemented on top of a relational database management system which is used to store all the relevant information and (2) the model to be supported comes in the form of an XML schema. The prototype implementation refers to web-based eLearning platform (Distant Learning Center) based on a solid pedagogical framework (Virtual University Paradigm). The interoperability model adopted is SCORM 1.2. The proposed approach is based on a multi-tier architecture and makes use of software components that allow for the automatic creation of Package Interchange Files (PIF) that could be uploaded to any SCORM conformant Learning Management System.

Keywords

SCORM, interoperability, eLearning

1. INTRODUCTION

The issue of education has historically received much attention in periods of social transition. All intellectuals aiming at changing the society or some aspects of it have also stressed the role of education. It is thus evident that the current era of transforming our post-modern societies in the so called "information society" pays much attention to education and training as a means to ensure that the necessary competencies and skills are mastered by all the population. Nevertheless the transition phase that we are experiencing today is even more dependent on education and training because of the inherent emphasis on knowledge and continuous learning processes in the new forms of economic and social behavior that are emerging. Effective learning and training is nowadays a necessary precondition to protect the citizens against unemployment and social exclusions as the skills related to the use of information and communication technologies become indispensable.

The parallel advances in information and communication technologies make it possible to develop new kinds of educational and training services overcoming the constraints of traditional learning processes that were monopolized by well established institutions. This is an opportunity that has already received much attention and a lot of resources have already been invested in developing educational material as well as establishing learning networks that bring together teachers and learners, trainers and trainees. Until recently no significant efforts were taken to integrate all these systems and learning networks so that the learning resources could be reused thus saving valuable resources. The proliferation of the Internet, the World Wide Web and interoperability standards for digital learning resources create bright new application opportunities for the future.

ADL's SCORM model (SCORM 2001) is probably the most important step towards the development of interoperable eLearning systems by providing the means to overcome the incompatibility issues posed by proprietary implementations through proper standard descriptions of learning content objects and related material and functionalities. It is an effort supported by almost all the big players in the eLearning field that gains thrifty acceptance in both commercial and academic environment. It provides reusability and

interoperability of learning resources by leveraging the advances in all fields related to eLearning through proper integration and extension of various existing models. This strategy of integrating, instead of substituting, existing popular learning resource specifications, amplifies its widespread acceptance. The prototype implementation described in this paper is yet another exploration of the great potential of SCORM and proves that it can effectively support the interoperability needs of an existing eLearning platform with no compromises.

A major need shows up: How can we take advantage of the already existing learning resources in digital form as well as the well established learning networks and at the same time take advantage of the interoperability specifications that have emerged? The answer is not unique and usually depends on the technical details of the existing infrastructure. A simplified classification of preexistent eLearning solutions is to distinguish between coarse-grained file-based and fine-grained database-oriented approaches in terms of learning resources. The first approach is more straightforward and monolithic, unable to provide flexibility and personalized learning experiences. The second, more sophisticated and flexible, represents also a greater effort in the design and implementation of the corresponding system functionalities as well as an undoubtedly heavier authoring of learning resources. Having this in mind it is evident that eLearning environments of the second kind are more interesting to investigate with respect to their interoperability enhancements. Their complex structure makes it more difficult to support an interoperability specification and their existing functionalities and learning resources represent a bigger investment that should be preserved.

Talking in general, a fine-grained database-oriented eLearning environment could be conceived as three layered system: At the lowest level, there is a (usually relational) database management system. At the middle there is the specific database of the system along with a number of database transactions used to store and retrieve data including software components for the creation of dynamic HTML pages in case of web-based solutions. At the upmost level there are various user-centered applications providing functions for browsing, authoring, user communication etc.

An effective strategy for supporting interoperability in such an environment should make the less possible modification to the preexistent system. The proposed solution operates at the middle tier (the database of the eLearning system and the dynamically created HTML pages) and it defines a mapping between the logical model of the system with the content model of the SCORM. This is accomplished through a data binder that provides a bridge between the SCORM model and the underlying database schema. An export procedure picks up all the necessary data from the underlying database and then creates a manifest file describing the organization and the contents of the content package that is being created. The raw material of the package is collected piece by piece using an HTML parser that handles the (dynamic or static) pages of the learning resources.

The rest of the paper presents first (Section 2) the generic architecture assumed in our work. Moreover, it describes how the logical model of an elearning system can be mapped to SCORM and presents the export process of learning resources to the SCORM format. Section 3 presents the prototype implementation on top of a preexistent web-based eLearning platform and Section 4 presents some related work. Finally, Section 5 concludes and presents the directions for future research and development activities.

2. SHARABLE CONTENT REFERENCE MODEL (SCORM)

SCORM (SCORM 2001) is probably the most important development currently occurring in the area of e-learning standards and specifications. Simply stated, SCORM is a set of specifications for developing, packaging and delivering high quality education and training materials whenever and wherever they are needed. The use of SCORM enables reusability, accessibility and durability of the learning material in technology changes, and interoperability between different e-learning platforms.

SCORM in its 1.2 version consists of two parts: The SCORM Content Aggregation Model and the SCORM Run Time Environment. The SCORM Content Aggregation Model further defines a common way by which learning content can be interoperable, interchangeable, reusable and accessible. The SCORM Content Aggregation Model defines how learning content is identified, described, aggregated into a “course” and moved between systems. Specifically, the learning resources comprising a learning experience get packaged into a zip file (SCORM package of Package Interchange File (PIF)). This file

contains not only the course files, it also contains an XML file, referred to as the manifest file, describing the course contents and content sequencing.

The SCORM Run-Time Environment defines the means by which SCORM learning content is made interoperable between LMSs. The SCORM Run-Time Environment provides details on how LMSs should launch resources/content, communicate with the content (using a predefined language and vocabulary) and exchange predefined data elements during execution.

3. A GENERIC ARCHITECTURE

The proposed architecture supports the export of the educational content of an e-learning platform to SCORM 1.2 Content Packages (IMS CP 2001), given that the content is stored in a relational database. SCORM contains a rich dictionary of metadata terms that can be used for describing educational content. This architecture supports also the transformation of the metadata contained in the relational database to the format described in the IMS Learning Resource Meta-data Information Model (IMS LRM 2001). These metadata, used to describe the learning resources, will be also included in the generated SCORM Content Packages. The SCORM model has been selected, since –because it is widely adopted and implemented– it will be undoubtedly the future eLearning standard. However, this architecture, with appropriate modifications, can support other standards, if they use XML Schema for the description of the educational content. In our case, the outcome of the export process will be –in terms of SCORM– the Package Interchange File (PIF), a zip file that contains not only the physical files that compose an educational experience, but also an XML file, referred to as the imsmanifest file, describing the course contents and the content sequencing.

The architecture (Figure 1) follows a multi-tier approach with three distinct tiers:

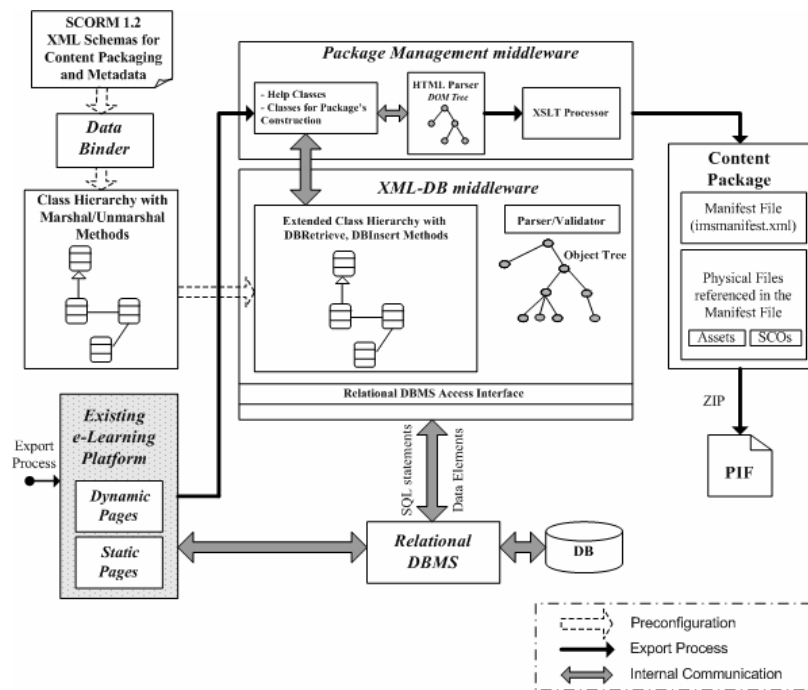


Figure 1: A generic architecture for extending an eLearning platform to support SCORM

1. The existing *e-learning platform*.
2. The *database tier*, which is a relational database management system along with the relational database, where the educational material and content metadata are stored.
3. The *middleware tier*, which realizes the transformation from database objects to the SCORM objects. The middleware consists of two parts:

- a. The *XML-DB middleware*, whose role is the mapping between the relational database schema and the two SCORM XML Schemas, one for the content packaging (IMS CP 2001) and the other for the description of learning resources with metadata (IMS LRM 2001), and
- b. The *Package Management middleware*, which is responsible for the transformation process.

The existing e-learning platform can be any platform that is implemented on top of a relational database to store all the information regarding the educational content. Both static and dynamic pages are supported in this architecture.

The relational database is responsible for the storage and retrieval of information regarding the educational content.

The transformation process is implemented through the middleware. For this to be possible, a mapping between the RDB Schema and the two SCORM XML Schemas (based on IMS Content Packaging Specification v1.1.2 and IMS Learning Resource Meta-data Specification v1.2) is needed, an XML document management system on top of the relational database management system. This is the role of the XML-DB middleware in this architecture. It is a set of software components responsible for the manipulation of SCORM XML documents. In this case the data management software does not rely on XML document modeling solutions (like DOM) but use another approach, called data binding (Bourret, 2001). Data binding offers a much simpler approach to working with XML and supports effective separation between document structure and data modeling.

The term “XML data binding” refers to the binding of certain XML documents to objects. These objects use a schema (classes) designed especially for the data in those documents. This allows applications to manipulate data that has been serialized as XML in a way that is more natural than using the DOM. For example, consider the following part of an XML document:

```
<Book bid="3">
  <BasicDescription>
    <Title>Professional JSP</Title>
    <Author aid="5">Mark Birbeck</Author>
  </BasicDescription>
</Book>
```

This could be bound to the Book, BasicDescription, Title and Author classes, so that when data is transferred from the XML document, the result is a tree of objects:



To develop the XML-DB middleware we have used the help of a data binder to bind the SCORM XML Schema to objects. There are numerous XML data binding products (Bourret, 2001) capable of transferring data between XML documents and java objects. They are classified in two broad classes, namely design-time binders (require configuration based on a DTD or an XML Schema before they can be used) and run-time binders (used directly in code to serialize and de-serialize objects as XML elements). Design-time binders are usually more flexible in the mappings they can support.

The architecture in Figure 1 assumes a design-time binder (Bourret, 2001). Thus, a configuration process (illustrated with dashed arrows) is necessary to create the classes corresponding to XML elements present in the document class to be handled. To be more specific, the two SCORM XML Schemas about the Content Packaging and the description of learning resources with metadata (based on IMS Content Packaging Specification v1.1.2 and IMS Learning Resource Meta-data Specification v1.2 respectively), should be used, as depicted, to create two set of classes, each one (class) corresponding to a particular

element type. The class hierarchy that has been generated includes also marshal methods. Marshal methods perform the serialization of the object tree to XML document and the whole process is called “marshalling”.

The classes created during the data binding configuration process are extended with DBRetrieve methods. DBRetrieve methods retrieve data from the database with the objective to build object trees that could be used thereafter to create SCORM XML documents. DBRetrieve methods rely on both the class hierarchy created by the data binding configuration process and the structure (relational schema) of the underlying database. The communication with the relational database management system relies on the use of standard interfaces like JDBC (Relational DBMS Access Interface in Figure 1). Standard SQL statements are used to store/retrieve data from the underlying relational database.

A SAX-based parser and corresponding validator can be used to ensure that the generated XML documents conform to the two SCORM XML schemas. The parsing process is closely related to the marshal methods. An assumption is made here that the code generated by the data binder handles all the necessary interactions with the parser.

The Package Management middleware does all the remaining work needed for the transformation. In practice, it manages the export process. Hence, it is responsible for finding all learning resources that compose an educational experience, and store them in form of SCO's and Assets inside the SCORM Content Package along with the metadata in appropriate format that describe them. The functionality of this part will be analytically explained later in this paper, where the whole export process is described.

3.1 Mapping the logical model of an eLearning system to SCORM

Before proceeding to the implementation of the transformation system, a mapping between the eLearning platform's elements and SCORM elements must be found. With simple words, one must firstly define which are the lowest level components or the minimum granularities of educational content that might be re-used in other learning experiences. Each of these components should provide useful learning content by itself and must be independent of learning context, to be reusable. If we add one more characteristic to the above, which is that these components must be designed to be launched and tracked by a SCORM-compliant LMS, we reach the SCORM definition for Sharable Content Objects (SCOs). All learning resources from which one SCO consists of, are the SCO's Assets. For example, some times an individual web page might be considered as a SCO, but usually more than one page is required to convey meaningful learning content. Once we have decided which are the SCOs and added some code for the SCO – LMS communication, we must also decide which will be their organization or organizations and sequencing inside a learning experience that will be packaged in SCORM format. The existing presentation order can be followed for the organization and sequencing of the learning resources inside the generated SCORM package.

The above points must have been considered, since they determine how the DBRetrieve methods will be constructed. In addition, the mapping between the DB elements and SCORM regarding the metadata should be considered. The appropriate values should be retrieved from the database to “fill in” the corresponding elements and attributes in SCORM XML Schemas.

3.2 The export process

As we already mentioned, the export procedure includes the transformation of the educational content that composes a learning experience and metadata describing it, which are stored in the RDB, to the SCORM 1.2 format. The outcome of this process is the creation of a SCORM 1.2 Content Package, which contains all learning resources (physical files – SCOs and Assets) composing the learning experience, and the `imsmanifest.xml` file, which describes these learning resources and their sequencing. The metadata that describe the learning resources can reside as inline data into the `imsmanifest` file, or in separate XML documents, one for each learning resource. In the last case, pointers are included in the `imsmanifest` file to the corresponding XML documents. Both cases can be supported.

Once the export procedure has been started, the DBRetrieve methods are sequentially executed and they build firstly the “manifest” and “organizations” objects. The number and the structure of the “organization” objects inside the “organizations” object depend on the way the DBRetrieve methods have been written.

In an eLearning platform, static or dynamic pages can be used to present the educational content to the learner. In the first case, the same static pages can be stored inside the SCORM Content Package to present the learning experience to the learner. In the second case, dynamic pages could not be saved inside the SCORM Content Package and used as they are, because if so the SCOs would not be context-aware. Hence, a mechanism should exist to store in Content Package the result of the execution of the dynamic pages in the server, which has the form of static pages. This mechanism is supported in this architecture (in Package Management Middleware) (Figure 1).

Another point is that a web page (either it is considered as a SCO or an Asset) may contain images, links to other files or even references to files that may not be apparent in page's interface, but are referenced or included in the HTML code (e.g. Stylesheets, JavaScript files). These files should also be stored in appropriate places in the Content Package, so that the pages could be presented correctly and transitions to other pages or files could be performed. For this reason, our architecture includes (in Package Management middleware) a mechanism that finds all pages' dependencies, downloads all referenced files and stores them in appropriate places inside the Content Package. An HTML Parser and an XSLT Processor are needed and the whole procedure is illustrated in Figures 2 and 3.

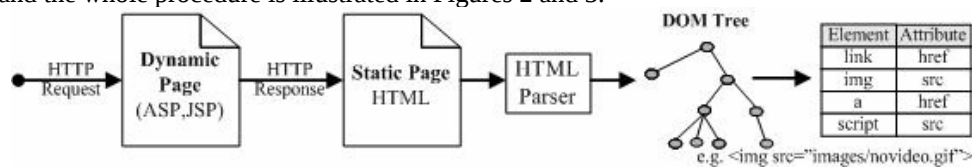


Figure 2: Detection of all pages' dependencies

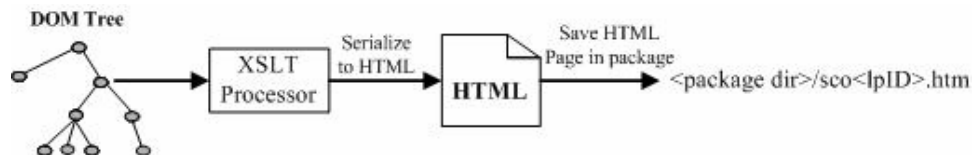


Figure 3: Serialization of the DOM Tree to HTML and storage of the generated HTML page in the package

In the DOM Tree that is generated from the HTML page code, all references to downloaded files are converted into relative paths. Moreover, some new elements can be automatically added to the DOM Tree, in order to add code for the SCO-to-LMS communication.

Instead of downloading and saving in the Content Package all the physical files needed, pointers can be placed inside the manifest XML document that point to the files' URLs. In this case, these files are "external resources".

After finding and storing in the Content Package all the physical files that are needed in a learning experience or pointers to the corresponding URLs, in case they are external resources, the "resources" element is created, which contains and describes all resources that compose the learning experience. These are SCOs or Assets. The information about the "resource" elements is known from the DB and the previous procedure.

Finally, the manifest instance gets marshalled (output as XML) to a file (imsmanifest.xml), inside the content package and the contents of package get zipped to a file (Package Interchange File), that can be imported to any SCORM compliant LMS.

For example, in Figure 4, a SCORM package, which has been generated from our implementation described in the next section, has been successfully imported in the ADL's Sample Run-Time Environment v1.2.1. Moreover, our generated packages have been tested with SCORM Version 1.2 Conformance Test Suite Version 1.2.3, which is provided by ADL, to ensure that they are 100% SCORM conformant.

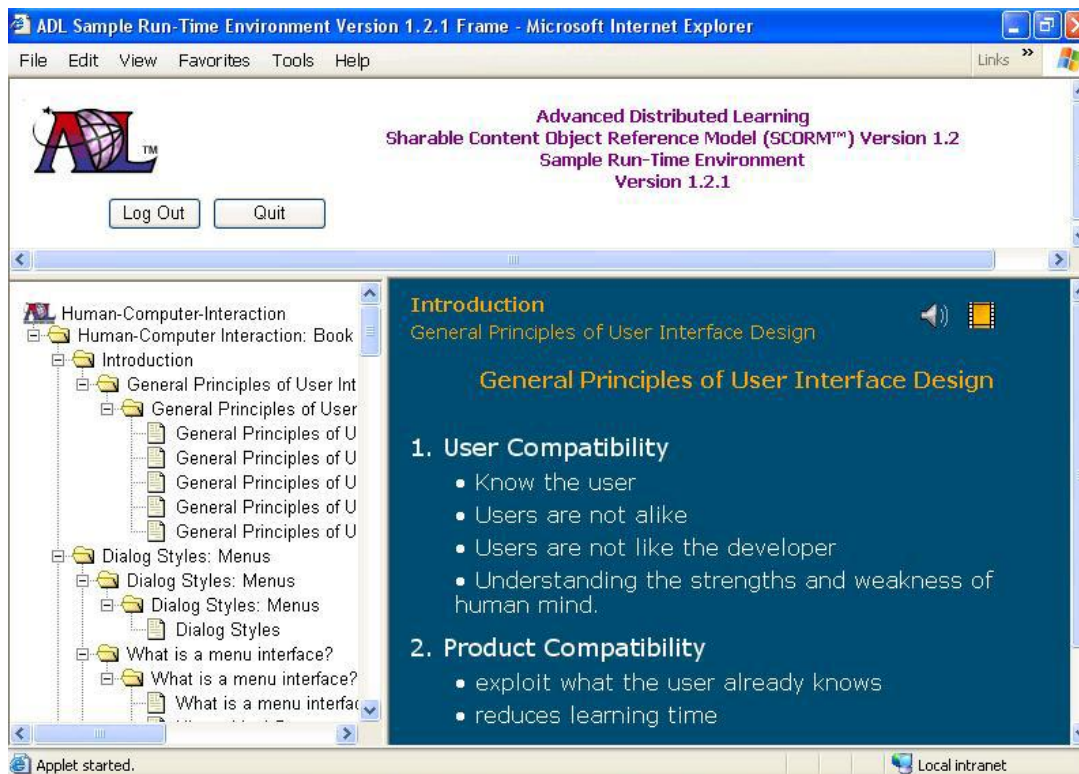


Figure 4: Import of a generated package to the ADL's Sample Run-Time Environment v1.2.1

4. PROTOTYPE IMPLEMENTATION

The proposed architecture has been successfully implemented on top of an existing e-learning web-based system, the Distant Learning Center (DLC) (Stellakis, 2000). DLC is a system that provides distance training through the use of the World Wide Web (Web-based training) and has been developed in the Laboratory of Distributed Multimedia Information Systems and Applications of the Technical University of Crete (TUC/MUSIC) for the needs of the European ARCHIMED project (Advanced Multi-media-System Architectures and Applications for Educational Telematics, INCO-COPERNICUS PL96106). DLC is based on Internet and on a web server for the information delivery to its users' browsers. It is very flexible, since all the information is stored in a relational database. The communication with the relational database management system relies on the use of an ODBC interface. HTML, DHTML and JavaScript have been used for the content presentation to the browsers and Active Server Pages (ASP) for dynamic pages' development.

There are two user categories in DLC: The *Learners*, which attend the courses and participate to the learning activities provided by the system, and the *Instructors*, which develop, organize and supervise these courses.

Learners can attend a course and see the courseware in an environment where they can browse content, see examples, and see related videos. They can also add annotations on the courseware, which can be visible only to them or to all users. Moreover, they can add annotations on annotations. The collaboration and communication between learners take place either through these annotations, or through their participation in discussion forums, places where learners can interchange opinions and ask questions by messages visible to all users. Learners can, also, be organized into groups, having this way the capability for a better collaboration. Finally, learners can take exams for each course. Their evaluation is made automatically from the system or after the instructor's mediation depending on the type of the exercise or the exam.

Instructors add courseware, examples, exercises and exams. They evaluate the learners' answers in these exercises, and, also, construct and supervise the discussion forums of their courses. Moreover, they

supervise the whole course, attending the public annotations of the learners or adding their own annotations.

The personal communication between the members of the system (learners and instructors) takes place through private messages, which are text messages that apply to one recipient.

The courseware in DLC, can be organized either as a book (with chapters, sections, and subsections) or as a lecture (with a series of slides referring possibly to the components of a book). The same book organization can support numerous lecture organizations. Moreover the DLC has a digital library of documents or files with content related to the courses.

Private or public annotations can be added to text objects, audio pages, video pages and examples. The role of an annotation is to explain or comment the object with which it is related.

For the learners' assessment, each course has a collection of test objects (Questions and Exams). The questions are related with the course's sections. They are of four kinds: Unique answer, True/false, multiple choice and Free answer questions.

4.1 Mapping the logical model of DLC to SCORM

In DLC, the chapters, sections, subsections and the presentations can be considered as candidates for becoming SCOs. Searching for the minimum content granularity that can be reused in another learning experience, one can easily see that the best selection is slide objects. If instead another class of objects (e.g. chapters) was selected, it would not be functional, since we could not reuse its lower components. It should be nevertheless stressed that one slide does not always have independent educational content. It could be part of a cohesive unit of slides that cover one educational issue. Although a slide is not always independent, we would like to be able to reuse this slide in other courses composition

Once we have defined the learning resources, we must decide which will be their organization or organizations. The two organizations of DLC (book and lecture) will be followed to organize the educational content inside a SCORM package. In figures 5 and 6 the mapping between DLC and SCORM elements in both cases is illustrated.

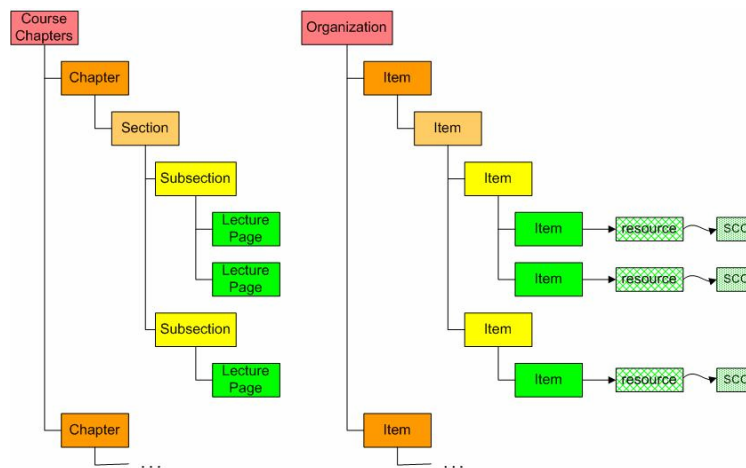


Figure 5: Mapping for book organization

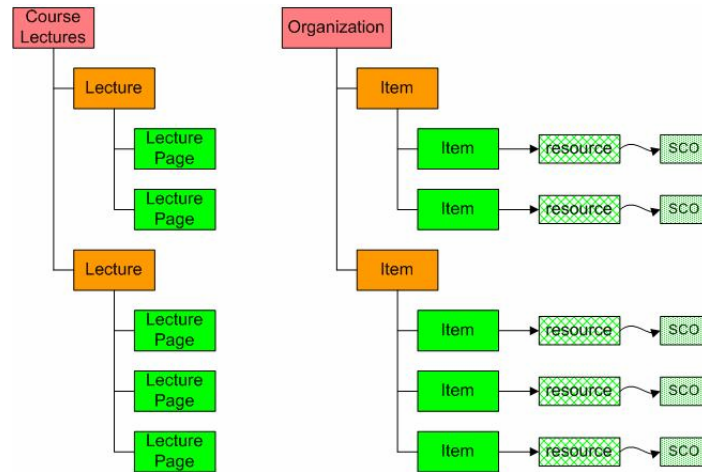


Figure 6: Mapping for lecture organization

Then, we proceed to the construction of DBRetrieve methods and we use the above architecture for the transformation process. The export and import of all kinds of Annotations is also supported. Annotations are considered as Assets. In case of a text annotation, a web page is stored in content package. If the annotation is an audio or video file, it can be placed also in content package, or a pointer to its URL can be used (external resource).

5. RELATED WORK

In general, describing a course through SCORM needs reconstruction of the course almost from the beginning. This process can be performed either at low-level, by finding a mapping of existing resources to SCORM content model components, writing appropriate XML documents and adding physical files to construct SCORM Content Packages, or at a higher level, using existing tools to design SCORM conformant courseware, as Aspen ToolBook Instructor. In the paper of Qu Ch. (Qu and Nejd, 2002), a SCORM conformant courseware has been implemented, by re-designing all learning resources contained in an existing Java course according to the SCORM Content Model and also reconstructing the course structure according to the SCORM Content Packaging. Considering a big collection of courses that have been developed in one learning platform, this manual redesigning of each course from the beginning would be a time and cost ineffective procedure. Instead of re-designing all courses that have been developed in a specific learning platform, the proposed architecture can be used to extend the platform's functionality performing this conversion process automatically.

6. CONCLUSIONS

To maintain and extend the value of preexistent investments in eLearning systems and preserve digital learning material there is a need to support interoperability specifications and emerging standards that allow learning resource reusability and integration of operational eLearning environments. The methodology and architectural considerations reported in this paper address this need of extending preexistent eLearning environments in order to support the most promising and successful interoperability specifications, which constitute the SCORM suite.

The baseline of this work is a generic reference architecture that assumes a sophisticated preexistent eLearning system based on a relational database management system. We propose effective mechanisms for transforming the contents of this system in the standard form proposed by the SCORM specifications. We also address the major implications and implementation issues and describe a successful implementation on top of an existing e-learning web-based system, the Distant Learning Center (DLC).

It is evident that the feasibility of the proposed approach constitutes a successful evaluation of the SCORM specifications and enables its faster proliferation as it puts forward a safe transition pattern for existing systems and learning resources repositories.

DLC, empowered with SCORM export functionality is used in two European vocational training networks: The first network addresses the needs of workers in the tourism and cultural sector (ADONIS) and the second focuses on the mass media industry (KNOSOS).

At this time, our system does not support the export or import of questions or exams. The main reason is that SCORM 1.2 does not contain a schema or guidance for the description of assessments. Assessments constitute a special kind of educational content and need special treatment. IMS Global Learning Consortium has developed the Question and Test Interoperability (QTI) specification, which defines an XML format for the coding of questions and exams. This specification makes able the transfer of such kinds of objects between different LMSs and is very possible in the future to be a part of SCORM.

One major direction of future research is the investigation of interoperability between SCORM and audiovisual standards. A first step towards this direction has been reported in (Frantzi et al, 2004). That work investigated the interoperability issues between SCORM and the TV-Anytime Metadata Model in order to provide effective specifications and system architectures that will transform digital TV into a learning medium thus opening new opportunities to the broad public. We believe that such integration will provide new delivery channels for learning services including ubiquitous computing devices allowing a transition from current situated learning paradigm to what we can call "ubiquitous learning". We are currently working on exploiting interoperability issues between SCORM and MPEG7.

REFERENCES

1. Arapi P., Moumoutzis N., Christodoulakis S. (2003). Supporting Interoperability in an Existing e-Learning Platform using SCORM, *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT 2003)*, Athens, Greece
2. Bourret R. (2001). XML Data Binding Resources, URL: <http://www.rpbouret.com/xml/XMLDataBinding.htm>
3. Frantzi M., Moumoutzis N., Christodoulakis S. (2004). A Methodology for the Integration of SCORM with TV-Anytime for Achieving Interoperable Digital TV and e-Learning Applications, *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, Finland
4. IMS CP (2001). IMS Content Packaging Specification v1.1.2, <http://www.imspj.org/content/packaging/index.html>
5. IMS LRM (2001). IMS Learning Resource Metadata Specification v1.2.1, <http://www.imspj.org/metadata/index.html>
6. Qu Ch., Nejd W. (2002). Towards Interoperability and Reusability of Learning Resource: a SCORM-conformant Courseware for Computer Science Education, *Proceedings of the 2nd IEEE International Conference on Advanced Learning Technologies (ICALT 2002)*, Kazan, Tatarstan, Russia
7. SCORM (2001). Sharable Content Object Reference Model, Version 1.2, U.S. ADL
8. Stellakis, D. (2000). *Design and implementation of a distance learning environment on the web*, Diploma thesis (in greek), Electronic and Computer Engineering Department, Technical University of Crete, Chania