BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES · Volume **5**, No 1

Sofia. 2005

Adaptive Learning in Web-based Educational Environments

Mariya Monova-Zheleva[1]

[1] *Bourgas Free University, San Stefano Str. No 62, 8001 Bourgas*

***Abstract**: This paper describes approaches and conceptual solutions suitable for delivering and supporting of individualized adaptive learning in networked Web-based educational environment. Functionality and conceptual architecture of interoperable e-Learning System as well as a methodology for the development of intended instructional flows are briefly described.*

***Keywords**: adaptive learning, e-learning, learning management system, simple sequencing, interoperability, reusability.*

1. Introduction

Society's transition to the information age is likely to impact learning and education in two ways. Firstly, rapidly improving technology will enable higher quality learning to be made available to an ever-growing audience through increasingly sophisticated modes of presentation. Secondly, the very nature of the information age may require a different kind of preparation (i.e. learning) than was the case in the industrial age. The European Union has put e-Learning on top of the agenda for several years. Now we can see many creative approaches in a wide variety of European projects [9-11]. This paper refers to a project called "The Knowledge Shop" [12]. This project is funded from Socrates – Grundtvig 2 Program. The main task of "The Knowledge Shop" is a network of Knowledge Shop Centers (KSCs) to be developed. KSCs are local Internet-based training centers, which integrate a set of tools and methods necessary for remote instructor-led and/or successful self-paced adaptive learning in a given knowledge domain with the help of interactive multimedia learning materials. The KSCs learning platforms and content have to be structured and standardized in order to provide adaptive learning to adult learners.

Adaptive learning, presenting the right material to the learner on demand, can be described and implemented using data representations from learning technology standards such as learner profiles, competency definitions, sequencing rules, learning

objects, etc. [8]. To operate a KSC successfully, interoperability and reusability of the learning materials have to be ensured as well as various facilities, services and options must be offered.

2. Interoperable, reusable and adaptive learning content – some critical issues

SCORM is a collection of guidelines that provide a foundation for how to build learning technologies for interoperability, accessibility and reusability. SCORM is widely accepted because it is designed to provide integration of existing popular learning resource specifications regarding learning resource metadata, aiming to fully describe a learning resource regardless of a specific learning context or educational purpose; learning content packaging, for moving whole instructional units; communication protocols between learning content and LMSs.

ADL SCORM [1] and IMS Learning Resource Metadata Specifications [6] both reference IEEE Learning Object Metadata or LOM [5]. These are the most significant specifications for metadata. The concept of the package as a standardized way to exchange collections of digital resources between different learning management systems, authoring tools, and content repositories is firstly introduced by AICC [2]. The IMS Content Packaging Specifications describe a whole package through a special file called manifest and SCORM references these specifications. Concerning the communication between LMS and learning content, AICC firstly presented Run Time environment functionality by means of an API and a Data Model. ADL, based on the experience of AICC, has presented its own API.

ADL has long recognized that SCORM 1.2 has serious limitations [3]. One of the most critical problems is that there is no mechanism to describe how a learner is to move through or is permitted to move through a learning experience. The only method for learning content selection and navigation is a rendering of a menu tree of content as a table of contents taken directly from the Content Package Organization element.

The IMS Simple Sequencing Specification (IMS SS) [7] was conceived of as a solution to this major issue. Its primary goal was to present a model and mechanism to describe learning content that captures common structures of learning patterns, including conditional delivery and branching. IMS SS provides a means to represent information needed to sequence learning activities in a variety of ways according to Tracking Model, and Activity State Model. Tracking Model is used to record information about the learner's interactions with activities, and the learner's record for objectives (e.g. completion, measure) to control the selection and sequencing of other activities. Activity State Model is used to record information about the status of learner's interactions with an activity and set of global attributes for activities. Sequencing Definition Model is an information model used to describe the desired sequencing behaviors. The IMS Content Packaging Specification provides a ready structure for relating learning activities to content resources – item elements and its relationship to resource elements. Therefore, IMS SS maps the concept of learning activity to an item element or a collection of item elements within an organization element, and to an organization element itself, as defined by the Content Packaging Specification. The Content Packaging XML Binding is extended by IMS SS to define how sequencing information is associated with packaged content. The specification defines the possible sequencing

operations and the informational elements required to describe the desired operations and their parameters, i.e. the desired sequencing behavior. All information defining sequencing behavior for an organization is presented within that organization. Any packaged collection of learning activities may contain sequencing descriptions defined by the IMS SS Specification. ADL has adopted an extended version of AICC Data Model in order to integrate the IMS SS Specification.

From the above said it is clear that SCORM 1.3 Specification has precedence in comparison with the other ones. This is the main reason why more and more learning technology products tend to be compliant with it.

3. SCORM compliant learning content and Run Time Environment

SCORM specifies a Content Aggregation Model (CAM) and a Run Time Environment (RTE) for learning objects. According to SCORM CAM several categories of learning content exist. *Assets* concern learning content in its most basic form that can be delivered in a Web client.

Sharable Content Object (SCO) is a collection of one or more assets that can be launched by LMS. There are two kinds of sharable content objects – basic SCO and data enabled SCO. Basic SCO uses the API for minimal communication with the LMS. The system can track only the time between the launch and the termination of such an object. Data-enabled SCO can communicate with the LMS using the RTE. In this way LMS can track down the learners' interactions with the learning content.

Content Aggregation concerns a content structure that can be used to aggregate learning resources into cohesive instructional units on the basis of structural and learning taxonomies [13].

From the technical point of view architecture of any SCO can be represented through a set of layers. Fig. 1 illustrates SCOs layered architecture.

<table>
<tr><td rowspan="2">SCO content, user interfaces, response processing
(HTML, custom scripts)</td><td>Applet or object
(e.g. simulation or Flash movie)</td></tr>
<tr><td>Translation functions
(e.g. JavaScript catchers for Flash FSCommands)</td></tr>
<tr><td colspan="2">SCO-specific logic
(e.g. navigation, common response processing, state)</td></tr>
<tr><td colspan="2">SCO-generic logic
(e.g. score and status management, etc.)</td></tr>
<tr><td colspan="2">SCO API method calls
(including data translation or reshaping where necessary)</td></tr>
<tr><td colspan="2">API adapter discovery,
management of the communication session with the API adapter</td></tr>
</table>

Fig. 1. SCO scripting layers

By separating functions into separate layers, layers can be maintained separately. Future versions of SCORM will probably introduce some changes in the API, the API protocol, the communication data model, or error codes. Revising only the lower, reusable layers can accommodate most of these changes.

The RTE has three key components: a launch mechanism, an API adapter and a Data Model. The launch mechanism refers to the responsibility of LMS to determine the learning resource that will be next delivered to the learner. The API is a communication mechanism between the LMS and the content. It is provided by the LMS, but it is responsibility of the SCO to locate the API adapter and make the appropriate API calls in order to communicate and exchange information with the LMS according to the Data Model. Some of the Data Model elements relate to the Tracking Model for the respective activity. A Tracking Model captures information regarding the learners' interactions with the resource associated with an activity. In run time, each activity experienced by the learner is associated with tracking status data, which may affect the selection and sequencing of other activities.

CAM specifies five types of metadata application profiles to describe assets, basic SCOs, data-enabled SCOs, activities and content aggregations. It concerns packaging of the manifest and related physical files into a zip file called Package Interchangeable File (PIF). With the integration of the IMS SS Specification, CAM supports the consistent sequencing of learning activities.

4. The e-learning system functionality

The possibility for adaptation of the learning content accordingly to the learner's performance and progress is a key issue. We refer the term "adaptive learning" to the capability to modify any individual student's learning experience as a function of information obtained through their performance on situated tasks or assessments.

All learning activities can be associated with sequencing information defined by the content author. In run time, each activity experienced by the learner is associated with tracking status data, which may affect the overall sequencing process. This means that learners with difficulties in satisfying the learning objective should be able to experience additional activities to improve their cognitive skills. Learners could not allow experiencing an activity for more than a predefined period of time or a number of attempts.

The achievement of the aforementioned goal needs of the implementation of the following functionality:

- Login/logout services and ability to maintain a list of registered users.
- Allow learners to register/unregister for a particular course.
- Import SCORM 1.3 learning content that can be grouped into a variety of categories;
- Possibility the registered users to change their preferences.
- Provide learners with the means that will help them navigate through a course's learning activities.
- Inform learners in the case an attempt to experience a learning activity is not allowed.
- Monitor the users' interactions with the content, as specified within a resource.
- Conditionally affect, in run time, the sequence of the activities according to the learner's progress and performance.
- Provide bookmark facilities and report on learner's performance on assessments.

A system administrator is responsible for adding to or deleting learners from the list accordingly to their intention to register to or to unregister from the LMS.

The courses can be developed using any SCORM 1.3 – compliant authoring tool such as LERSUS 3.0.0.11, knowledgeWorks, etc. The courses are in .zip format (in the form of a PIF). An administrator is responsible for defining the different categories the imported courses can be divided into, as well as for importing new courses.

Registered learners enter the LMS through a login session and have the possibility to survey a list of the available courses. A learner can select the course(s) and make the appropriate registration(s). The LMS provide the table of contents and other means that help the learner in the navigation through learning activities contained in a course. If any of learner's navigation requests violates the sequencing rules specified by the content developer, the learner is being informed. Otherwise, the learner can view the learning resource and interact with it.

The sequencing rules are based on learner's progress and performance and affect the availability of the learner is allowed to experience. The overall sequencing process is controlled by sequencing engine, which is provided by the LMS. The sequencing loop describes how the sequencing process is invoked, what happens during the sequencing process and the potential outputs of the process. The diagram in Fig. 2 shows the sequencing loop.

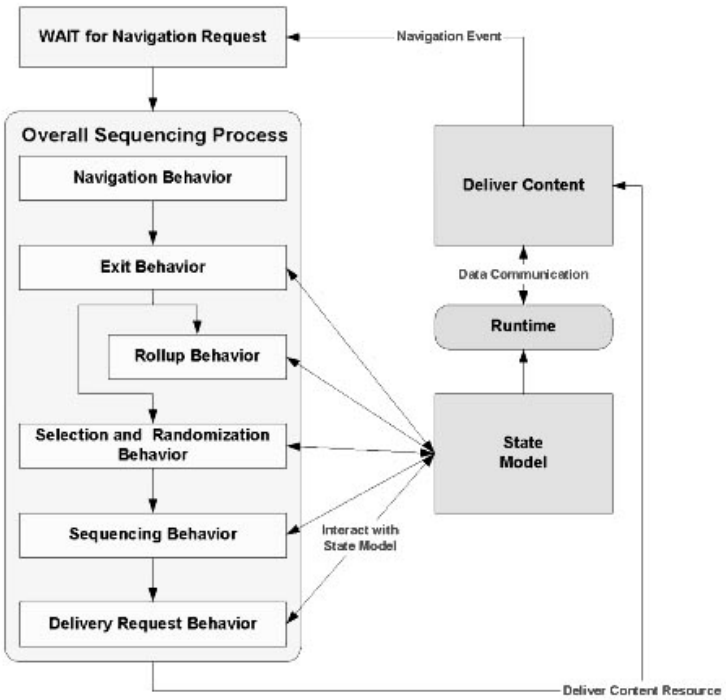

Fig. 2. Sequencing loop

The system waits for a navigation event, which is translated into a navigation request and passed to the Overall Sequencing Process. This process acts as the process control by invoking various sequencing processes. Navigation Behavior acts on the Navigation Request and result is Exit and/or Sequencing Request. Exit Behavior acts on any pending Exit Request and results in a Sequencing Request. Sequencing Behavior

acts on any pending Sequencing Request and may result in a Delivery Request. Delivery Behavior acts on any pending Delivery Request and may result in the delivery of some activity to the learner.

5. Model of the system architecture

SCORM API and Data Model are exploited by LMS as the standardized means for monitoring the learners' interactions with the content [4]. The LMS intends to be a learning system for delivering and managing learning content, which adapts accordingly to the learners' knowledge level. Each learning activity is associated with a predefined learning objective. Information that can be exchanged between the system and the content concerns SCO enabled navigation events, the session state and total time a learner spends on an activity, the learning objectives accomplishment status, the learner's preferences and performance on assessments, etc.

The LMS is based on client-server architecture with the following components:

- Server side – the server delivers learning resources in a consistent way, according to the learners' requests and based on the information associated with each available course. The server component consists of the following sub-components:

– package handler is responsible for accessing and extracting the manifest file contained in the package;

– manifest handler is responsible for accessing information contained in the manifest file and storing it into the database;

– RTE Data Model is responsible for providing the communication mechanism between the server and the SCOs;

– sequencing engine is triggered by various navigation events. The events could be learner or SCO enabled. Information gathered through the RTE Data Model is mapped into information regarding learning activities. The sequencing engine captures this information and information from the manifest file in addition in order to start the overall sequencing process.

- Client side – Learners will be able to access the learning content and other LMS services, through a browser window via Web client. The LMS provides a variety of user interfaces in order to manage the different functions, such as registrations, reports, learner's profile management and course administration. The LMS provides a browser window that encapsulates a SCORM API adapter. The content is delivered in a frame of this API-enhanced window. In another frame, LMS provides learners with navigation controls to help them navigate through the content and trigger the overall sequencing process.
- Database.

A database is used to provide data persistence. In the database is stored information regarding data extracting from the manifest file, data resulted from the communication between the server and the various SCOs, as well as information about the different courses, and about the registered learners (e.g. preferences, existing competencies and properties such as measure, expiration date etc.).

6. Development of adaptive learning content – conceptual solution and methodology

With the integration of the IMS SS Specification, SCORM allows the learning strategies to be translated into sequencing rules and actions, which are associated with the activities a learning experience consists of. Any SCORM-compliant LMS is then able to sequence activities in a way consistent with the instructional process and deliver learning resources based on the learner's interactions with them.

The process of defining a specific sequence of learning activities begins with the creation of an aggregation of content. The outcome of this process is the creation of a content package, which contains all learning resources (physical files) composing the learning experience, and the imsmanifest.xml file, which describes these learning resources and their sequencing.

The implementation of adaptive learning in an e-learning environment could be promoted and facilitated by providing of sequencing templates for the development of instructional flows.

The author has developed the following methodology described below. The sequencing template has to be considered as a sample package. The development of the content structure of the template is based on structure ontology with tree levels of abstraction – course, module, and lesson. The imsmanifest.xml file of the package contains a description of course organization and the actual XML encoding of a concrete sequencing strategy. The course content is represented as a sequence of learning resources. The learning content may include simple static pages, MIME resources of any type (e.g. DOC, PDF files), services and dynamically created objects. The sequencing engine transforms the information from RTE Data Model into tracking model information and uses it together with the sequencing information elements from the imsmanifest.xml to control sequencing process. For passive content (static pages) certain values in the tracking model are set automatically. Active content is responsible for setting and updating the values of the tracking model and objectives (locally and globally scoped data items) associated with the corresponding activities. Such content contain references to files that are not apparent but are referenced or included in the HTML code (e.g. JavaScript files or Stylesheets). JavaScripts are used for storing and retrieving the definitions of sequencing objectives and information about sequencing activities and their relationships to sequencing objectives as well as for updating the learner's tracking information. The JavaScript files are stored in appropriate places in the package, so as the content pages could be presented correctly and planned transitions to be performed. The sample package contains template web pages, which can be edited by the instructor (content author).

Hence, the sequencing template itself describes the conceptual organization of the learning content as a sequence of template pages and provides the learning strategy implementation translating it into sequencing strategy. Such sequencing template can be used in different knowledge domains from different instructors who want to follow the described in the package content organization and the implemented learning strategy. In this case, instructor is responsible only to identify (or create) and then to incorporate the relevant multimedia content in each of the template pages accordingly the subject matter of the course taking into consideration the concrete learning objectives and context. Another important task is the set of appropriate metadata elements to be

added so that the package content to be more fully described. The metadata elements such as course title, description, keywords, author, etc. are used for cataloging a content package and allow an instructor, or learner, to search and to identify the learning content they want to use or view. The final step of the adaptive learning content package development process is content of the package to be saved into a .zip file (Package Interchange File), that can be imported to any SCORM compliant LMS.

Suppose that there is a packaged aggregation of content which contains of one examination object named "Pretest" and two content modules: module "Remediation" comprising general learning content representation and module "Enhancement" comprising advanced learning content representation[1]. On Fig. 3 is represented the package learning content organization.

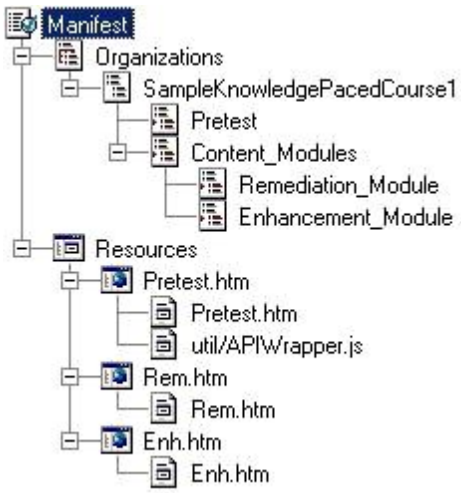

Fig. 3. Organization of the content package

If content packaging organization element ("SampleKnowledgePacedCourse1") and each item element within it have no defined sequencing information the learner is free to choose learning activities at will, in any order, without any restrictions on number of attempts. LMS shall provide some user interface devices that enable selection of learning activities such as table of contents and/or navigation buttons. Hence, the learner can progress through the content pieces ("Pretest", "Remediation Module", and "Enhancement Module") in any order.

In the rest of this section a description of manifest instance fragments are presented, which contain sequencing information associated with the "SampleKnowledgePacedCourse1" organization and items within it so that the "Knowledge paced learning with pretest and remediation" learning strategy to be assured. This strategy is based on delivering and managing learning content, which adapts to the learner's knowledge level and progress.

According to this strategy the learner must experience and complete the pretest first. While experiencing a pretest, the learner cannot choose to exit before completing it (the learner must attempt each question). The learner can attempt to answer of the pretest questions in any order.

[1] Only learning objects associated with leaf nodes of the hierarchical tree structure of the organization are launched and tracked as SCOs.

The LMS reports on learner's progress and performance on assessment. This helps learners to be informed of their cognitive skills as well as to be motivated to achieve a better performance. If learner does not attempt to answer all the pretest questions the other pieces of the content cannot be viewed. The learner "satisfies" the pretest learning objective if he/she has answered correctly all the test questions. Description of the sequencing rules associated with the "Pretest" item element in the "SampleKnowledgePacedCourse1" organization follows:

```
<item identifier="Pretest" identifierref="Resource_Pretest">
    <title>Pretest</title>
    <imsss:sequencing>
        <imsss:sequencingRules>
            <imsss:preConditionRule>
                <imsss:ruleConditions>
                    <imsss:ruleCondition operator="not" condition="attempted"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action="stopForwardTraversal"/>
            </imsss:preConditionRule>
        </imsss:sequencingRules>
        <imsss:objectives>
            <imsss:primaryObjective objectiveID="primaryobj" satisfiedByMeasure="true">
                <imsss:mapInfo targetObjectiveID="content"
                    readNormalizedMeasure="false" writeSatisfiedStatus="true"/>
                </imsss:primaryObjective>
            </imsss:objectives>
        </imsss:sequencing>
</item>
```

The delivery of other course material is based on the learner's score in the pretest. If the learner does not satisfy the pretest objective the content of remediation will be delivered first and then learner can view the content for the knowledge enhancement. If the learner satisfies the pretest objective the content for remediation will be skipped and the content for knowledge enhancement will be delivered directly to the learner. If this learner wants he/she can view the content for remediation by signaling previous event. The learner decides when to progress through the remediation and enhancement content and in what direction. The sequencing informational elements are described below:

```
<item identifier="Content_Modules" isvisible="false">
    <item identifier="Remediation" identifierref="Resource_Rem">
        <title> Remediation_Module</title>
        <imsss:sequencing>
            <imsss:controlMode flow="true"/>
            <imsss:sequencingRules>
                <imsss:preConditionRule>
                    <imsss:ruleConditions>
                        <imsss:ruleCondition condition="satisfied"/>
                    </imsss:ruleConditions>
                    <imsss:ruleAction action="skip"/>
                </imsss:preConditionRule>
            </imsss:sequencingRules>
            <imsss:rollupRules rollupObjectiveSatisfied="false"/>
            <imsss:objectives>
                <imsss:primaryObjective objectiveID="primaryobj">
                    <imsss:mapInfo targetObjectiveID="content"
                    readNormalizedMeasure="false" readSatisfiedStatus="true"/>
                </imsss:primaryObjective>
```

```
            </imsss:objectives>
        </imsss:sequencing>
</item>
<item identifier="Enhancement" identifierref="Resource_Enh">
        <imsss:sequencing>
            <imsss:controlMode flow="true"/>
            <imsss:rollupRules rollupObjectiveSatisfied="false"/>
         </imsss:sequencing>
    </item>
    <imsss:sequencing>
        <imsss:controlMode choice="true" flow="true" forwardOnly="false"/>
    </imsss:sequencing>
</item>
```

The content aggregation ("SampleKnowledgePacedCourse1") exits after the learner has experienced or skipped each piece of content at least once. The sequencing information elements are described below:

```
<imsss:sequencing>
        <imsss:controlMode choice="false" flow="true" forwardOnly="true"/>
            <imsss:sequencingRules>
                <imsss:exitConditionRule>
                    <imsss:ruleConditions>
                                <imsss:ruleCondition condition="completed"/>
                    </imsss:ruleConditions>
                    <imsss:ruleAction action="exit"/>
                </imsss:exitConditionRule>
            </imsss:sequencingRules>
            <imsss:rollupRules>
                <imsss:rollupRule childActivitySet="all">
                    <imsss:rollupConditions>
                                <imsss:rollupCondition condition="satisfied"/>
                    </imsss:rollupConditions>
                    <imsss:rollupAction action="satisfied"/>
                </imsss:rollupRule>
                <imsss:rollupRule childActivitySet="any">
                    <imsss:rollupConditions>
                                <imsss:rollupCondition operator="not" condition="satisfied"/>
                    </imsss:rollupConditions>
                    <imsss:rollupAction action="notSatisfied"/>
                </imsss:rollupRule>
        </imsss:rollupRules>
</imsss:sequencing>
```

The sample package contains tree template pages – for the pretest, for the remediation content, and for the enhanced content. Each page can be edited by the instructor through system's or external authoring tools. Pretest consists of multiple-choice questions. The evaluation is made automatically from the system. The instructor has to fill the question forms prepared in advance with appropriate content (e.g. text of the question/s and alternatives, weight/s, etc.) and to put the relevant multimedia content for the remediation and knowledge enhancement into the other two template pages. There is no restriction of the number of questions, i.e. the instructor can delete or add questions in the pretest page if necessary. The number of the multimedia files (HTML documents, videos, audios, etc.) that can be used to represent the remediation and enhanced content in a better and more interactive way is not limited also.

The manifest instance is marshaled to imsmanifest.xml file in the content package and the contents of package get zipped to a Package Interchange File that can be imported in any SCORM compliant eLearning system. The sequencing rules described in the imsmanifest.xml file of the template package assure the learning strategy implementation but the effectiveness of learning process considerably depends of the quality and pedagogical relevance of used learning content resources and test questions.

7. Conclusions and future work

The described methodology has been used by the author in the development of sequencing templates according to the limited number of widely adopted learning strategies such as:

- knowledge paced learning with pretest and remediation;
- knowledge paced learning with posttest and remediation cycle;
- knowledge paced learning with pretest, posttest and cycle of remediation.

Such templates help and facilitate the instructors/authors without big experience in programming to describe flows of learning activities through content according to the outcomes of a learner's interactions with the content.

SCORM provides reusability and interoperability of learning resources by leveraging the advances in all fields related to eLearning. With the integration of the IMS SS Specification SCORM 1.3 enables authors to translate learning strategies into sequencing rules. One important trend of our future work is the sequencing templates for other learning strategiesto be developed. ADL's SCORM 1.3 Specification is a new one. The majority of eLearning systems support the previous version – SCORM 1.2. The main disadvantage of SCORM 1.2 is that this Specification does not define how to sequence SCOs and user-choice sequencing is assumed. Therefore the next step in this direction of the work is the adaptation of these templates in order to be used in SCORM 1.2 compliant LMS which are for the moment most widespread.

References

1. Advanced Distributed Learning.
   http://www.adlnet.org
2. Aviation Industry CBT Committee.
   http://www.aicc.org
3. Blackmon, W., D. R. Rehak. Customized Learning: A Web Based Approach. – In: Ed-Media World Conference "Educational Multimedia Hypermedia&Telecommunications", 2003, Issue 1, 6-9.
4. Bouras, C., M. Nani, T. Tsiatsos. A SCORM-Conformant LMS. – In: ED-Media World Conference "Educational Multimedia, Hypermedia and Telecommunications", Volume 2003, Issue 1, 10-13.
5. IEEE Learning Technology Standards Committee.
   http://ltsc.ieee.org
6. IMS Global Learning Consortium Inc., IMS Learning Resource Metadata Specification.
   http://www.imsproject.org/metadata/index.html
7. IMS Global Learning Consortium Inc., IMS Simple Sequencing Specification.
   http://www.imsproject.org/simplesequencing/index.html
8. Martin, V. Butz, O. Sigand, P. Geard. Internal Models and Anticipations in Adaptive Learning Systems. – In: Proceedings of the Second Workshop "Anticipatory Behavior in Adaptive Learning Systems 2004", 17 July, Los Angeles, USA, 2004, 86-110.

9. Project ARCHIMED, web site.
http://www.music.tuc.gr/Research/Managerial/ARCHIMED/Managerial.htm
10. Project ADONIS, Web site.
http://www.adonisproject.hu
11. Project KNOSOS, Web site.
http://knosos.music.tuc.gr
12. "The Knowledge Shop" Project Web Site.
http://www.the-knowledge-shop.org
13. W a t a n a b e, H., S. K o g a, K. K a t o. Development of Learning Management System and SCO Presentation Program Based on SCORM. – In: Proceedings of the 4th IEEE International Conference "Advanced Learning Technologies", 30 August – 1 September, Joensuu, Finland, 2004, 870-871.