Applying learning styles to SCORM compliant courses

Ioannis Kazanidis, Maya Satratzemi

Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
{kazanidis, maya}@uom.gr

Abstract

This paper proposes a general framework to develop SCORM compliant courses that provide adaptation according to user learning style. The SCORM standard as well as some of the most popular learning style models and adaptive systems that implement them are briefly presented. The general framework is introduced and a case study with some evaluation outcomes is discussed.

1. Introduction

Free navigation through Internet courses as well as the large number of users with different characteristics and needs utilizing online courses create some quite major problems, like disorientation, cognitive overload, and discontinuous flow to the e-learning procedure [1]. Students who do a course on the Internet tend to be more heterogeneously distributed than those found in a traditional classroom situation [2]. Therefore, learners tend to have differing initial knowledge of the domain, educational goals, progress in a course, and learning styles. Web-based courses should adapt the learning process according to these differences [3] and provide personalized navigation and presentation of the educational content. Interactivity and adaptability of the courses is proposed as a solution to the above problems.

A new research trend has led to the development of the Adaptive Educational Hypermedia Systems (AEHS). These systems through the Adaptive Navigation (AN) and Adaptive Presentation (AP) techniques provide personalized courses over the Internet. Generally, AEHS are autonomous hypermedia systems without having a common development framework. However, some of them, like ProPer [4], Opal and Vibora are trying to adopt a standard such as SCORM in order to achieve interoperability and reusability of the educational content. The main aim of SCORM is to offer compliant courses that are “RAID” (Reusable, Accessible, Interoperable, Durable).

SCORM consists of three components. The first is the Content Aggregation Model (CAM), which describes elements used in the learning process, as well as the methods used for packaging. CAM enables an exchange from system to system using Extensive Markup Language (XML). In accordance with CAM, the learning content is made up of Assets, SCOs and Content Organization. Assets are electronic representations of media and other pieces of data, like text, video, images, HTML fragments etc. The most basic educational resources are there, and one or more of them constitute a SCO (Sharable Content Object). SCOs should be independent of their learning context and contain as much instruction as necessary to convey their topic. They are described by metadata, are sharable, and can be used by any other SCORM compliant course. Content organization is a map of structured instruction units, referred to as Activities. Leaf Activities (those that do not consist of other activities) are associated either with a SCO or an Asset resource.

The second SCORM component is the Run-Time Environment (RTE). It defines the requirements of the Learning Management System (LMS) in order to manage and communicate with the instructional material, as well as to track learners’ interactions with the content. RTE provides an Application Program Interface (API) with predefined functionalities in order to pass information from content to LMS and vice versa. It also defines the required vocabulary that can be used to pass on that information. The main goal of RTE is to provide interoperability between SCOs and LMSs.

The third component is Sequencing and Navigation (SN). It describes the sequence of the content through a set of navigation events and defines the navigation rules that can be set by the instructor so as to make sequencing more efficient. SN is responsible for the identification of the next appropriate learning activity for delivery.

SCORM can support limited AN techniques, as well as most of the AP techniques of the educational content. Thus, a content developer can take advantage...
of SCORM standard and provide course adaptation according to user needs.

A SCORM compliant LMS, named ProPer [4] has been developed by the authors, which provides adaptation using both system and SCORM capabilities. In this paper emphasis is given to the adaptation using the SCORM rules and specifications. A framework that lets any SCORM compliant course to provide AP of its educational content according to user’s learning style is proposed. A SCORM compliant course that adapts the presentation of its learning material has been developed and tested in our system’s summative evaluation process as a case study.

2. Learning Styles

In order to achieve better learning outcomes, several research streams are attempting to provide adaptivity of the learning process. One of these streams exploits educational theories about student learning styles. It tries to provide adapted presentation and organization of learning content according to particular user learning style. In this section some of the most well known learning style theories and models are presented in brief.

Honey and Mumford [5]. According to this model learning is a process of knowledge construction through four distinct stages of a circle: a) Having an experience, b) Reviewing the experience, c) Concluding from the experience, and d) Planning the next step. The student can start from any stage of the cycle and continue to the others. Each stage is related to a learning style. Thus, the corresponding learning styles to the above stages are the: Activist, Reflector, Theorist, Pragmatist. In order to categorize the students the Honey and Mumford model uses an 80-item true/false questionnaire.

Felder and Silverman [6] is also a popular learning style model. It incorporates five dimensions that constitute the student learning style. These are: Sensing-intuitive, Visual-verbal, Indicative-deductive, Active-reflective, Sequential-global. A questionnaire consisting of 44 two-choice answers is used in order to find student learning style.

Witkin, Moore, Goodenough and Cox [7] distinguish students into two learning categories, field dependent (FD) and field independent (FI) which constitutes the homonymous model. In this model, student style can be defined through the Group Embedded Figures Test.

All the learning style theories and models differentiate students according to the factors that influence their learning experiences. Every model proposes a specific instruction framework that will help students to gain better learning outcomes. However, this requires a personalized learning process according to the individual student’s learning style. The use of adaptive web-based systems is an ideal way to provide this kind of personalized instruction.

3. Related Work

Many adaptive systems try to incorporate adaptation according to student learning style. The adaptation can be mainly applied by: a) changing the presentation type of the educational material (video, sound, text, animation etc), b) defining a different concept sequence (individual learning path), c) changing the alternative content sequence of a concept, and d) providing (or not) navigational tools. Some of the most popular adaptive hypermedia systems that adapt according to users’ learning style are the following:

INSPIRE [8] categorizes students according to the Honey and Mumford model through a questionnaire they are required to answer before starting their study. INSPIRE uses the same learning material, each time focusing on different perspectives of the presented topic according to the user’s learner type. Concepts in INSPIRE can be presented by example, activity, theory and exercise. Similar to INSPIRE, Tangow [9] presents the contents by example and theory.

AES-CS [1] is another system that adopts the Witkin’s Field Dependent/Field Independent Model in order to state which instructional strategy to apply to any particular learner.

WINDS [10] and CS383 [11] implement the Felder-Silverman model and provide different types of media, like text, videos, graphs etc. Arthur [12], on the other hand, delivers content in three different media forms (text, sound, images). More specifically, it initially provides students with the content through random media. If the student does not succeed in the test that follows each theory unit, the system then provides the same content with different media. iWeaver [13] implements the Dunn and Dunn model [14] to adapt content presentation similar to that of Arthur, and to provide specific learning tools according to learner’s preferences.

All these systems adapt either content presentation or the applied navigational process according to user learning style and other personal characteristics. However, the developed courses can be used only from the specific systems they are designed for. In addition, an instructor wanting to apply adaptation in a course has to use one of these systems. It would be most accommodating if course authors could apply individual teaching strategies according to users’ learning styles on every SCORM compliant course.
4. Framework for the development of SCORM compliant Adaptive Courses to user Learning Style

The main goal of this work is to propose a general framework for the development and design of SCORM compliant courses that adapt to user learning style. Adaptation can also be applied to other user characteristics but such a discussion is not within the scope of the current research.

The proposed framework consists of two main layers: the Pedagogical Design Layer and the Technical Layer.

The Pedagogical Design Layer includes three basic activities and is similar to that proposed by Barrera–Sanabria [15]. First, the educational goals of instruction have to be defined. These goals allow evaluation and/or measurement of the learning process. Second, the instructor has to decide on the contents to be presented to the user, which must cover all the predefined educational goals. And finally, the third step of this layer is to effect the definition of the applied instructional strategies. Here, the instructor studies all the parameters, such as the generated student groups according to the particular learning style, the teaching strategies for each group, the methods of acquiring user learning style and then decides the possible adaptation strategies that can be applied for each learning style model. Selecting the applied learning style model completes the Pedagogical Model.

The Technical Layer can be divided into five main steps.

Finding the educational resources. The user needs to find or create the necessary educational material for the course. The material should be related to the predefined course content. Additional resources may be necessary in the case that the learning style model requires the same information to be delivered through different types of media (text, graphs, videos etc), or other types of educational material (theory, activities, examples etc). Further still, a range of difficulty or even a variety of written languages could be provided to cover the different levels in user knowledge.

Course construction design. At this stage the instructor designs the course structure. The course is separated into concepts and the course map as well as the relations between each concept, are defined.

Pretest development. Most learning style models incorporate a pretest, which can categorize users into different learning style groups. The developer has to build a pretest that can discover the user’s learning style and store it through the SCORM RTE. At this stage, users can also develop a pretest that measures their initial knowledge in the domain. This, however, is beyond the scope of this work and so it will not be dealt with further.

SCO’s development. In the next step, the course SCOs have to be developed. SCOs must contain all the necessary educational material plus the code that will communicate with the SCORM RTE. In the case that a specific learning strategy has to be applied according to the adopted learning style model, SCOs must comprehend additional Javascript with the required intelligence.

Course Manifest development. Course construction is completed by the development of the manifest file. This file includes the essential information about the course structure and sequence according to CAM. Instructors can either write the appropriate XML code in a text editor or use special graphical tools.

5. Applying the Case study

The authors have developed an adaptive SCORM compliant LMS, named ProPer. In order to evaluate the system we needed to develop a SCORM compliant course. It was decided to provide AP according to the Honey and Mumford [5] learning style theory, like INSPIRE. ProPer supports the AN of a course according to user goals, preknowledge, and progress. However, in order to provide AP of the content, the framework presented below should be followed.

5.1 Pedagogical Design Layer

The educational goals of the course developed were to teach students the basics of Object Oriented Programming in Java. After having defined the main goal, the course contents were classified based on the sub-goals. Each user was grouped into one of the four model categories according to their answers on the Honey and Mumford Questionnaire. Alternatively, by reading a simple description, users could decide which learning style better suited them. The teaching strategy that had to be followed required the presentation of the same content but with emphasis on specific knowledge modules for every category of learners. In order to achieve this, SCOs with various knowledge modules presented on one page, had to be created. The module presentation sequence depended on the user’s learning style. For example, if the user has been categorized as an ‘Activist’, then the presentation of the educational material would start with a Question, which is followed by an Example and a Theory hint. Similarly, a
Reflector’ starts his/her study with Theory and continues with an Example and a Question.

5.2. Technical Layer

Following the discussion of all the pedagogical aspects, the authors proceed to the implementation of the course.

Finding educational resources. The course’s educational content was derived from the Sun Java tutorials.

Course construction design. The structure of the course included five main chapters (OOP, Language Basics, Classes, Objects, More on Classes). Each chapter had to comprise the appropriate SCOs. Additionally, the weight of each SCO on the course grade was defined.

Pretest development. As user learning style had to be determined by a questionnaire, an online test with the appropriate true/false questions was developed. The user simply answers the questions and the course automatically selects his/her learning style through the Javascript intelligent code. The main problem at this stage was how to store the retrieved learning user style. We decided to store it as a score in a SCORM objective, named “lstyle” using the SCORM RTE API. A score from 0.1 to 0.4 corresponds for each learning style category. Thus, other SCOs can later read the “lstyle” objective’s value and find the user learning style by providing the appropriate adaptation. In order to achieve this, an additional code had to be written both for every SCO web page and the manifest file.

SCO’s development. Each SCO was constituted by one or more knowledge modules. First it had to initialized, read the “lstyle” objective’s value and provide the appropriate presentation of the educational material. SCO will decide which knowledge module to present first as well as the sequence of the other modules’ hints according to the user learning style. Finally it had to store the user score and be terminated. All the intelligent of the SCOs is laid on Javascript code. The developed course was based on code of ADL’s Photoshop Examples Version 1.1.

Course Manifest development. We used the Reload program in order to create the course manifest file and package the course. Special care was given to the Objectives declaration. The Learning Style Selection SCO should have the privileges to both read and write the “lstyle” objective, while all the other SCOs have the privilege only to read. In addition, at this stage the weight of each SCO on the course was declared. A snapshot of the imsmanifest.xml file code is shown in Figure 1.

Figure 1 Part of the imsmanifest.xml code
5.3. Course Evaluation

This course is used in the summative evaluation of ProPer. Thirty-one subjects studied the course and were asked to answer a questionnaire, which consisted of Likert type (5-range) questions. The subjects agreed that the learning style adaptation was both useful and well applied (4.0 and 3.8). Only one asked for a different presentation of the contents and recommended changing his learning style model manually.

6. Discussion & conclusion

Although AEHS provide a wide range of adaptive features and techniques, their courses still have shortcomings in regards to reusability and interoperability. It would be a step forward if developers were able to create SCORM compliant courses that provide adaptation to user characteristics, such as learning style. This paper proposes a framework for just that, i.e., developing SCORM compliant courses that provide AP according to user learning style. As an example the development of a course that provides adaptation according to the Honey and Mumford learning style model was discussed.

The same framework can be followed in order to adapt SCORM courses to other learning style models. However this framework requires the knowledge of both programming (javascript, HTML) and SCORM specifications. It would be a great impulse for the course construction the development of appropriate tools for easy course authoring. Finally, it is worth mentioning that similar to our approaches could be taken to provide both AP and AN to other user characteristics, such as progress, preknowledge, language etc.

7. References


