Efficient Authoring of SCORM Courseware Adapted to User Learning Style: The Case of ProPer SAT

Ioannis Kazanidis and Maya Satratzemi

Department of Applied Informatics, University of Macedonia, 54006, Thessaloniki, Greece
{kazanidis,maya}@uom.gr

Abstract. Online courses are the most popular way to deliver knowledge for distance learning. New researches attempt to personalize the educational process with the use of the Adaptive Educational Hypermedia Systems. Moreover, due to the significant amount of time, money and effort devoted to creating online courses, developers strive to incorporate standards, such as SCORM, for the reusability, interoperability and durability of the educational content. However, it is a difficult task for teachers without programming knowledge to design and author adaptive courses. This work presents ProPer SAT, an authoring tool implemented for quick and easy SCORM courseware construction which can also be adapted to specific user learning styles.

Keywords: adaptive hypermedia, learning styles, SCORM, course authoring.

1 Introduction

Online courses are one of the most popular means of distributing knowledge worldwide. A large number of learners, irrespective of their abode, age or study constraints can have access to them, mainly through Learning Management Systems (LMSs). Since these learners have differences in the knowledge levels of the domain, educational goals, personal characteristics and learning styles, a lot of effort is devoted to the personalization of instruction with the intention of improving the learning outcome. Adaptive Educational Hypermedia Systems (AEHSs) aim to individualize the learning process to meet students' characteristics. The development of these systems, however, is not based on a common framework, as there are no fixed rules, techniques or methods acceptable to all. Consequently, although developers spend a lot of valuable time, money and effort on their applications, they pay nowhere near enough attention to instructional strategies. Moreover, due to the absence of a common framework, the courses along with the educational content produced are deficient in reusability, interoperability and durability. More specifically, it is difficult to apply the educational content from one course to another; or to distribute a course from one AEHS to another, since most times they are not compatible. A proposed solution to this problem is the adoption of recently accepted educational standards.

The most popular education standard is SCORM (Sharable Content Object Referenced Model) [1]. In previous work [2][3] we combined the adaptive features of AEHSs with the adoption of the SCORM standard and its specifications. Furthermore, we promoted the concept that it is possible for native SCORM compliant courses to

M. Spaniol et al. (Eds.): ICWL 2009, LNCS 5686, pp. 196–205, 2009.

be adapted to user learning style [4]. Despite all of this, however, instructors still need to invest a great deal of effort to construct adaptive SCORM compliant courses, which apart from the SCORM framework, also require knowledge of learning style theories, and programming capabilities (JavaScript, HTML). It is therefore inevitable that instructors with little or no prior programming knowledge come up against difficulties during course construction, as there is a definite lack of easy authoring tools for non-programmers.

In this paper we propose a framework for easy authoring courses that are SCORM compliant and/or adaptive to learning styles. We present an authoring tool named ProPer SAT that enables authors, with no prior knowledge of programming, or SCORM specifications, to construct quick and easy adaptive SCORM compliant courses. These can instantly be distributed by ProPer - an adaptive SCORM compliant LMS we have developed - or exported as a SCORM package for later use by any SCORM compliant LMS. The paper also includes a short description of the SCORM specification, AEHSs technology and the possible adaptation to learning styles of these systems. In addition, work related to adaptive systems with adaptation to learning styles, SCORM compliant systems, and course authoring tools are presented. Moreover, an introduction of the developed authoring tool as well as a discussion of the proposed framework are given. Finally, conclusions are put forward and further work recommendations made.

2 Theoretical Background

The proposed system combines technologies and characteristics from three domains of web based instruction research. It uses adaptive technologies, such as AEHSs, it conforms with SCORM specifications and incorporates adaptation to user learning styles. This section briefly presents SCORM standard, AEHSs and some of the most popular learning style models.

SCORM specification

The SCORM standard was developed by the Advanced Distributed Learning (ADL) initiative and is based on previous standards (AICC, ARIADNE, IEEE LTSC, IMS). Its main aim is to offer RAID (Reusable, Accessible, Interoperable, Durable) courses.

SCORM includes all the technical specifications that are necessary for the development, organization and distribution of the learning content. It is based on learning objects known as SCOs (Sharable Content Objects), which are the smaller logical entities that can be delivered by a compliant course and communicate with the LMS. Every SCO consists of information about the creation, discovery and aggregation of the appropriate educational resource in their most basic form (assets). Assets are digital media, such as text, images, video or any other digital data that can be delivered through a web based system.

The latest version of SCORM has three main components [1]: (i) The Content Aggregation Model (CAM), describes the parts used in a learning experience; how to package them for exchange; describe them for search and discovery; and define sequencing rules. (ii) The Run-Time Environment (RTE) defines the LMS requirements for managing and communicating with the educational material. RTE provides an

Application Program Interface (API) for communication between SCOs and LMS. (iii) Sequencing and Navigation (SN) describes the information and behaviors an LMS must apply to offer a designed learning experience.

Adaptive Educational Hypermedia Systems

AEHSs find user specific characteristics, preferences, learner progress and goals, in an attempt to personalize the learning experience according to these factors. They consist of three main components: (i) The Domain Model (DM) represents the system's domain knowledge; (ii) the User Model (UM) depicts the user's knowledge of the domain as well as his/her individual characteristics; and (iii) the Adaptation Module (AM) describes how the adaptation will be applied and which items will be adapted.

Adaptation is implemented using two major technologies: the first, Adaptive Presentation (AP) provides adaptation of the content level. It can adapt either the presentation and/or the content according to the factors stored in the UM. According to Brusilovsky [5] there are three main AP technologies: Adaptive text presentation, Adaptive multimedia presentation, and Adaptation of modality. The second is Adaptive Navigation (AN), which adapts the links or the course's link structure in order to steer the user towards certain links and away from others. The main AN technologies are: direct guidance, link hiding, link disabling, link removal, link sorting, link annotation, link generation, hypertext map adaptation.

Learning Styles

According to Honey and Mumford [6] learning style refers to a person's habits and patterns of behavior that determine the desired means of learning. Some of the best known learning style models are: Kolb's experimental learning model [7]; Honey and Mumford based on Kolb's model; the Felder and Silverman model [8]; the Witkin's Field Dependent/Field Independent model [9]; and Gardner's theory of multiple intelligences (MI) [10]. All have been used in several AEHSs with the intention of helping students gain a better learning outcome. The system we propose uses the Honey and Mumford categorization based on Kolb's model.

According to **Kolb's experimental learning Model**, learning is a process of knowledge construction through four distinct stages of a cycle: a) concrete experience (CE); b) abstract conceptualization (AC); c) reflective observation (RO); and d) active experimentation (AE). The student can start from any point in the cycle and continue in that order to pass through all the stages. This model distinguishes learners into four categories each representing the combination of two preferred styles: Divergers (CE-RO), Assimilators (AC-RO), Convergers (AC-AE) and Accommodators (CE-AE).

Similar to Kolb's the learning cycle in **Honey and Mumford model**, is consisted by four stages: 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 point in the cycle and progress to the others. Each stage is related to a particular learning style. Thus, the corresponding learning styles are the: a) Activist, b) Reflector, c) Theorist, and d) Pragmatist.

3 Related Work

Currently many systems are attempting to incorporate adaptation to user learning styles. Next we can see some of them and the learning model they implement:

- INSPIRE [11] implements Honey and Mumford model;
- TANGOW [12], LSAS [13], CS383 [14] incorporate several dimensions of the Felder-Silverman model;
- AES-CS [15] adopts the FD/FI model;
- EDUCE [16] is based on Gardner's MI theory.

There are additional tools, which assist authors in the creation of adaptive courses. These systems allow the instructor to define specific rules in order to apply adaptation to the course or they provide ready instructional strategies. Below we briefly present some such systems.

AHA! [17] is an open source web server extension that adds adaptation to applications, such as on-line courses. WELSA [18] is a system consisting of an authoring tool and a course player. By integrating characteristics from several models, it creates a unified learning style model. It uses a set of metadata to describe learning content and it takes into account some behavioral patterns for learner tracking functionality. MOT [19] is an online environment for the authoring of adaptive educational hypermedia. In MOT the author can either select an adaptive strategy that corresponds to an instructional strategy created by a different author and apply it to an arbitrary concept map or lesson map [19] or define his/her own instructional strategy.

Most of the above systems are inappropriate for teachers who are non-technical and who have little or no programming knowledge. However, to their aid comes VIDET and REDEEM . VIDET [20] is a visual authoring tool for designing adaptive courses, whose goal is to support easy course authoring for non-technical instructors and attempts to give the teacher full control over the adaptive operations being performed. It provides authoring tools for manipulating the hypertext structure, the content, the user model and the adaptive interaction model. REDEEM [21] allows instructors with little technological background to import pre-existing courses and provides them with tools to define how they want to teach the material. Its focus is not on the construction of domain material but on authoring pedagogy.

Even if some attempts for reusable educational content have been made [19], all the above authoring tools do not conform to a standard such as SCORM. Thus, they have limited reusability. Even when many popular LMSs support SCORM courses (Moodle, Claroline, Web-CT) they do not include tools for course authoring. Hence, they support only the import of ready SCORM compliant courses developed by external SCORM authoring tools.

Some of the most common SCORM authoring tools are the systems Reload and eXe, available for free, both of which support SCORM package construction and SCOs metadata definition. A drawback, however, is that even though these tools provide fast course packaging implementation the author is still required to have good knowledge of SCORM specifications. Moreover, the educational content must be designed in an HTML editor or the author might even need to invest in expensive authoring tools, such as Lectora or Lersus editors. Worse still in the case were

adaptive presentation of content to user learning style is required authors have to write in a more or less complicated JavaScript.

4 Introduction to ProPer

ProPer [2][3] incorporates a combined architecture of AEHSs and SCORM LMSs. It involves four main modules: DM, UM, AM (from AEHSs general architecture) and RTE Sequencer from SCORM LMSs architecture. Adaptive functionality can be provided in two ways: AN using the system's Adaptation Module and AP and/or some AN technologies by taking advantage of SCORM functionality at the course authoring phase. Both formative [3] and summative evaluation of ProPer have completed. Formative evaluation results demonstrated that students using ProPer can navigate more goal oriented into a course, avoiding unnecessary concepts. As a result a course can be accomplished faster. Furthermore, summative evaluation results verified that ProPer can improve the learning outcome in comparison with a simple hypermedia system. In addition, users stated that they like studying with ProPer and they find it simple and useful. Writers are also found ProPer very easy and useful for their courses delivery since it allows them design personalized instruction. At the same time, existent educational material can be easily retrieved, accessed and reused taking advantage of SCORM functionality.

SCORM not only excels in reusability issues but also allows authors to design both sequentional and content instructional strategies. In particular, authors can create educational content in the form of HTML pages. These pages have to include the appropriate JavaScript to communicate with the LMS. Moreover, every SCORM conformant course must contain an xml manifest file, which includes all the data for course structure, SN. In previous work we have shown that SCORM standard allows compliant courses to provide adaptivity according to user learning styles [3]. However, as is known, designing a course that is adaptive to user learning style is not an easy task. Although a framework for applying learning styles to SCORM compliant courses has been proposed [4], the implementation of these adaptive courses requires that authors have programming skills and deep knowledge of the SCORM standard. Therefore, we decided to design an authoring tool that allows non-programmer authors to create SCORM compliant and/or adaptive courses.

5 ProPer SCORM Authoring Tool (SAT)

ProPer SAT is a simple online SCORM authoring tool for both web page design and course package creation enabling tutors to effectively and efficiently design SCORM compliant adaptive courses. Its focus is on simplicity with the intention of attracting instructors with a low technological background. The main goals of ProPer SAT are: Help instructors compose SCORM compliant courses; Provide course patterns for easy authoring of adaptive courses to user learning styles; and Enable easy reusability for both SCOs and content fragments.

In order to keep the tool simple we do not provide advanced sequencing options with prerequisites and conditions for SCORM compliant courses. For these actions

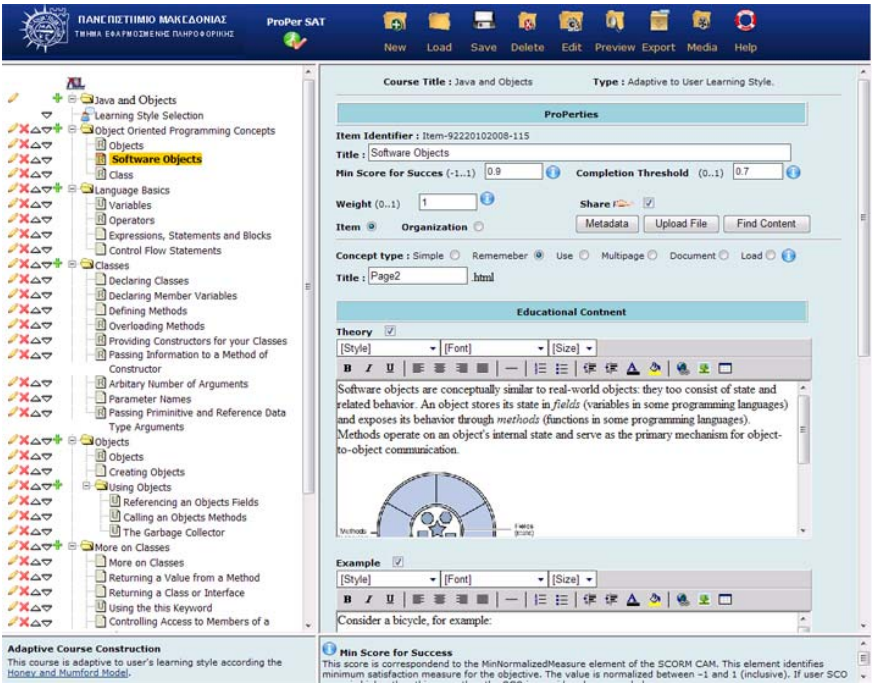

Fig. 1. The ProPer SAT interface

the author may use a dedicated SCORM package editor, such as Reload. On the other hand, the prototype does support content writing, course structure construction and course packaging functionalities. Moreover, the system enables instructors to easily create adaptive courses according to the Honey and Mumford model through a guided authoring process based on a framework we have proposed elsewhere [4].

The ProPer SAT interface is comprised of four main frames (Fig. 1). At the top of the page the toolbar functions are related to the whole course. On the left- hand side, a tree view of course structure and possible actions relative to course items/organizations is presented. The screen's main frame contains all the appropriate properties of course item/organization. In the case where the current selection is an item (SCO), WYSIWYG editors appear that enable the author to easily compose the educational content. The bottom of the screen is divided into two frames that contain information on the course and help about available options.

Authoring

Particular care was taken to ensure that authoring of courses remain as simple as possible. An initial screen (Fig.2) allows the author to give a title and select the course type. So far there are two types available: Simple SCORM courses without adaptive presentation and courses adaptive to user learning style. The latter adopts the Honey and Mumford model whose implementation is used by ProPer and INSPIRE. Beside the course type options a simple explanation is displayed. Moreover, the author can select one of the predefined templates for the course interface. A picture with a

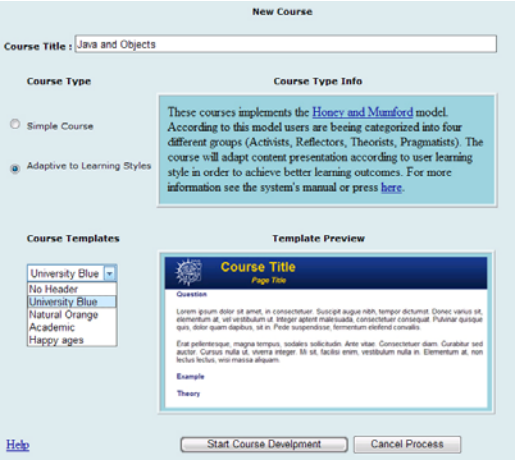

Fig. 2. New course options

preview of the selected template is displayed on the right hand side of the screen for better usability. Once the author has made their selections they can begin creating the course by pressing the appropriate button at the bottom of the screen.

In the next step a screen identical to Figure 1 is displayed with the initial course structure. For a simple course initially only the title with the "add" option (+ symbol) appears in the tree view. In the adaptive type courses, however, a predefined SCO for learning style selection has been added, which now allows authors to add items or organizations to their course. For every new item the author has to define item properties or leave the default as is. The user can also load an older SCO from the system's repository. With this option the system enables the author to find and use a SCO from other courses and/or authors. Once the SCO is inserted the author can proceed by manipulating both its properties and contents. Authors can also provide some of the metadata information that SCORM standard proposes. This metadata can later be read for the retrieval and reuse of relevant educational material.

Following this, the author starts the content orientation. A choice has to be made between three or five content types for simple and adaptive courses respectively. The available content types are:

- Simple: where the user writes the content in a WYSIWUG editor;
- Multipage: when a SCO constitutes more than one page. Once the user selects this option he/she must state the page number and the appropriate number of WYSIWYG editors is presented for content writing;
- Document: where the produced page presents the content of a document, such as Word, Excel, PowerPoint etc.; and
- Remember or Use: these types are available only in adaptive courses. They allow the user to create content for different knowledge modules (Theory, Example, Question, Activity) within a SCO. These modules are presented in a different order according to the user learning style. The module selection and their adaptive presentation are based on the framework presented by Papanikolaou [11]. For instance, the modules' order in a Remember page for an Activist user will be Question,

Example, Theory; while for a reflector it will be Theory, Example, Question. A typical screen for the creation of a Remember content type is displayed in Figure 1.

During course creation the author can view the page by selecting the Preview button at the top of the screen. Once SCO creation has been completed it can be saved and a new one added through the icon on the course structure frame. The same procedure continues until completion of course construction. Furthermore, the author may select an item from the course structure frame in order to change its properties and/or content. Items can also be deleted or their positions changed. A noteworthy feature is that each item in the course structure frame is adaptively annotated. The course can be exported to a ZIP archive according to the SCORM 2004 3rd edition standard. Authors can either upload the course into any SCORM compliant LMS or proceed to further changes. The changes can be relative to the package, SN of the course or related to page content. These changes can be made with common SCORM package software and an HTML editor, such as Reload and Dreamweaver respectively.

System architecture and implementation

The system adopts a three tier architectural design (Fig. 3). The first tier is the system's front end, which is the client side of the prototype and involves the user interface. The middle tier contains the system's intelligence made up of the **Domain Model** (DM) and the **Course Production Module** (CPM). More specifically the DM reads data from the interface; it also forms and stores it onto the database. The CPM reads data from the database and dynamically creates the user interface. It communicates with the DM and the database, interprets user preferences into the appropriate XML or HTML code and creates all the required course files. Following, it compresses these files into a ZIP archive and delivers it to the user.

Lastly, Data Storage is the back end tier and contains the system's database as well as a file server, whose files fall into three categories:

- SCORM package files necessary for the creation of a SCORM package. These are copied as they are in the exported SCORM course packages. SCORM standard requires the existence of some standard files in every compliant course. Moreover, the produced courses are based on specific JavaScript and CSS files according to the selected type and template respectively. These files pre-exist in the system's file server.

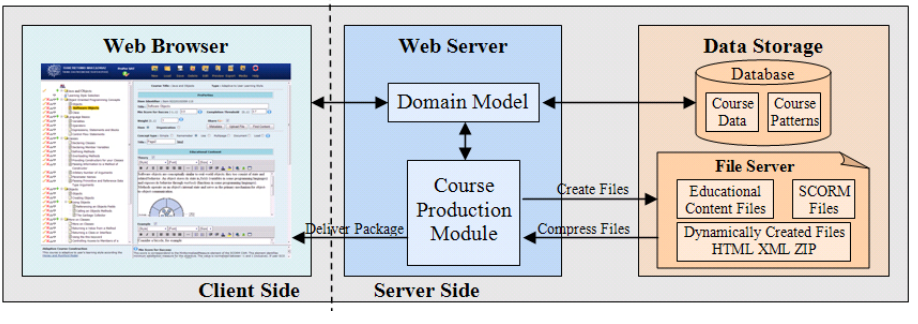

Fig. 3. System architecture

- Educational content files (photos, PowerPoint documents etc.) uploaded from users to appear on course pages. When users want to insert a photo or a presentation etc. into their course, they first must upload the file and later state it in the appropriate course SCO.
- Dynamically created files (HTML, XML and ZIP files) according to user preferences and submitted educational material. When the user chooses to export a course the system dynamically creates the HTML files with the course content and the imsmanifest.xml file necessary for the course package. In the final step it creates a zip file containing all the appropriate course material.

Apache Tomcat 5.5 used as a web and application server and MySQL 5 as a database server. Java Server Pages (JSP) and Java Servlets used for the middle tier intelligence while HTML, CSS and JavaScript for the front tier development.

6 Conclusion and Future Works

In this paper we have presented ProPer SAT, a tool for the efficient and effective construction of SCORM compliant and/or adaptive courses. Compared with the related work ProPer SAT and ProPer can create and deliver courses adaptive to user learning style in the same way that AES-CS and INSPIRE do, providing a tool for quick course production. Further, ProPer SAT is able to construct simple SCORM compliant courses along with their required educational content. ProPer SAT has an advantage over other simple authoring tools, such as MOT, VIDET etc. for its simplicity and SCORM standard adoption. Moreover, most of the SCORM compliant LMSs do not provide a tool for easy course construction nor do they support adaptive features as ProPer does. Even if advanced users can employ tools, such as Reload or Lectora etc., for course packaging and content development, our system provides an easy way for basic function implementation without needing to have programming skills or detailed knowledge of SCORM. In sum, ProPer and ProPer SAT offer a promising solution for non-technical teachers who want a simple way to be able to create and deliver SCORM compliant and/or adaptive courses quickly and easily.

ProPer SAT is still in the process of development, therefore, some features have not as yet been implemented. Despite our conviction that its prototype, as it stands, is an exceptionally useful tool for teachers, more functionality needs to be added. Our goal is not to make this tool support the implementation of all the SCORM functionalities, however, a possible use of SCORM rules and conditions is under study. ProPer SAT will help authors create adaptive courses and define sequencing and navigation procedures. Moreover, some additional course-content types, such as tests, can be added to support user assessment. Finally, a pilot operation and evaluation of system effectiveness will follow in order to check for any system weaknesses, as well as assess the system's functionality, usability and usefulness. It is imperative to note, however, that the simplicity of the authoring process, which is the driving principle behind the development of ProPer SAT will not be compromised to any extent, irrespective of any future changes that may be warranted.

References

1. ADL, SCORM 2004 4th edn., Overview - Sharable Content Object Reference Model (2004), http://www.adlnet.gov (accessed April 2009)
2. Kazanidis, I., Satratzemi, M.: Adaptivity in a SCORM compliant Adaptive Educational Hypermedia System. In: Leung, H., Li, F., Lau, R., Li, Q. (eds.) ICWL 2007. LNCS, vol. 4823, pp. 196–206. Springer, Heidelberg (2008)
3. Kazanidis, I., Satratzemi, M.: Adaptivity in Pro Per: an adaptive SCORM compliant LMS. Journal of Distance Education Technologies 7(2), 44–62 (2009)
4. Kazanidis, I., Satratzemi, M.: Applying learning styles to SCORM compliant courses. In: 9th IEEE International Conference on Advanced Learning Technologies (to be published) (2009)
5. Brusilovsky, P.: Adaptive hypermedia. User Modeling and User Adapted Interaction. In: Kobsa, A. (ed.) Ten Year Anniversary Issue, vol. 11(1/2), pp. 87–110 (2001)
6. Honey, P., Mumford, A.: The manual of learning styles. Maidenhead, Berkshire (1992)
7. Kolb, D.: Experiential Learning: experience as the source of learning and development. Prentice-Hall, New Jersey (1984)
8. Felder, R.M., Silverman, L.K.: Learning Styles and Teaching Styles in Engineering Education. Engr. Education 78(7), 674–681 (1998)
9. Witkin, H., Moore, C., Goodenough, D., Cox, P.: Field-dependent and field-independent cognitive styles and their educational implications. Review of Educational Research 47(1), 1–64 (1977)
10. Gardner, H.: Multiple Intelligences: The theory in practice. Basic Books, New York (1993)
11. Papanikolaou, K.A., Grigoriadou, M., Magoulas, G.D., Kornilakis, H.: Towards new forms of knowledge communication: the adaptive dimension of a Web-based learning environment. Computers & Education 39, 333–360 (2002)
12. Carro, R.M., Pulido, E., Rodríguez, P.: Dynamic generation of adaptive Internet-based courses. Journal of Network and Computer Applications 22, 249–257 (1999)
13. Bajraktarevic, N., Hall, W.: Fullick. P.: Incorporating learning styles in hypermedia environment: Empirical evaluation. In: Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2003), pp. 41–52 (2003)
14. Carver, C.A., Howard, R.A., Lavelle, E.: Enhancing student learning by incorporating learning styles into adaptive hypermedia. In: ED-MEDIA 1996, pp. 118–123 (1996)
15. Triantafillou, E., Pomportsis, A., Demetriadis, S.: The design and the formative evaluation of an adaptive educational system based on cognitive styles. Computers & Education 41(1), 87–103 (2003)
16. Kelly, D., Tangney, B.: Adapting to intelligence profile in an adaptive educational system. Interacting with Computers 18, 385–409 (2006)
17. De Bra, P., Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, 115–139 (1998)
18. Popescu, E.: An Artificial Intelligence Course Used to Investigate Students' Learning Style. In: Li, F., Zhao, J., Shih, T.K., Lau, R., Li, Q., McLeod, D. (eds.) ICWL 2008. LNCS, vol. 5145, pp. 122–131. Springer, Heidelberg (2008)
19. Stash, N., Cristea, A., de Bra, P.: Authoring of learning styles in adaptive hypermedia: Problems and solutions. In: WWW 2004 Conference, pp. 114–123 (2004)
20. Armani, J.: VIDET: a Visual Authoring Tool for Adaptive Websites Tailored to Non-Programmer Teachers. Educational Technology & Society 8(3), 36–52 (2005)
21. Ainsworth, S.E., Major, N., Grimshaw, S.K., Hayes, M., Underwood, J.D., Williams, B., Wood, D.J.: REDEEM: Simple Intelligent Tutoring Systems From Usable Tools. In: Murray, T., Blessing, S., Ainsworth, S. (eds.) Authoring Tools for Advanced Technology Learning Environment, pp. 205–232. Kluwer Academic Publishers, Dordrecht (2003)