20

Towards the integration of adaptive educational systems with SCORM standard and authoring toolkits

Ioannis Kazanidis and Maya Satratzemi
University of Macedonia
Greece

1. Introduction

Web courses and hypermedia systems deliver knowledge to a wide number of users with different characteristics, preferences and knowledge of the domain, irrespective of where they live, their age or their study credentials. However, these systems do appear to have some quite major problems which have been identified and documented through research studies and have differentiated into three distinct categories. The first deals with problems related to disorientation, cognitive overload, discontinuous flow (Murray et al., 2000), content readiness and distraction. The main solution that research proposes is the use of adaptive and/or interactive systems. The second category of problems is those that arise from the absence of a common development framework for course construction. Course content, thus, lacks reusability, durability and interoperability. A suggested solution is the adoption of common educational standards for course construction and delivery. The third category involves instructors who come up against difficulties during course construction as most of the time course development requires not only specific programming capabilities but also deep knowledge of adaptive strategies and educational standard specifications. Easy authoring tools for non-programmer instructors may reduce these difficulties and allow more people to create easy and fast web and/or adaptive courses.

The present work deals with three aspects of web learning systems: adaptivity, adoption of educational standards, and authoring tools. Initially it discusses the problems of web-based courses at both the construction and delivery stages, and the possible solutions. It then introduces the reader to the theoretical background in the area of adaptive learning systems with emphasis on user modelling, adaptation technologies, and learning style models. A discussion on educational standards and their usefulness follows with focus on SCORM and its specifications. Next, it reviews adaptive systems and their functionality. It also presents systems that provide adaptive features according to user learning style and SCORM compliant systems. Finally, this chapter deals with the available authoring tools that support either the construction of adaptive or SCORM compliant courses.

Special emphasis is given to the functionality of a new system we have developed, named ProPer, which implements all the above aspects. ProPer combines adaptive technologies with the adoption of SCORM standard and is also accompanied by an authoring tool of SCORM compliant and/or adaptive courses that allows easy course construction without any programming knowledge prerequisites.

2. Problems and solutions of hypermedia courses

Web courses can be used either as an independent integrated solution for distance learning or as supplementary material for conventional classroom education. Furthermore, web courses are available to a broad range of users with different characteristics, preferences, educational goals and knowledge of the domain. Users can follow different navigational paths and study courses with alternative structure and content according to their goals and individual needs (Murray et al., 2000).

Some of the above abilities, however, can actually become drawbacks for these courses since it is not reasonably practicable to cover the range of needs and preferences of every individual user. Brusilovsky et al. (1998) agree with this conviction and state that a course or system that has been designed for a particular class of learners may not suit learners of another class. In addition, research has documented the following major problems of web-based courses: disorientation, cognitive overload, discontinuous flow, content readiness (Murray et al., 2000) and distraction (Scheiter & Gerjets, 2007). More specifically, users are disoriented when they are uncertain of what they have or have not read; are not sure where to find the information they need (Chen et al., 2006); or simply when they become lost in hyperspace (Conklin, 1987). Disorientation is more likely to occur when the knowledge domain is too big or the learner is a novice. The availability of huge quantities of information or the large variety of options and functions a system provides can leave some users bewildered and overwhelmed (Murray et al, 2000; Ng et al., 2000). This constitutes the cognitive overload situation. The discontinuous flow problem distinguishes two issues: narrative flow and conceptual flow. Narrative flow refers to the way the text proceeds, while conceptual flow refers to the stream of ideas presented in the text (Murray at al., 2000). The content readiness problem or "prerequisite problem" arises when the learner finds the content either too easy and is bored, or too difficult and is overwhelmed (Murray at al., 2000). According to Foss (1989) distraction problems arise when the users find too many interesting things or too many relevant topics to explore that distract their attention away from the course's main goal.

Hence, the use of adaptive and/or interactive hypermedia systems were proposed as a promising solution (Brusilovsky, 1996; Prentzas & Hatziligeroudis, 2001). Adaptivity in e-learning is a new research trend which personalizes the educational process through the use of Adaptive Educational Hypermedia Systems (AEHS). These systems attempt to create an individualized course according to the user's personal characteristics, such as language, learning style, preferences, educational goals and progress. In this way, instructors expect to solve some of the main problems of web courses and hope to succeed in achieving a better learning outcome.

Apart from requiring authors to expend a significant amount of time, course development and more so the construction of an adaptive course often involves spending vast amounts of resources. 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. Moreover, the reuse of educational content and its recall require additional effort from course authors, not to mention that many times, the actual structure of the educational material changes due to the educational platform being upgraded. As a consequence, emphasis is placed on the development of the application where most time and effort is spent, at the expense of instructional strategies development (Sidiropoulos & Bousiou-Makridou, 2005). Adopting educational standards, like SCORM, comes as a solution to the above problems for content reusability, accessibility interoperability and durability. It is expected that the adoption of such a standard will help authors to construct more effective courses faster with less effort and at a lower cost.

Another problem of hypermedia systems lies in the lack of simple authoring tools for SCORM and/or adaptive courses that are appropriate for the non-programmer. This situation naturally, hinders instructors from constructing web-based courses. Thus, the construction process could also be improved with the use of authoring tools that allow fast and easy course construction even by non-programmer authors.

In accordance with the above, we believe that an integrated solution for web-based learning would concern the development of an AEHS which adopts the SCORM standard and provides a tool for easy course authoring.

3. Adaptive Educational Hypermedia Systems Background

By combining the tutor-driven learning process of Intelligent Tutoring Systems (ITS) and the flexibility of student centred Hypermedia Systems (HSs), AEHS have integrated several technologies from both these systems. More specifically, typical AEHSs include a Domain Model (DM) which represents the systems domain; maintain a User Model (UM) which records the user's personal characteristics and knowledge; and are able to adapt the course structure and presentation according to the UM (Eklund & Brusilovsky, 1998) through the Adaptation Module (AM). These systems provide personalized training according to UM records through a set of adaptive rules prescribed by course instructors. They adapt the learning process in a way that enables users not only to learn better but also much easier and faster. Thus, through the use of various techniques, adaptive courses initially and/or during the learning process acquire the requisite user information and store it on the UM. Through a set of adaptive rules, AEHSs use Adaptive Presentation (AP) and Adaptive Navigation (AN) technologies so as to provide various instructional strategies that are personalized to each individual learner.

3.1 Adaptation factors

The exploration of user characteristics in order to ensure the provision of adaptation is an important issue for the success of AEHS. An ideal system would be one that adopted every factor that can affect both the learning process and the user's progress in a course (Wegner, 1987). However, due to difficulties in representation, the large effort that is required by course designers, as well as the complexity of implementation, results in only a few specific factors in practice being modelled by the systems (Kavcic, 2000). The most popular of these, according to Brusilovsky & Millan (2007) are: user knowledge, educational goals, preferences, background and experiences, personal traits (learning style, aptitude), and technology infrastructure.

- **Knowledge.** User knowledge of the domain is one of the most important features that is modelled by AEHSs (Kavcic et al., 2002; Brusilovsky & Millan, 2007; Brusilovsky 2003).

Estimated user knowledge can initially be retrieved through a set of questions (pre-test) or even directly by user declaration. The user's acquired knowledge during the learning process can be retrieved either through assessment questions or by the system's observation of the user's behaviour throughout the course. A variety of techniques and models can be implemented for knowledge representation, such as the overlay model, the differential model, the perturbation model etc., which are presented in a later section of this chapter.

- **Educational goals.** The modelling of this factor allows learners to have individual educational goals in a course or a system (Brusilovsky, 2003). For instance, they may choose a subset of domain concepts as their current educational goal. During the learning process, the student may have a primary goal that consists of several predefined elementary goals. Goal orientation can be carried out either by the course author, the teacher or the learner (Brusilovsky, 2003) itself.
- **Preferences.** Each user may prefer some nodes, links or page fragments over others (Brusilovsky, 1996). User preferences are mainly applied for the course's adaptive presentation. They can be retrieved either directly through the user state or indirectly through observation of user behaviour.
- **Background and experiences.** These concern the user's previous experience that may be relative but outside the core domain of a system (Brusilovsky & Millan, 2007). User experience may concern user familiarity with hypermedia systems and web or likely knowledge of technical definitions used in a course. Systems mainly use background and experiences to adapt course content.
- **Personal traits.** The user's cognitive and learning styles, as well as their aptitudes comprise some personal traits, which many systems take into consideration when adapting their courses (INSPIRE, AES-CS, CS383, TANGOW etc.). Personal traits can be retrieved through specially designed psychological tests (Brusilovsky & Millan, 2007). Learning style theories and models are further expanded on in the next section.
- **Technology infrastructure.** This is an important factor for distance and online learning. For instance, low connection speed may produce problems in the study content based on video. Moreover, content presentation needs to be adapted according to the user device (PDA, mobile, PC etc.).

3.2 Technology Background

According to Eklund and Brusilovsky (1998) all the adaptive hypermedia systems comprise a DM, which consists of elementary pieces of knowledge and their relationships, a User or Student Model as it is referred to in many works, that is responsible for the recording of the factors which adaptation depends on, and an AM, responsible for system adaptation according to the UM.

3.2.1 Domain Model

The role of the DM is to represent the knowledge that is to be taught (Wegner, 1987). Various techniques have been used for knowledge representation,, however, a formal DM is commonly comprised of three layers.

The first layer is made up of the concepts of the cognitive domain. The breadth of knowledge that covers each concept is differentiated depending on the size of the domain, the thematic region and the choices of the course designer (Brusilovsky, 2003). The concepts can be independent, in the simpler form of the DM, or related to each other, thus forming a conceptual network which represents the structure of the domain (Brusilovsky, 2003). A simple relation between two concepts is the link connection that leads to the page of the corresponding concept. One of the most popular relations is the prerequisite one. This is used when it is considered that it would be good for users to know certain concepts before applying others that it is an essential precondition to studying the relative educational material (Prentzas and Hatziligeroudis, 2001). Other relations are: "part of" where certain concepts are part of a more complex concept; and "is a" where concepts constitute characteristic cases of another concept (Prentzas and Hatziligeroudis, 2001).

Each concept may correspond to one or more web pages, which constitutes the second layer of the DM. Each author can choose whether a concept's educational material will be presented in one or more web pages. The passage from a web page to another can be either sequential, by selection and/or adaptive.

The final layer of the DM is comprised of smaller cognitive fragments, such as text, picture, animation, video etc. Each web page constitutes at least one of these fragments. The choice of the fragments that will be presented in each web page can be either static, where it remains the same, or dynamic where the fragments are automatically selected according to user properties. A dynamic web page is able to adapt the presentation of a fragment itself or even be composed of a combination of different fragments.

3.2.2 User Model

The UM constitutes a representation of user knowledge, preferences, characteristics and educational goals (Brusilovsky, 1996). An ideal UM should include all the properties that influence the user's learning and progress within a course (Wenger, 1987). According to Kavcic (2000) three aspects have to be considered with regard to the UM: (i) the information that will be stored in the UM and how it can be retrieved; (ii) the representation of this information in the system; and (iii) the process of forming and updating the model. The representation of information in the UM can be achieved through a variety of methods. Some knowledge representation models include the scalar model, the overlay model, the differential model, the perturbation model etc. presented further below. The methods for forming and updating the UM are similar to those for constructing it (Kavcic, 2000).

The information stored in the UM can be separated into static and dynamic depending on the system's retrieval mechanism (Kavcic, 2000). Static information remains immutable during the learning process (unless the user decides to redefine it); it is collected once and refers mainly to the user's particular characteristics, such as language, occupation and learning style. On the other hand, dynamic information can be altered during learning and requires continuous updating. Information of this category may include user knowledge, progress, goals etc. Modelled information can be either domain-independent or domain dependent. The former contains information regarding user properties which usually remains stable during learning, while the latter contains information related to the DM, such as user knowledge, educational goals etc.

Knowledge representation models

As previously mentioned, user knowledge is one of the main factors of adaptation. Some of the proposed models for knowledge representation are the following:

- **Scalar Model**. The simplest form of knowledge representation is via scalar models. These models can be quantitative, evaluating the user's knowledge on a scale (e.g. from 1 to 5) or qualitative, classifying the user into stereotypes, such as novice, intermediate, advanced according their knowledge level (Brusilovsky & Milan, 2007). However, scalar models represent user knowledge for the entire course. In cases where user knowledge is different for particular concepts of the course, then the use of structural models is needed.
- **Overlay Model.** This is the most popular of the structural models. It is based on the assumption that user knowledge constitutes a subset of expert knowledge in the domain (Beck et al, 1996). A user may acquire knowledge with the ambition to reach the level of an expert but s/he cannot learn something more or different from this. The overlay model keeps a value for every DM concept, which represents the coverage of expert knowledge in that particular concept.
- **Differential Student Model**. The differential model improves on the overlay model (Holt et al., 1994). It does not view all gaps in student knowledge as necessarily undesirable. Similar to the overlay model, user knowledge is a component of expert knowledge. The difference, however, lies in the fact that knowledge in this model is divided into two categories: (a) expected knowledge, and (b) knowledge that the learner could not be expected to attain. Accordingly, it is not necessary for the former to be the expert's knowledge but can comprise its subset.
- **Perturbation Model**. The two structural models stated above cannot represent likely knowledge queries or misconceptions that the user may have outside the boundary of expert knowledge. In this model learner knowledge and aptitudes are considered to be a perturbation of expert knowledge rather than a subset (Martins et al., 2008). The perturbation model extends the overlay model by representing possible user misconceptions also called faulty or buggy knowledge. More specifically, it combines the overlay model with a representation of faulty knowledge.
- **Genetic Graph.** The previous models give an instant representation of user knowledge. In contrast to the overlay model, which besides representing user knowledge as a part of expert knowledge also deals with the time involved in knowledge gain, the Genetic Graph (Goldstein, 1982) records the time factor involved in the process of knowledge development
- **Bounded Model.** (Elsom-Cook, 1988) This can be considered as a variant of the overlay model which rather than representing the exact user knowledge acquired, employs fuzzy bounds by setting a low and high limit.

Instead of using one specific model, many systems tend to combine functions and properties from two or more models. One of the most usual combinations is to initially classify learners using stereotypes and later during the learning process construct an overlay model for a more individualized adaptation.

3.2.3 Adaptation Module

The Adaptation Module (AM) includes a set of rules and conditions, which through the application of various adaptive technologies individualizes the course to each individual user. More specifically, AM is triggered by specific user actions and initially reads information from the DM and UM. Following, through its set of rules, it checks for any possible adaptation needed. If this is the case, it uses the appropriate adaptation technologies to adapt navigation and/or presentation of the system while at the same time updates the UM with the new data.

The rules that the AM contains can be applied before the delivery of a new page (pre-conditions), for instance, when the system checks if the user knows the prerequisite concepts. They can also be applied (a) at the page exit (post-conditions), e.g., the UM is updated with the user's latest knowledge acquisition of a concept; or (b) during the study of a page, e.g., if the test score is above a specific limit then the user is allowed to study the following concept.

Various adaptation technologies are implemented to personalize the system. These can adapt either the content of a web page or the links provided. Thus, in accordance with the Brusilovsky taxonomy (Brusilovsky, 1996, 2001), adaptation technologies can be separated into two major categories: Adaptive Navigation and Adaptive Presentation.

There are three main **Adaptive Presentation technologies**:

(i) Adaptive multimedia presentation. Sometimes the use of multimedia may create problems for users due to their not having adequate technological infrastructure. This technology adapts the quality and size of the multimedia that is used in a course according to the user infrastructure.

Adaptive text presentation. This allows the adaptation of text to user preference. It is further divided into Natural language adaptation and Canned text adaptation which is the most usual. Canned text adaptation according to Brusilovsky (2001) uses the following technologies:

- Inserting/removing fragments: System inserts or removes specific text fragments.
- Altering fragments: System provides many alternative text fragments and only the most appropriate is presented.
- Stretchtext: Additional text or explanations can be collapsed or un-collapsed on a page depending on user preferences.
- Sorting Fragments: the sequence of text fragments can be changed.
- Dimming Fragments: system dims or fades text that according to the UM is inappropriate for study, without completely removing it.

Adaptation of modality. Some times the same concept is presented in different media, like text, video, sound etc. This technology chooses the best media for concept presentation and delivers it to the user.

On the other hand, **Adaptive Navigation** aims to help the user find the optimal path within a course (Brusilovsky, 1996). The main AN technologies (Brusilovsky, 2001) are:

- direct guidance - the system proposes the next best concept for study;
- link sorting – adapts the order of the links;
- link hiding – the system presents links as simple text (hiding), disables links (disabling) or totally removes them (link removal);
- link annotation - links are annotated corresponding to the user model;
- link generation – the system generates new links;
- hypertext map adaptation - the system provides and appropriately changes a graphical representation of the link structure.

3.3 Learning Styles

Each individual has his/her unique way of learning. Thus, learning style greatly affects both the learning process and the outcome (Carver and Howard, 1999). 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 in order to gain a better learning outcome. Some of the most well known learning styles are:

- **Kolb's learning style theory** (Kolb, 1984). Learning is a process of knowledge construction through a cycle of four distinct stages: Concrete Experience (CE), Reflective Observation (RO), Abstract Conceptualization (AC) and Active Experimentation (AE). The student can start from any point in the cycle and continue going through the remaining stages in sequence. This model classifies learners into four categories represented by the combination of two preferred styles: Divergers (CE, RO), Assimilators (AC, RO), Convergers (AC, AE) and Accommodators (CE, AE). Users are placed in one of the above categories by answering a 12-sentence questionnaire describing how they best learn (Kolb, 1981).
- **Honey and Mumford** (1992) **Model**, which is based on Kolb's learning style theory. The four stages of the learning cycle are: (i) having an experience; (ii) reviewing the experience; (iii) forming conclusions from the experience; and (iv) planning the next step. Similar to Kolb's circle the student can start from any stage and continue to the others. Each stage corresponds to a related learning style: Activist, Reflector, Theorist and Pragmatist. Learners are classified with an 80-item- true/false questionnaire (Honey and Mumford, 2000).
- **Felder & Silverman** (1988) **Learning Style Model (FSLSM)**. This model classifies learners by using a five dimensions sliding scale: sensing-intuitive, visual–verbal, inductive-deductive (similar to Kolb's CE-AC), active-reflective (similar to Kolb's AE-RO), sequential-global. However, the inductive-deductive dimension is abandoned with the development of the assessment instrument (Felder & Soloman, 1996). The student learner style can be acquired through the Index of Learning Style (ILS) questionnaire (Felder & Soloman, 1996) which consists of 44 two choice answers.
- **Witkin's Field Dependent- Field Independent Model** (Witkin et al., 1977). Field dependent individuals are global on their perceptions, intrinsically motivated, enjoy cooperative learning and require externally defined goals. On the other hand, field independent people are more analytical in their approach, enjoy individualized learning and tend to develop self oriented goals (Triantafilou et al, 2002). Learner's style can be defined through the Group Embedded Figures Test (Witkin et all, 1971).
- **Dunn & Dunn** (1978) **Model**. This model is based on the theory that individuals have unique sets of biological and developmental characteristics that impact on the way of learning. It involves 21 elements that are grouped into the following five "stimuli" categories: environmental, emotional, sociological, physiological and psychological preferences. The first versions of learning style inventory for this model were developed for children and consisted of 104 Likert scale questions, while the current adult version (Rundle & Dunn, 2000) is comprised of 118 five-point Likert questions.
- **Grasha-Riechmann Student Learning Styles Scale (GRSLSS)** (Riechmann & Grasha, 1974). This model classifies three bipolar dimensions in respect to students' social interaction: competitive-collaborative, avoidant-participant, dependent-independent.

To find the student's learning style, GRSLSS uses the Student Learning Styles Scale (Grasha & Riechmann, 1975) of 90 questions, 15 for each subcategory.

- **Gardner's theory of multiple intelligences** (Gardner, 1993). Gardner identifies eight aptitude-like traits, which he refers to as "intelligences": mathematical-logical, musical, linguistic-verbal, visual-spatial, bodily kinesthetic, interpersonal and intrapersonal. A further intelligence has been added called naturalistic. A multiple intelligence inventory by Shearer (1996) is available.

4. Educational standards and SCORM

Authors would save much time and effort if they could easily find and reuse qualitative educational content from other courses and/or platforms. Moreover, they would save time if there where no need to update their courses when the host platform was updated to a new version. Thus, the need to have reusable, accessible, interoperable and durable (RAID) content has led to the creation of learning technology specifications. For the time being the most popular educational standard is SCORM (Sharable Content Object Reference Model) (ADL, 2009) which was implemented by the ADL (Advanced Distributed Learning) Initiative.

SCORM is a collection of specifications and standards for the development, packaging and delivery of educational content. More specifically, it describes the components used in learning and how to package them for exchange between compliant systems; how they should be described using metadata in order to enable search and discovery; and how to define sequencing rules for the content objects (ADL, 2009). SCORM consolidates the work of other standards and organizations, such as ARIADNE, AICC, IMS, and IEEE's LTSC into one unified reference model. The application of SCORM ensures the reusability, accessibility and durability of the educational material, as well as interoperability between learning management systems (LMS).

SCORM Components

SCORM is comprised of three main components (ADL, 2009): Content Aggregation Model (CAM), Run-Time Environment (RTE), and Sequencing and Navigation (SN).

CAM describes the format of content structure, how to package, describe and identify them, and how to define sequencing information (ADL, 2009). It is made up of five components (ADL, 2009): (i) Assets: electronic representations of media, such as text, images etc.; (ii) Sharable Content Objects (SCOs) which are collections of one or more assets. It must be noted, however, that SCOs are differentiated from assets by their ability to communicate with the LMS using the IEEE ECMAScript Application Programming Interface (API); (iii) Activities: structured units of instructions. These may provide either a SCO or an Asset to the learner or even be composed of subactivities; (iv) Content Organization: representation that defines the intended use of the content through the activities; (v) Content Aggregation: describes the composition process of related content objects so that the set can be applied in a learning experience.

RTE provides a means of interoperability between SCOs and LMSs. Its goal is the interoperability of educational content between different LMSs, independent of how these were developed. In order for this to be achieved there must be a common content launch process, communication with the LMS and predefined tracking data elements that are exchanged between the LMS and content objects. The launch process defines a common way for content object initialization. The communication of content objects with the LMS is performed via an API used both for retrieving and storing data between LMSs and SCOs. A Data Model is used for the definition of the information being tracked for a SCO (ADL, 2009).

The SN describes how content objects may be delivered to the learners through a set of navigation events and how to control the delivery sequence using run-time based programming rules. SN includes subjects such as sequencing concepts and terminology, sequencing definition and behavior models, navigation controls and requirements, and a navigation data model (ADL, 2009).

A SCORM compliant course is made up of a collection of assets, such as, images, text etc., one or more SCOs and an XML manifest file that is found at the top of the course folder and stores data not only about the course structure but sequencing and navigation rules as well.

The application of SCORM standard allows instructors to reuse educational material and thus save time on course development.

5. Related Work

The first pioneer adaptive systems were implemented in the early 1990s. Brusilovsky classifies the systems that have been developed since then into three generations (Brusilovsky, 2004). The first generation systems (1990-1996) were experimental and were developed to explore innovative ideas. The second (1996-2002) were developed to be used in real life problems (Brusilovsky, 2004). Some of the best-known, such as ELM-ART, TANGOW, AHA! and Interbook were developed between 1996-2002. The second generation research can be split into three main streams (Brusilovsky, 2004): (i) systems that re-used existing technologies and explored a number of approaches and various subject areas; (ii) work on producing new adaptive hypermedia techniques, like adaptation to student learning style; and (iii) research on frameworks and authoring tools for adaptive hypermedia development. Despite the fact that these second generation systems tried to solve real life problems, they failed to influence practical Web-based education (Brusilovsky, 2004) which relies on LMSs. The third generation of AEHS attempts to compete with these systems following three alternative streams: (i) systems which incorporate as many functionalities of LMSs as possible; (ii) systems working on open corpus web content; and (iii) systems that focus on interoperability and reusability of educational content using appropriate educational standards, such as SCORM.

Our work is based on three of the previous presented research streams. Principally, our aim has been to combine adaptive hypermedia with the SCORM educational standard. Moreover, our prototype, called ProPer, supports adaptation to various user learning styles with an evaluated application of the Honey and Mumford learning style theory. A framework for the creation of SCORM compliant courses that are adaptive to learning styles has been proposed in (Kazanidis & Satratzemi, 2009b). Furthermore, we are working on an authoring tool, called ProPer SAT, for the development of SCORM compliant and/or adaptive courses.

This section deals with the areas that we are working on presently. Firstly, the area of AEHS is presented. Particular emphasis is given to the second-generation systems that exploit learning style theories in order to achieve a better learning outcome, as well as on the authoring tools of adaptive courses. Next, we present third generation systems, which combine adaptation with the adoption of educational standards. Lastly, authoring tools for SCORM compliant and/or adaptive courses are presented.

Adaptive systems

Many adaptive systems are similar to ProPer. Following are some such:

ISIS-Tutor (Brusilovsky & Pesin, 1994) is an intelligent tutor for the CDS/ISIS library system that was developed by UNESCO. It belongs to the first generation AEHS and applies adaptive link annotation, hiding and removal, as well as direct guidance using an overlay student model. ELM-ART (Brusilovsky et al, 1996) is a second generation adaptive system and many subsequent models have their roots in it. It is a hyperbook which also provides adaptive link annotation and sorting, index-based guidance and problem solving support in order to assist programming in LISP. It was the first to use the adaptive link annotation technique. For user modelling it employs a multilayer overlay model. One system based on ELM-ART is KBS-Hyperbook (Henze & Nejdl, 1997). It is an open hypermedia system, which delivers content from information resources located anywhere in the WWW. KBS-Hyperbook follows the constructivist educational process building on project based learning, group work and discussions (Henze & Nejdl, 1997). It uses Bayesian networks for user modelling and provides direct guidance and adaptive link annotation. Another well-known AEHS is ALICE (Kavcic et al., 2002). It is an electronic textbook on the Java programming language, which uses link insertion, link annotation and direct guidance as its main adaptation techniques. For user modelling it includes the elements of knowledge uncertainty.

Adaptive systems focused on learning style

While the systems presented above endeavour to individualize the educational process to user knowledge and navigational history, some others adapt instruction to user learning style. INSPIRE (Papanikolaou et al., 2003) generates adaptive courses that provide adaptive link annotation and adaptive presentation of the educational content according to user knowledge level and learning style. It presents the same educational material in a different sequence of knowledge modules (e.g. activities, examples, hints from theory, exercises) for particular learning styles. It adopts the Honey and Mumford (1992) learning style model and categorizes students accordingly through an appropriate questionnaire (Honey and Mumford, 2000). AES-CS (Triantafillou et al., 2002) incorporates the FD/FI learning style model. It adapts content presentation and provides navigation support according to the user's prior knowledge and cognitive style. Cognitive style is used in order to provide adaptive learner control, contextual organisers and lesson structure support. TANGOW (Carro et al, 1999) implements the sensing-intuitive dimension of the FSLSM. It presents the contents by example and theory similar to INSPIRE. The ILS questionnaire (Felder and Soloman, 1996) is applied to classify user into their corresponding learning style. If the questionnaire results are balanced, the default order defined by the designer is presented; appropriate adaptivity is also provided. Several dimensions of FSLSM (global-sequential, visual-verbal, sensing-intuitive and inductive-deductive) are also implemented by CS383 (Carver et al, 1996), which provides different types of media, such as text, movies, graphs, slideshows etc. User learning style is identified through the ILS questionnaire. LSAS (Bajraktarevic, 2003) is another system that incorporates the global sequential dimension of FSLSM. Sequential learners are provided with advanced organizers, more structured lessons and maximum instruction with feedback, while global learners are guided with an overview and summaries of the lessons (Stash et al., 2004). iWeaver (Wolf, 2002) is based on the Dunn and Dunn learning style model and for the adaptation of content presentation it incorporates two dimensions: the perceptive domain and the psychological domain. In addition, it provides specific learning tools in accordance to user preferences. User learning style is established through a questionnaire (Rundle, & Dunn, 2000). EDUCE (Kelly & Tangney, 2006) is a system based on Gardner's MI theory (Gardner, 1993). It classifies users through a questionnaire (Shearer, 1996), as well as dynamically during the educational process. EDUCE allows learners to study with/without adaptivity. While in the adaptive mode, the student is guided to a specific MI resource type but does have the option to go back and view alternative resources.

Course authoring tools

The existence of appropriate course authoring for the above systems is a crucial parameter. Most of them are available for specific domain knowledge but only programmers are able to develop additional adaptive learning material. However, there are adaptive systems that incorporate appropriate authoring tools to enable authors to develop various courses. Some of the most well known are the following: Interbook (Brusilovsky, et al., 1998) is a tool for authoring and delivering adaptive textbooks on the web. It provides adaptive guidance, navigation and help, following ELM-ART's adaptive methodology. InterBook addresses authoring by allowing tagged text to be imported from word processing files. NetCoach (Weber et al., 2001) is another authoring tool derived from ELM-ART. Authors can provide content in HTML with its presentation parameters and specify through appropriate forms prerequisite and inference links. AHA! (De Bra, et al., 1998) is an open-source software and addresses authoring through XML-based mark-up language, as well as through certain form-based authoring tools (Brusilovsky, 2003). It supports adaptive hiding of fragments, adaptive link annotation and hiding. Early modifications (Stash et al., 2004) enable AHA! to provide adaptivity to user learning style. It does not provide any one particular learning style support, but rather attempts to create enough flexibility to make it possible for authors to design many variations for learning styles (Stash et al., 2004). Learning style in AHA! is discovered by manual user declaration, instead of through a particular questionnaire. MetaLinks according to Murray (Murray et al., 2000) seems to be the only hypermedia system to have a fully featured GUI authoring system. Among other things, it can adapt content depending on where the learner came to this page from and it provides focused as well as exploratory learning. It should, however, be noted here that the use of this particular system presupposes that authors not only provide a good but also careful organisation of the instructive material, which should create and connect all the web pages of the electronic book that s/he wants to create with concrete criteria. Another authoring tool of adaptive courses is WINDS (Specht, 2001). The author can design adaptive courses through the use of relations, such as prerequisites, part_of, related_to. In a course WINDS implements direct guidance, link annotation, sorting and hiding, as well as additional explanations, various sequences of contents and even different graphs. For all these adaptive methods the author should write the appropriate rules; needless to say, a difficult task for non-programmers. Some other authoring tools, like REDEEM and VIDET aim to provide easy course authoring for teachers who are not programmers. REDEEM (Ainsworth et al., 2003) allows instructors to import pre-existent courses and provide tools to define the instructional strategies. VIDET (Armani, 2005) is a visual authoring tool which lets authors manipulate the hypertext structure, the content, the user model, and the adaptive interaction model.

Systems based on SCORM

Despite the attempts for reusable educational content (Stash et al., 2004), none of the above adaptive systems and authoring tools conforms to a widely accepted standard, resulting in their having limited reusability. Nevertheless, it must be stated that some systems are trying to incorporate the SCORM specification in order to obtain the RAID educational content. Some of these third generation systems are the following:

OPAL (Conlan et al., 2002) is an LMS which supports learning resources tagged by SCORM metadata. It performs adaptivity at the level of content packages by employing stretch-text type methods to selectively hide or remove individual content. However, it does not support SCORM compliant courses and its adaptive features are restricted. VIBORA (Morales, 2003) supports SCORM and lets students choose between three types of sequences in the course: pre-established sequence, manual selection of activities through the table of contents (TOC), and by following the "Socratic" method, where the student is evaluated and if necessary additional activities (exercises, examples and explanations) are provided. Even though VIBORA is SCORM compliant, its adaptivity is very restricted. AdeLE (Gütl & Mödritscher, 2005) is a project intended to support existing systems and enhance them with adaptivity. Its current implementation utilises ADL's SCORM RTE as the front-end of the system and is quite similar to our prototype. AdeLE builds its user model by tracking user interaction with the LMS and by tracking eye movements. Its adaptation process is highly dependent on SCORM specifications and is applied through an improved tree-view navigation. More specifically, in addition to SCORM RTE, AdeLE provides (i) automatically generated concepts, (ii) alternatives to the adapted decision about the current instruction, and (iii) information about the learner model. Not only the new but also the older systems are endeavouring to incorporate SCORM by extending their functionality. For example, a new version of AHA! enables authors to import (and soon to export) SCORM compliant courses (Romero et al., 2005). However, imported courses do not include all the facilities that AHA! offers, so further changes to navigation, content etc. using AHA! authoring tools is required.

As is apparent, much effort is expended on the development of adaptive systems and authoring tools that are compliant or that partially use an educational standard, such as SCORM, in order to provide quality personalized instruction and at the same time accommodate reusability of the educational material.

6. ProPer

ProPer (an acronym for the Greek translation of Adaptive Environment) (Kazanidis & Satratzemi, 2009a) is an LMS which is not only compliant to SCORM, but is also enhanced with adaptive and adaptable features. ProPer allows authors to upload SCORM compliant courses and provides adaptation to user knowledge, goals and progress. We have shown in (Kazanidis & Satratzemi, 2009b) that it is also possible to apply adaptation to user learning style as well as to other personal characteristics via SCORM specifications. ProPer tries to address the problem of reusability and durability that AEHSs encounter by adopting the SCORM standard. Thus, the educational content used by ProPer is reusable, easily accessible and interoperable. Furthermore by providing adaptive and adaptable technologies, ProPer provides dynamic and adaptive educational content in contrast to the majority of SCORM compliant LMSs. In addition, ProPer provides an Online Java Editor in order to help support students on Java programming.

6.1 Design and implementation

Therefore, its architecture adopts modules from categories of both systems. ProPer is comprised of four main modules: DM, UM, AM and RTE Sequencer. The DM provides the essential educational material in the form of concepts with specific properties and relationships between them. It contains the entire mandatory from the SCORM Content Aggregation Model and SCORM RTE Data Model data, for every single concept of the course. The concepts in the DM are implemented by SCOs. The DM is also surrounded by additional data, like the permission of manual knowledge declaration, FAQ and their answers. The UM keeps records of user characteristics, knowledge, goals and behavior so as to enable the system to provide the appropriate adaptation. Details about user modeling and UM architecture are presented later in the chapter. The AM is responsible for the system's adaptation. It interacts both with the UM and DM and provides all the system's adaptive features which are likewise presented in a later section. Finally, the RTE Sequencer is the system's sequencing engine. It is triggered by course navigation controls, interacts with the DM and AM and delivers the appropriate educational content to the learner. It sets up the user navigation process applying SCORM specification and rules.

For the implementation of ProPer, rather than use a brand new LMS, we decided to rely on ADL's SCORM Runtime Environment 1.3.3. Therefore, our main exertions focused on the development of adaptive and intelligent technologies and not on simple LMS functionalities. Nevertheless, the database was changed to MySQL and several extensions where added in order to enhance the AM and to enable the system's adaptive functionality.

Our concern was to design a simple and friendly interface (Figure1) based on SCORM RTE layout. On the top of the screen a functionality toolbar was added while the TOC was extended with appropriate adaptive annotations. A new frame was added at the bottom-left of the screen, which provides visualization of user progress, manual knowledge declaration and direct guidance controls. The main frame of the interface is assigned to the presentation of the educational content.

6.2 User Characteristics for adaptation

The user characteristics considered in ProPer for adaptation include: user knowledge, educational goals, user actions, learning style, language, which can be further extended using SCORM specifications.

6.2.1 Knowledge

The user's knowledge level is a crucial parameter for adaptive systems. ProPer stores the user's previous knowledge and estimates in real time his/her current knowledge. More specifically, the system initially assumes that the user has no previous knowledge of the domain. However, it lets the user declare whether s/he knows (or does not) the course concepts through the "User Model" screen. Once the user defines their previous knowledge, the system updates the UM and adapts its interface accordingly. In contrast to other systems, ProPer enables users to change their previous knowledge declarations even during the learning process. In addition, the system records user progress and counts user knowledge at three levels: a) in each activity; b) in the entire course; and c) in accordance with the user's educational goals.

Knowledge at the activity level is obtained by the SCO's mechanism and stored in the UM at the time the learner leaves the SCO. Course designers need to take special care to allow the UM to update, even when the user changes the web page into a multipage SCO. In actual fact the system, through a progress visualization mechanism (PVM), calculates and updates the user's total course score every time a new activity score is sent to the UM. Additionally, the PVM calculates a score based on user goals. It reconstructs the DM structure dropping out activities that do not constitute a part of the user's goals and changes the organization and activity weights accordingly. The outcome of this process is to annotate TOC in real time and display user's progress on two bar graphs that represent the percentages of course knowledge and the goals of the activities that the user has actually achieved.

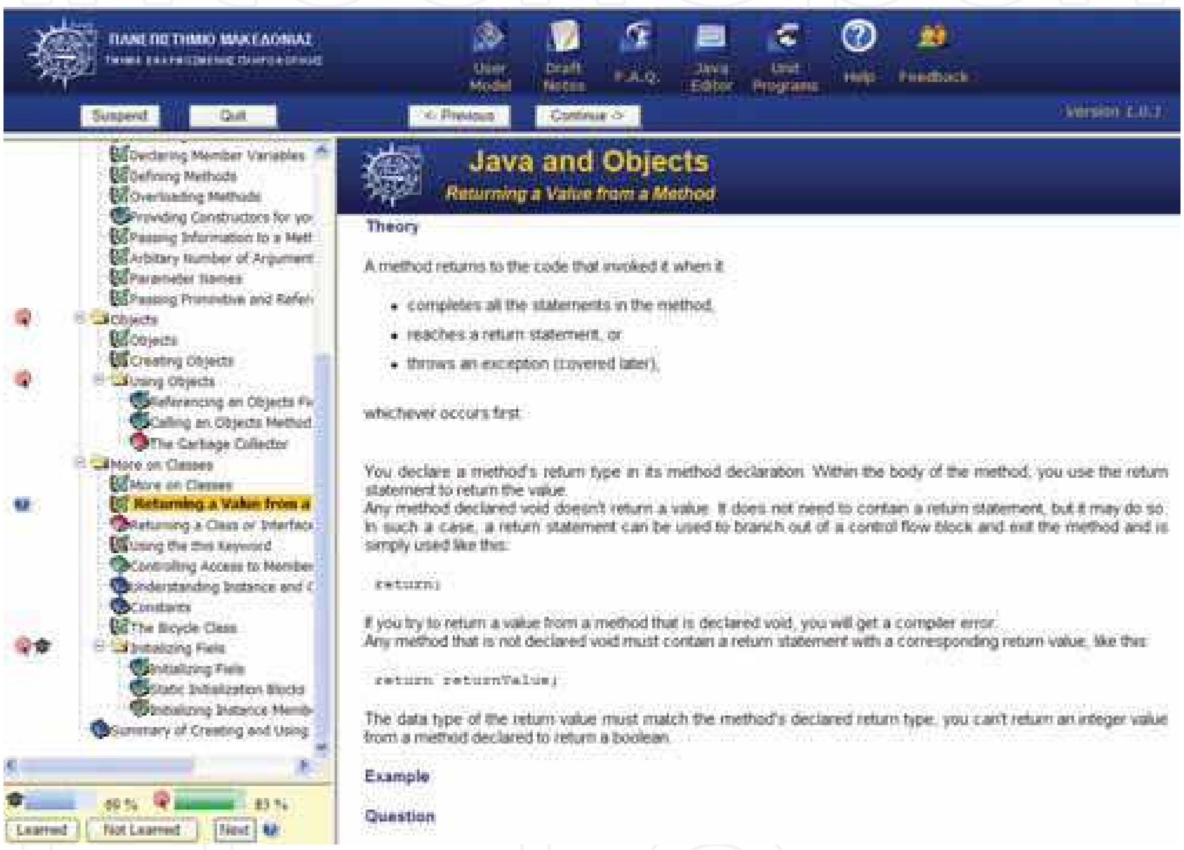


Fig. 1. ProPer's interface

The user knowledge in a SCO can be calculated by various methods, such as, by the visit(s) to a page, the study time spent on a page, how many subpages of a SCO the user has studied, the results of the assessment questionnaire etc. However, it must be mentioned that there is dispute as to the accuracy of user knowledge estimation by adaptive systems. It is true, we cannot assume that a visit to a page means that the user has learned the corresponding concept. For that reason, ProPer under specific conditions (i.e. with the instructor's permission) allows users to directly define whether they have or have not learned the concept of that activity. In this way we attempt to anticipate any possible faulty UM estimations

6.2.2 Educational Goals

As mentioned earlier, educational goals is a very important factor for adaptation. Users may have stable educational goals, e.g. learn the whole course content or the first two chapters, or might even change their goals during the learning process. Therefore, ProPer has been designed in a way that permits students to declare their educational goals at any time throughout the duration of the learning process. The user goals declaration, like the knowledge declaration, is carried out by the "User Model" screen. However, this time the system at the beginning assumes that every activity constitutes a user goal. There are situations where novice students may not know what exactly they want to learn and thus have problems with declaring their goals. Obviously n this case, it would be helpful to the student if they had a clearly described set of goals to select from. ProPer allows just this to happen by enabling instructors to predefine particular sets of goals that have been elucidated with a title and a short description as to which level they refer to, e.g. beginner, intermediate, advanced etc. In this way, different classes of users are initially categorized into stereotypes and then afterwards they are given the opportunity to appropriately modulate their goal selection (overlay model) in order to personalize them to their individual needs.

When the system knows what exactly the user wishes to learn, it can inform them in real time about the status of each activity (whether it is a goal or not), by annotating the TOC accordingly, as well as propose an optimal learning path which omits non-goal activities. Moreover, through the PVM system a score and a bar graph representing how much area has been covered of the user's educational goals are displayed.

6.2.3 User Actions

The user actions that are recorded in the system's UM are his/her navigational history and the study time for each course activity. In courses with many activities, it is essential for the user to have a clear idea which of them have already been visited. Thus, the system records whether or not the user has visited an activity, annotating the TOC accordingly. ProPer, however, goes one step further by recording the number of visits for each activity. Hence, it can provide statistical data to the user about the number of times a page has been visited, as well as, to apply structure adaptation. The statistics are available to the course instructor in order to be able to infer if an activity is difficult or confusing for students and where necessary, improve course design.

In addition, the system records the time a user spends on each activity. We have taken special precautions for the case where the user might have left the course window open without actually studying the content. If an activity page is visited for longer than the predefined time, then the counter is automatically set to zero. In this way, we hope to achieve better user modelling and subsequently more effective adaptation. In addition, the time a user spends on an activity can be used for content adaptation. Moreover, by way of the course statistics, the student can see the total time spent on an activity, while the instructor can view the average time students spent on any particular activities. From this instructors are able to make an inference about an activity, for instance, if less time is spent on a particular exercise then it is most probably easy or known to most users. In contrast, if students spend more time, this most likely means that the activity is difficult and/or confusing. These assumptions can bring about course improvement through structure and content modifications.

6.2.4 Learning Styles

As we have already seen there are many learning style theories and models that various AEHSs have adopted. ProPer is able to record the user's preferred learning style applying SCORM specifications. However, this method has both pros and cons. On the one hand, it enables instructors to design courses that adopt any learning style model they like with the appropriate content adaptation, but on the other hand, instructors may not actually have the necessary knowledge of learning style theories and SCORM specifications in order to develop such courses. Although we sought to address this problem by proposing a framework for easy learning style adaptation in SCORM compliant courses (Kazanidis & Satratzemi, 2009b), we believe that it would most certainly be better if an automated tool helped authors in the development of these types of courses.

For our summative evaluation we developed such a course that provides adaptation according to the Honey and Mumford learning style model. The first course activity retrieves the student's learning style either by declaration or through a specific questionnaire. The system then stores a corresponding value in a SCORM objective called "lstyle" (Kazanidis & Satratzemi, 2009b). Every time students visit an activity the SCO communicates with the LMS, reads the "lstyle" value and adapts its content presentation accordingly.

6.2.5 Other characteristics

As previously stated, instructors can use SCORM specifications to provide further adaptation to their courses. In this way, the system can also model the user learning style or any other user characteristic can be acquired through a questionnaire or user declaration. For instance, adaptation can be provided according to user language, knowledge of specific concepts (pre-requisites), occupation, age etc. Writers following the framework presented in (Kazanidis and Satratzemi, 2009b) have already designed courses that support learning style, language and pre-requisite adaptation.

6.3 Learner Modeling

This section presents the implemented UM architecture so as to be able to keep records of user characteristics, which are necessary to generate adaptation. ProPer's UM (Kazanidis & Satratzemi, 2009a) stores three categories of data: (i) knowledge of the domain, (ii) user actions and goals, and (iii) domain independent data, such as username, password, mail, language and privileges.

For knowledge representation a multilayered overlay model is used which consequently follows the DM structure. For every DM concept the UM maintains a number of four different layers. The first layer stores navigation history, in other words, data that shows whether the user actually visited the corresponding web page. The second layer contains a value which represents the percentage of expert knowledge that the user has already learned through study. The third layer stores the user's previous knowledge of the domain. Consequently, by this data being stored in different layers, means that every layer can be independently updated. Therefore, user knowledge data from one layer does not overwrite identical information from another.

The second category of data (number of visits, study time and whether an activity is a goal or not) is stored in an additional layer since they too are domain dependent. The user goal

model is a combination of overlay and stereotype models since the user can select one of the tutor's predefined sets of goals according to a category (stereotype model) or manually define his/her goals (overlay model). In this category can also be included, some domain dependent data that is not used for adaptation, but rather to inform both the user and the instructor. The user's draft notes and Java program details in regard with an activity constitute this particular data. Draft notes are kept by users to enable them to keep useful information about each activity. The information stored according to the Java programs is (i) the program code in every compilation, (ii) the total time the user spent writing this program, (iii) the number of compilations, (iv) the time the user spent between two subsequent compilations, and (v) the output and execution of each compilation. This data is available only to instructors in order to study users' behaviors and programming skills.
Finally, the user's personal data is stored separately as it is static and domain independent.

6.4 Adaptive System functionality

Every educational system and consequently the AEHS aims to make not only the learning process but also the learning outcome more effective. ProPer improves the latter through adaptive and adaptable features. A thorough presentation of the system's functionality has been made in (Kazanidis & Satratzemi, 2009a). Here, we will briefly present the adaptive and intelligent features in regards to the type of user they are designated for. As a typical LMS, ProPer supports three types of users: students, instructors and administrators. As far as administrators are concerned, the system provides some basic LMS functions. In the next section are presented the students' and instructors' functionality.

6.4.1 Adaptive and intelligent functionality for students

ProPer provides a variety of adaptive and intelligent features in order to personalize learning and help the user learn faster and easier. These features are presented below.

- **Adaptive link annotation**. ProPer annotates the TOC's links appropriately so as to inform the user whether the corresponding activities: have already been visited, constitute user goals, seem to be known or are suggested for study. The current activity is also annotated accordingly. Furthermore, the user can instantly see his/her estimated score in an activity or folder by placing the mouse over the icon s/he is interested in. Additionally, annotation is applied to the course structure folders, showing whether the activities therein are considered as known and whether the folder's goals have been accomplished. This functionality prevents users from coming up against disorientation, narrative flow and distraction problems during their study.
- **Link hiding**. Instructors can use conditions in order to apply link hiding, disabling and removal. This function supports students' navigation by reducing both cognitive overload and the number of navigation steps needed for a certain goal and in addition can address the distraction problem. Nevertheless, in agreement with de Bra and Calvi (1998) we too discourage the use of link removal.
- **Direct guidance**. There is a button that guides users to the next appropriate activity for study according to their previous and current knowledge, goals and navigation history. This function facilitates novice learners in particular to pursue the best learning path for their study.

- **Adaptive text presentation**. ProPer provides adaptive text presentation using SCORM specifications and JavaScript. The course author may use the framework presented in (Kazanidis & Satratzemi, 2009b) in order to provide adaptivity according to learning style, knowledge level or other factors. This technology reduces content readiness and narrative flow.
- **Adaptation of modality** and **Multimedia adaptation** can both be applied by ProPer following the same framework as the adaptive text presentation. This technique assists authors to provide students with the appropriate media according to learning style, preferences, disabilities or technological infrastructure.
- **Progress visualization**. ProPer displays in numerals as well as in bar graphs a visual image of the user score for the entire course which are associated with user goals. We strongly believe that this feature will motivate students to want to proceed with their learning.
- **Goal and previous knowledge orientation**. The system lets students, via the specific interface, to declare their previous knowledge and their educational goals in order to provide the appropriate adaptation.
- **Manual knowledge declaration**. In order to avoid faulty knowledge estimation by the UM, ProPer provides a mechanism to manually declare if an activity is considered known by students.
- Some other remarkable functions ProPer provides are the **Java Online Editor**, which can compile and execute Java programs recording user behavior during the development process, a **dynamic FAQ mechanism**, writing **draft notes** for every course activity and **analytical feedback** for users' progress.

6.4.2 Adaptive functionality for instructors

ProPer provides considerable functionality for instructors. It supports most of the usual LMS functionalities (course upload, course and user management etc.) as well as some features that promote adaptivity.

- **Permission for manual knowledge declaration**. As already mentioned, the system lets users declare if they consider an activity as known or not. However, sometimes (e.g. in assessment activities) instructors may want to prohibit users manually declaring their knowledge. For this reason ProPer asks authors to define the activities where the manual knowledge declaration is permitted.
- **Easy creation of group goals**. ProPer supports adaptation to user educational goals. However, sometimes users, especially novices, have problems defining their goals since they do not know exactly what they want to learn. For this reason, ProPer allows authors to define various sets of goals with a short description for whom the goal is designed in order to facilitate novice learners to choose one of the available groups.
- **Analytical statistics and feedback for both the users and the course.** Learning is a process that is liable to continual readjustment and improvement. Authors may export useful conclusions for the above statistics and feedback that will help them improve their courses.

6.5 Evaluation

The evaluation of ProPer is based on the procedures presented in (Samarakou, et al., 2006), which includes formative and summative evaluations. The overall aims of our evaluation were to examine the system's usefulness and how easy it is to use in accordance to the Technology Acceptance Model (TAM) (Davis, 1989), to identify any possible improvements in the learning outcome, as well as to detect the existence and the absence of the major problems often encountered in hypermedia courses.

For the formative evaluation the Tessmer Model (Tessmer, 1993) was adopted, which involves the following four phases: (i) Expert review, (ii) one-to-one evaluation, (iii) small group evaluation, and (iv) field trial. A revision of the system was carried out following the completion of each phase. Experts identify the system's navigation and the adaptive annotation as its strong points. However, it was suggested that better annotation symbol explanations and a mechanism for user feedback be provided. One-to-one and small group evaluations perceived some bugs in the system, which helped us to better organize the field trial that followed. The system was revised according to the proposals offered by the experts and the bugs were eliminated.

For the field trial 22 subjects were divided into two equal groups. Group A worked with ProPer and Group B with ADL SCORM Runtime Environment 1.3.3, which ProPer originated from. Both groups studied a course on Java and Object Oriented Programming. In brief, the results clearly showed that Group A participants not only proceeded faster but were also more goal oriented than those of Group B (P=.001). In addition, qualitative results indicate that all participants agree that the system greatly facilitated their learning and they the vast majority reported being pleased with using the system (90%). Following the formative evaluation the system and the interface were debugged and revised according to the evaluation results.

For the summative evaluation, we designed a new course that went beyond the formative evaluation course, which provides adaptation to the user learning style according to the Honey and Mumford (1992) learning style model. We also intentionally increased the number of subjects to 64, which were divided into four equal subgroups (ProPer vs SCORM RTE, adaptivity to learning styles vs no adaptivity). The evaluation process was similar to that of the formative field trial procedure. The results found that subjects who used ProPer gained significantly more knowledge (p=.046) than the others. However, there appears to be no significant difference among the four subgroups possibly due to the small number of subjects in each. Overall, the summative evaluation results confirmed the system's usefulness and usability that the formative evaluation had initially revealed.

In addition, an important finding in both the formative and summative evaluation results was that there was no evidence whatsoever of the major problems often found in hypermedia courses mentioned earlier.

7. ProPer SCORM Authoring Tool (ProPer SAT)

The problems that instructors without a technological background may have in course construction have already been mentioned. The matter becomes even more serious when adaptation is applied. Even when with the use of an educational standard, such as SCORM, issues of reusability, durability and interoperability can be addressed, instructors still have to spend a lot of valuable time in programming or constructing their courses, applying more or less complicated adaptation rules and strategies etc. ProPer was developed with simplicity in mind; we wanted it as simple as possible both for instructors and students. However, it does require the pre-existence of SCORM compliant courses that can be uploaded into the system. Instructors, thus, need to know not only SCORM specifications but also how to apply adaptation, which is not an easy job. For this reason, our aim is to implement a SCORM Authoring Tool, called ProPer SAT, which will enable authors with little or no prior knowledge of programming or SCORM specifications, to construct quick and easy adaptive and adaptable SCORM compliant courses.

So far, the design for ProPer SAT has been completed and we are now working on its development. The ProPer SAT design was based on three main axes: (i) to help instructors with little or no programming knowledge compose SCORM compliant courses; (ii) to provide course patterns for easy authoring of adaptive courses to user learning styles; and (iii) to enable easy reusability for both SCOs and content fragments. The prototype will support content writing, course structure construction and course packaging functionalities. Additionally, the system will enable instructors to easily create adaptive courses according to the Honey and Mumford model following the process proposed in (Kazanidis & Satratzemi, 2009b). However, in order to keep it as simple as absolutely possible, we had to leave functionalities, such as prerequisites and conditions out of the prototype provisions, since authors can use specialized free SCORM package editors, like Reload.

The ProPer SAT interface (Figure 2) keeps the main layout of ProPer. Thus, there is a functionality toolbar on the top of the screen, on the left is the course's TOC, and at the bottom of the screen is a place where further explanations and help about system functionalities are displayed. On the main frame of the screen, the user may define all the essential properties of the activity and write the appropriate educational content.

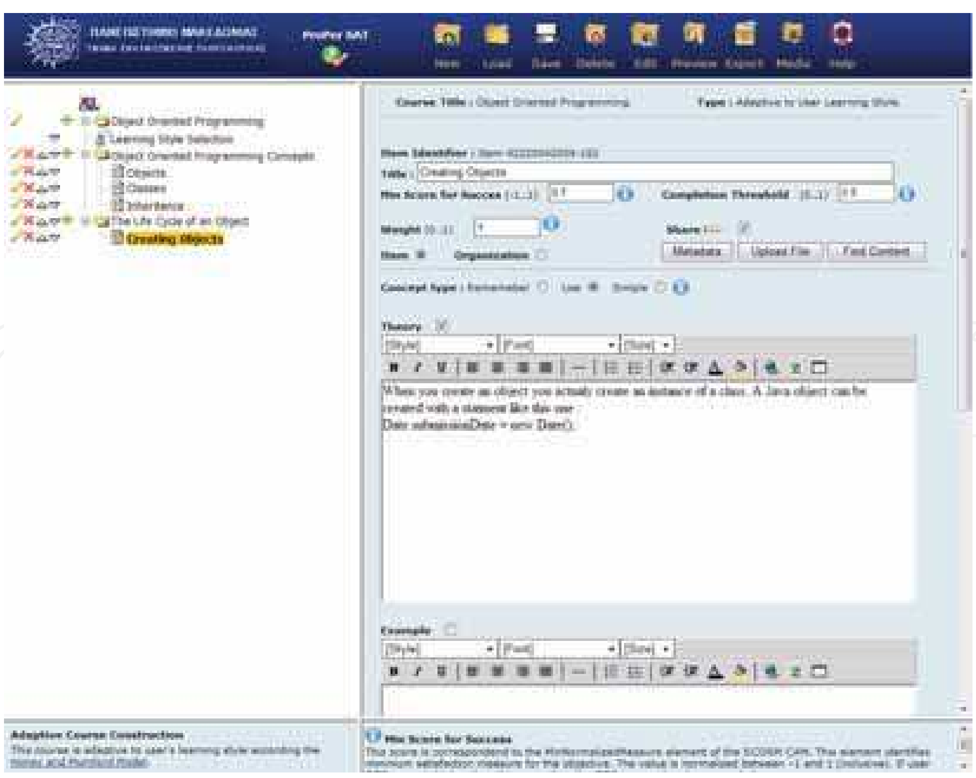


Fig. 2. ProPer SAT interface

For the implementation of both ProPer and ProPer SAT we use Apache Tomcat as a web and application server and MySQL 5 as a database server, while system's intelligece is written in Java Server Pages (JSP) and Java servlets. By adopting the same web technologies for both systems, they can function either complementarily or independently. Therefore, since ProPer SAT developed courses can be uploaded into every SCORM compliant LMS, this makes it a most useful tool for every instructor who wants to develop SCORM compliant and adaptive courses.

8. Conclusion

This work puts forward a proposal for distance education technology through the presentation of an integrated framework, based on three axes: adaptivity, educational standards and appropriate authoring tools. Our strong conviction is that knowledge over the Internet should be delivered through adaptive and interactive systems. Adaptivity will help students follow personalized instruction while interaction will make them active participants in the education process. This can address the major problems of web-based systems presented above. In addition, adaptivity to user learning style will certainly further improve the learning outcome. However, in order to develop a web-based course – even more so an adaptive course - a lot of effort is required. The adoption of a common educational standard, like SCORM, accepted by most of the web-based educational platforms allows the discovery, reusability, interoperability and durability of the educational content. It is of major significance if authors are able to reuse the material from quality courses and adapt it to their needs. This fact would enable distance educators to focus on the educational perspective of e-learning, instead of trying to solve design and system compatibility issues of the educational content. However, since authors without a firm background in technology may run into difficulties in constructing SCORM compliant and/or adaptive courses, the development of appropriate simple authoring tools is a crucial parameter of the e-learning process.

We have already implemented an adaptive to user preferences, knowledge and learning style SCORM compliant LMS (ProPer) and are in the process developing of a simple authoring tool (ProPer SAT) for such courses, which will also be appropriate for authors with little or no programming knowledge.

ProPer supports most of the adaptive technologies of Brusilovsky (2001) taxonomy. It personalizes instruction according to user knowledge, goals and navigational history, such as Interbook, KBS Hyperbook, Netcoach, INSPIRE, as well as the user learning style, like INSPIRE, AES-CS, CS383, iWeaver, EDUCE etc. It provides adaptive navigation as do the majority of AEHS as well as adaptive presentation of educational content, as do AHA!, INSPIRE, AES-CS and TANGOW. Similar to ELM-ART, AES-CS, INSPIRE ProPer uses a multilayered overlay model for user modelling. It can also include pre-tests and post-tests like ELM-ART and AES-CS and it allows students to explicitly define their knowledge on a course activity by using the appropriate buttons on the screen. This feature, which is not found in any other AEHS, we believe can improve the system's accuracy of user knowledge estimation. Furthermore, unlike other systems (e.g., Interbook, AES-CS) the system lets students change the UM during their study of the course. Besides being very competitive to similar AEHSs, ProPer's main strength over them lies in the combination of adaptivity with the adoption of the widely accepted learning standard SCORM. Thus, in contrast to the majority of AEHSs, it is able to offer RAID courses. It must also be noted that while some other systems, like OPAL, VIBORA, AdeLE support either SCORM metadata or even SCORM compliant courses, they do not present the range of adaptivity that our system does. Moreover, with both systems (ProPer and ProPer SAT) we provide an integrated solution for the adaptive educational process supporting instructors at the course authoring phase much like AHA!, Interbook and MetaLinks. ProPer SAT, stands out for its simplicity through identical authoring tools, as well as supporting the construction of both adaptive and SCORM compliant courses. To the best of our knowledge up until now with the creation of ProPer SAT, AHA! was the only authoring tool for adaptive and SCORM compliant systems. Like AHA!, ProPer SAT enables the development of such courses yet goes one step further and supports adaptivity to user learning style. Furthermore, in contrast to the other authoring tools, ProPer SAT provides patterns for easy and guided course authoring.

Our immediate aim is to finalise the development of ProPer SAT and evaluate both its usability and usefulness. Next, the courses that will be constructed with the use of this authoring tool will form supplementary educational material for traditional face-to-face classroom lessons, whose results will be checked for confirmation with the evaluation outcomes. Finally, following the completion of ProPer SAT's first development phase, we aim to enhance it with new features and patterns for better and more effective adaptive course construction.

9. References

ADL, (2009). SCORM 2004 4th edition Documentation Suite, [available on line] http://www.adlnet.gov

Ainsworth, S. E., Major, N., Grimshaw, S. K., Hayes, M., Underwood, J. D., Williams, B. & Wood, D. J. (2003). REDEEM: Simple Intelligent Tutoring Systems From Usable Tools, In: *Authoring Tools for Advanced Technology Learning Environment*, Murray, T., Blessing, S. & Ainsworth S., (Eds.), pp. 205—232, Kluwer Academic Publishers

Armani, J. (2005). VIDET: a Visual Authoring Tool for Adaptive Websites Tailored to Non-Programmer Teachers. *Educational Technology & Society*, Vol. 8, No. 3, pp. 36--52

Bajraktarevic, N., Hall, W., Fullick. P. (2003). Incorporating learning styles in hypermedia environment: Empirical evaluation. *In : Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2003)*, pp 41--52

Beck, J., Stern, M. & Haugsjaa E. (1996). Applications of AI in Education, *ACM Crossroads Student Magazine*.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-adapted Interaction*, Vol. 6, pp. 87–129

Brusilovsky, P. (2001), Adaptive hypermedia, *User Modeling and User Adapted Interaction, Ten Year Anniversary Issue,* Alfred Kobsa, (Ed.), Vol.11, No. 1/2, pp. 87-110

Brusilovsky, P. (2003). Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools, In: *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*, Murray, T., Blessing S. & Ainsworth S., (Eds.), pp. 377-409, Brusilovsky, P. (2004). Adaptive Educational Hypermedia: From generation to generation, *Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education*, pp 19-33, Athens, Greece

Brusilovsky, P. & Millan, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems, In: *The Adaptive Web. Methods and Strategies of Web Personalization*, Brusilovsky, P., Kobsa, A. & Nejdl, W. (Eds.), pp. 3-53, LNCS 4321 (2007), Springer-Verlag, Berlin Heidelberg

Brusilovsky P. & Pesin L. (1994). ISIS-Tutor: An adaptive hypertext learning environment. *Proceedings of JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering,* pp. 83-87, Pereslavl-Zalesski, May 10-13, 1994. Tokyo

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. *In Computer Networks and ISDN Systems. Proceedings of Seventh International World Wide Web Conference*, Vol. 30, pp. 291–300

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In: *Intelligent Tutoring Systems*, Frasson, C., Gauthier, G., & Lesgold, A. (Eds.), pp. 261-269,LNCS 1086 (1996). Springer Verlag Berlin

Carro, R.M., Pulido, E. & Rodríguez, P. (1999). Dynamic generation of adaptive Internet-based courses. *Journal of Network and Computer Applications*, Vol 22, pp. 249--257

Carver, C, Howard, R. & Lavelle, E. (1996). Enhancing student learning by incorporating learning styles into adaptive hypermedia. Proceedings of EDMEDIA 96, pp.118–123

Conklin, J. (1987). Hypertext: An Introduction and Survey. *IEEE Computer*, Vol. 20 No. 9, pp. 17-41

Conlan, O., Wade, V., Gargan, M., Hockemeyer, C., & Albert, D. (2002). An architecture forintegrating adaptive hypermedia services with open learning environments. In *Proceedings of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Barker P. & Rebelsky, S., (Eds.), pp. 344-350

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, Vol. 13, No.3, pp. 319-340.

De Bra, P & Calvi, L. (1998). AHA! an open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia*, Vol. 4, pp. 115–139

Dunn, R., Dunn, K. (1978). *Teaching students through their individual learning styles: A practical approach.* Reston Publishing, Reston, VA

Eklund J., & Brusilovsky, P. (1998). Individualising Interaction in Web-based Instructional Systems in Higher Education, *AUC Academic Conference 98*, University of Melbourne, Melbourne, Australia

Elsom-Cook,M. (1988). Guided Discovery Tutoring and Bounded User Modelling. In: *Artificial Intelligence and Human Learning - Intelligent Computer-aided Instruction.* Self, J., (Ed.), pp. 165-178, Chapman and Hall, London, New York

Felder, R.M. & Silverman, L.K., 1988. Learning and teaching styles in engineering education. *Engineering Education*, Vol.78 No. 7, pp. 674–681

Felder, R. M. & Soloman, B. A. (1996). Index of Learning Styles Questionnaire, [available on line] http://www.engr.ncsu.edu/learningstyles/ilsweb.html

Foss, C. L. (1989). Detecting lost users: Empirical Studies on browsing hypertext. *Technical Report 972*, INRIA, France, Fevrier 1989.

Gardner, H.: *Multiple Intelligences: The theory in practice.* New York: Basic Books. (1993)

Chen, S.Y., Fan, J.P., Macredie, R.D. (2006). Navigation in hypermedia learning systems: experts vs. novices. *Computers in Human Behavior*, Vol. 22 No.2, pp. 251–266

Goldstein, I. P. (1982). The Genetic Graph: A Representation for the Evolution of Procedural Knowledge. In: *Intelligent Tutoring Systems*, Sleeman,D.H., Brown,J.S., (Eds.),. Academic Press, London.

Grasha, A. F., & Riechmann, S. W. (1975). Student Learning Styles Questionnaire. University of Cincinatti, Faculty Resource Center, Cincinatti, OH

Gütl C., & Mödritscher F. (2005). Towards a Generic Adaptive System applicable for Web-based Learning Management Environments. *Proceedings of ABIS 2005*, Saarbrücken , Germany.

Henze, N. & Nejdl, W. (1997): A web-based learning environment: Applying constructivist teaching concepts in virtual learning environments. *In IFIP 3.3 and 3.6 Joint Working Conference The Virtual Campus: Trends for Higher Education and Training*, Madrid

Holt P., Dubs S., Jones M. and Greer J. (1994). The State of Student Modelling. In: *Student Modelling: the Key to Individualized Knowledge-based Instruction,* Greer, J.E., McCalla, G.I. (Eds.), pp. 3-38. NATO

Honey P., & Mumford A. (1992). *The manual of Learning Styles*, Peter Honey, Maidenhead, Berkshire

Honey, P. & Mumford, A. (2000). *The learning styles helper's guide,* Peter Honey Publications Ltd., Maidenhead

Kazanidis, I. & Satratzemi, M. (2008). Adaptivity in a SCORM compliant Adaptive Educational Hypermedia System. In: *ICWL 2007*, Leung, H., Li, F., Lau, R., Li, Q. (Eds.), pp. 196—206, LNCS 4823, (2008), Springer, Heidelberg

Kazanidis, I. & Satratzemi, M. (2009a). Adaptivity in Pro Per: an adaptive SCORM compliant LMS. J*ournal of Distance Education Technologies* Vol. 7, No.2, pp. 44--62

Kazanidis, I. & Satratzemi, M. (2009b). Applying learning styles to SCORM compliant courses. Proceedings of ICALT09, Riga, Latvia

Kavcic, A. (2000). The Role of User Models in Adaptive Hypermedia Systems. Proceedings of Tenth Mediterranean Electrotechnical Conference, Lemesos, Cyprus

Kavcic, A., Privosnik, M., Marolt, M., Divjak, S. (2002). Educational hypermedia system ALICE: an evaluation of adaptive features. *Advances in multimedia, video and signal processing systems, (Electrical and computer engineering series), WSEAS*, pp. 71-76

Kelly, D., Tangney, B. (2006). Adapting to intelligence profile in an adaptive educational system. *Interacting with Computers* , Vol. 18, pp. 385-409

Kolb, D. (1981). Experiential learning theory and the learning style inventory: A reply to Freedman and Stumpf. *Academy of Management Review*, Vol. 6, No. 2, pp. 289-296.

Kolb D. (1984). *Experiential Learning: experience as the source of learning and development*, Prentice-Hall, New Jersey

Martins, A. C., Faria, L., Vaz de Carvalho, C., & Carrapatoso, E. (2008). User Modeling in Adaptive Hypermedia Educational Systems. *Educational Technology & Society*, Vol. 11, No. 1, pp. 194-207

Morales, R. (2003). The VIBORA project. *Proceedings of World Conference on E-Learning in Corporate, Government, and Higher Education Healthcare,* Richards G., (Ed.), pp. 2341-2344

Murray, T., Shen, T., Piemonte, J., Condit, C., & Tibedau, J. (2000). Adaptivity for conceptual and narrative flow in hyperbooks: The Metalink system. In: *Adaptive Hypermedia and Adaptive Web-based system*. LNCS 1892, pp. 155-166.

Ng, M.H., Hall, W., Maier, P. & Armstrong, R. (2002). The Application and Evaluation of Adaptive Hypermedia Techniques in Web-based Medical Education. *Association for Learning Technology Journal*, Vol.10, No.3

Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H., & Magoulas, G.D. (2003). Personalizing the interaction in a Web-based educational hypermedia system: the case of INSPIRE. *User-Modeling and User-Adapted Interaction*, Vol. 13, No. 3, pp. 213–267.

Prentzas, D., & Hatziligeroudis, I. (2001). Adaptive Educational Hypermedia : Principles and Services. *Proceedings of First Panhellenic Conference in Open and Distance Learning*, Greece, [in Greek]

Riechmann, S. W., and Grasha, A. F. (1974). A Rational Approach to Developing and Assessing the Construct Validity of a Student Learning Style Scales Instrument. *Journal of Psychology*, Vol. 87, pp. 213-223

Romero, C., Ventura S., and Hervas, S. (2005). AHA! Meets SCORM. *Proceedings of IADIS International Conference Applied Computing 2005*. Algarve, Portugal

Rosson, M. B., & Carrol, J. M. (1990) Climbing the Smalltalk Mountain. *ACM SIGCHI Bulletin* Vol. 21, No. 3

Rundle, S. M., and Dunn, R. (2000). *The Guide to Individual Excellence: A Self Directed Guide to Learning and Performance Solutions*. Performance Concepts International, New York

Samarakou, M., Prentakis, P., Karolidis, D., & Papadakis, A. (2006). Formative and Summative Evaluation Procedures for an e-learning environment: the case of e-Study. *GESTS International Transactions on Computer Science and Engineering*, Vol. 27, No. 1, pp. 65-76.

Scheiter K. & Gerjets, P. (2007) Learner control in hypermedia environments, *Educational Psychology Review*, Vol. 19 , pp. 285–307.

Shearer, B. (1996). *The MIDAS Handbook of Multiple Intelligences in the Classroom. GreydenPress*, Ohio

Sidiropoulos D., & Bousiou-Makridou D. (2005). Perspectives for Improvement of Distance Learning Environments (SCORM). *E-mentor Magazine, Warsaw School of Economics, Center for Development of Distance and Permanent Education*, Vol. 5. No.12

Specht, M., Kravcik, M., Pesin, L., & Klemke, R. (2001). Authoring adaptive educational hypermedia in WINDS. In N. Henze (Ed.), *Proceedings of the ABIS 2001 workshop*.

Stash, N., Cristea, A. & De Bra, P. (2004). Authoring of learning styles in adaptive hypermedia: Problems and solutions. *In Proceedings of the WWW2004 Conference*, pp. 114--123

Tessmer, M. (1993). *Planning and Conducting Formative Evaluations: Improving the Quality of Education and Training*. London: Kogan Page.

Triantafillou, E., Pomportsis, A. and Georgiadou, E. (2002). AES-CS: Adaptive Educational System based on Cognitive Styles, *Proceedings of Workshop on Adaptive System for Web-based Education, held in conjunction with AH'2002*, Malaga, Spain

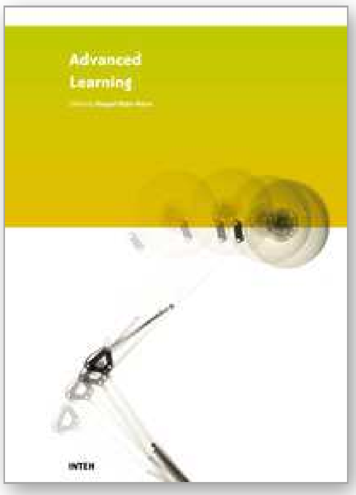
Weber, G., Hans-Christian, K., & Weibelzahl, S. (2001). Developing adaptive internet based courses with the authoring system NetCoach. *In Revised Papers from Int'l Workshops OHS-7, SC-3, and AH-3 on Hypermedia*, LNCS 2266, pp. 226-238, Springer

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan Kaufmann Publishers, Inc., Los Altos

Witkin, H. A., Moore, C.A., Goodenough, D.R., Cox, P.W. (1977). Field-dependent and field-independentcognitive styles and their implications. *Review of Educational Research* Vol. 47, pp. 1–64

Witkin, H. A., Ottman, P. K., Raskin, E., & Karp, S. A. (1971). *A manual for the embedded figures tests*. Palo Alto, CA: Consulting Psychologists.

Wolf C. (2002). iWeaver: Towards an Interactive Web-Based Adaptive Learning Environment to Address Individual Learning Styles. *Interactive Computer Aided Learning Workshop*, 25-27 September 2002



Advanced Learning

Edited by Raquel Hijn-Neira

ISBN 978-953-307-010-0
Hard cover, 444 pages
Publisher InTech
Published online 01, October, 2009
Published in print edition October, 2009

The education industry has obviously been influenced by the Internet revolution. Teaching and learning methods have changed significantly since the coming of the Web and it is very likely they will keep evolving many years to come thanks to it. A good example of this changing reality is the spectacular development of e-Learning. In a more particular way, the Web 2.0 has offered to the teaching industry a set of tools and practices that are modifying the learning systems and knowledge transmission methods. Teachers and students can use these tools in a variety of ways aimed to the general purpose of promoting collaborative work. The editor would like to thank the authors, who have committed so much effort to the publication of this work. She is sure that this volume will certainly be of great help for students, teachers and researchers. This was, at least, the main aim of the authors.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821