Journal of Universal Computer Science, vol. 21, no. 9 (2015), 1168-1209
submitted: 5/9/14, accepted: 18/8/15, appeared: 1/9/15 © *J.UCS*

Adaptive Courseware: A Literature Review

Ani Grubišić
(University of Split, Faculty of Science, Split, Croatia
ani.grubisic@pmfst.hr)

Slavomir Stankov
(University of Split, Faculty of Science, Split, Croatia
slavomir.stankov@pmfst.hr)

Branko Žitko
(University of Split, Faculty of Science, Split, Croatia
branko.zitko@pmfst.hr)

Abstract: This paper gives a detailed description of how adaptivity is realized in e-learning systems, with a focus on adaptivity of courseware. The review aims to answer three basic questions on adaptive courseware: what has been adapted, how and why. Researchers have tried to adapt according to these different elements: student knowledge, behaviour, interests, preferences, learning style, cognitive style, goals, courseware presentation, navigation, annotation, sequencing and knowledge testing. An historical background of adaptivity is described: from word definition, through adaptive instructions and adaptive systems, all the way to adaptive courseware. The biggest challenge was to find the most prominent representatives of adaptive systems in courseware generation that were selected after an exhaustive search through relevant scientific paper databases. Each prevailing system is briefly described and the student features, to which the system adapts, are highlighted, together with the level of adaptation. This paper aims to provide important information to researchers, educators and software developers of computer-based educational software ranging from e-learning systems in general to intelligent tutoring systems in particular. Such an analysis has not been done so far, especially in a way that adaptive systems are described uniformly. Finally, a comparative analysis of those systems and conclusions and demands about the ultimate adaptive system is given. This paper can be used as a guide for making decisions about what to adapt, how and why, while designing an adaptive e-learning system.

Keywords: E-learning, intelligent tutoring systems, adaptivity, courseware
Categories: L.2.0, L.3.0, L.3.6

1 Introduction

Throughout the history of computers much attention has been paid to the idea of using computers as intelligent collaborators (as personal teachers), which reveal and explain the complicated domain knowledge. The complexity of this task was observed once the final goal in the realization of these ideas was focussed on intelligent teaching. What has come to be known as *Artificial Intelligence* (AI), is the area that connects computers and intelligent behaviour, and this occurred at the end of the 1950s and early 1960s, with pioneers Turing, Minsky, McCarthy and Newell [Urban-Lurain (1996)]. AI is essentially oriented to knowledge representation, natural language understanding and problem solving, all areas equally important for the development of the concept of intelligent teaching [Beck, Stern, and Haugsjaa (1996)].

In the 1960s, scientists developed numerous systems for learning and teaching from computers such as *Computer Assisted Instruction* systems (CAI) [Uhr (1969)]. The term *Computer Based Training* (CBT) comprises all systems that enable a "dumb" way of learning and teaching. CAI systems usually expose a problem to students and remember the student's response. They are not overly concerned with issues of how students learn: if a student is given the information, it will be learned. The technical challenge in programming expensive and cumbersome "mainframe" computers was considered a bigger problem.

In the late 1960s and early 1970s, scientists moved from merely presenting tasks to students, to the recognition that students themselves are factors which affect the learning and teaching process [Suppes (1967)]. Some of the systems that were developed changed the presentation of educational content depending on the student's answers. The developers had to know in advance all possible student responses and decide what information to present. This marks the beginning of student modelling, although in this period only student behaviour was observed, and student knowledge was not modelled.

In 1982, Sleeman and Brown gave an overview of recent developments in the field of computer assisted instruction and adopted the term *Intelligent Tutoring Systems* (ITS) so as to distinguish them from the old CAI systems [Sleeman and Brown (1982)]. The learning in these new systems was based on the principle of "learning by doing". As one of the first researchers in this field, Carbonell combined CAI and AI when implementing the system Scholar, which is considered to be one of the first intelligent tutoring systems [Carbonell (1970)].

The new term that began in the late 1990s is *e-learning*. E-learning comprises the world of Information and Communication Technology (ICT) and the world of education [Stankov, Grubišić, and Žitko (2004)]. In comparison with traditional teaching in the classroom that has the teacher at the centre controlling the classroom, educational content and the learning and teaching process, e-learning puts students at the centre allowing them to learn interactively at their own pace, in a simple, flexible and distributed learning environment [Khan (2001)]. The most commonly used definition of e-learning is that it is a set of applications and processes, such as web-based learning, computer-based learning, virtual classrooms and digital collaboration, which allows access to educational content using a variety of electronic media [ASTD (2001)]. E-learning provides access to educational material anywhere, anytime and anyone [Albert (2001)]. *Intelligent e-learning systems* have the ability to behave appropriately in situations that arise in the learning and teaching process. A special class of intelligent e-learning systems is intelligent tutoring systems.

Intelligent tutoring systems are computer systems that are intended to support and improve the learning and teaching process in the selected knowledge domain, while respecting the individuality of students, as in the case with traditional human-based one-to-one tutoring ([Wenger (1987)], [Ohlsson (1987)], [Sleeman and Brown (1982)], [Self (1974)], [Shute and Psotka (1996)]). Intelligent tutoring systems fall into the category of *knowledge-based systems* because they encompass domain knowledge, pedagogical knowledge and knowledge about the student, which is the backbone of ITS.

More than forty years of development divides today's intelligent tutoring systems from the first attempts. Many systems have been developed, implemented and tested in the teaching process in schools and universities, and a unanimous agreement on the architecture of ITS has been reached. The traditional structure of ITS consists of four components [Burns and Capps (1988)]: domain knowledge, the tutor module, student module [Self (1974)] and a communication module [Woolf (1992)].

The *expert module* performs two important specialist functions. Firstly, it serves as a source of knowledge for students. Secondly, it is a standard for evaluating students' performance. The expert module is central to every intelligent tutoring system because it contains a domain knowledge base, and knowledge is the key to intelligent behaviour [Anderson (1988)].

The *student module* records the student's understanding or misunderstanding of the domain knowledge, that is, it captures the student's progress. Sleeman and Brown [Sleeman and Brown (1982)] first adopted the term student model to describe an abstract representation of a student. There are three categories of student models: (1) an overlay model, which records a student's knowledge as a subset of the expert knowledge, (2) a differential model, which records the differences between the student's knowledge and the knowledge of experts, and (3) a perturbation model, which records the student's misunderstanding of the expert knowledge.

The *tutor module* is closely associated with the student module and decides on how to teach each student individually. The tutor module must incorporate three characteristics: (1) a tutor must have control over the selection and sequence of the educational content, (2) a tutor should respond to student questions, and (3) a tutor must recognize when students need help and what help they need. These tasks that are handled by the tutor module are called learning scenarios [Rickel (1989)].

The communication module controls the interaction between the ITS and the student. The interaction can be realized through both dialogue and graphical user interfaces.

This paper presents a detailed description of adaptivity realization in e-learning systems, with the focus on adaptivity of courseware in intelligent tutoring systems. An historical background of adaptivity is described in the second section: from word definition, through adaptive instruction and adaptive systems, all the way to adaptive courseware. The third section describes the most prominent representatives of adaptive systems with courseware generation; these were selected after an exhaustive search through relevant scientific paper databases. Each system is briefly described and the student feature to which the system adapts is explicitly mentioned, as well as, the level of adaptation. Such an analysis has not been done so far, especially in a way that uniformly describes adaptive systems. Finally, a comparative analysis of those systems is given and conclusions and demands about the ultimate adaptive system have been drawn.

2 Background of adaptive courseware generation

The most important characteristic of tutoring is the ability that human tutors have to adapt the learning and teaching process to the students, taking into account their specific characteristics. First, an explanation of what adaptation is and what are the most commonly used derivatives of that word is presented; this is to ensure the precision of terminology used in this paper. Then there follows an explanation of what is adaptive instruction and its application in computer systems. Finally, the adaptivity realization in e-learning systems is explained, with special emphasis on the elements of courseware generation.

2.1 Adaptation

The word "adapt” is interpreted as: "to make an appropriate alteration to the current situation." Here are the definitions from the most acknowledged dictionaries:

- "Make fit for, or change to suit a new purpose" or "conform oneself to new or different conditions" - WordNet (wordnetweb.princeton.edu)
- "Make it fit (as for a specific or new use or situation) often by modification" or "to become adapted" - Merriam Webster (www.merriam-webster.com)
- "Make (something) for a suitable new use or purpose; modify" or "become adjusted to new conditions" - Oxford (oxforddictionaries.com)

The first meaning of this definition is changing something to meet some requirements or purpose. The second meaning involves the act of getting used to the changing environment.

The root word “adapt” can be combined with various suffixes to obtain words with different meanings and different uses. Here only the morphological variants of words that are important to this research and their meaning in the context of the subject are mentioned: adapt**ive** (showing or having a capacity for or tendency towards adaptation, capable of performing adaptation - the first meaning), adapt**able** (able to change or be changed in order to fit or work better in some situation or for some purpose – the second meaning), adapt**ation** (the process of changing to fit some purpose or situation, the process of adapting - the first and second meaning), adapt**ivity** (referring to the quality and capacity of adaptation), adapt**ability** (referring to the quality and capacity of adaptability), adapt**ive** engine (the component that recognizes, begins and executes adaptation – adaptation machine).

2.2 Teaching methods, techniques and strategies

A common error among teachers is to use interchangeably terms such as approach, method, strategy and technique. These terms are necessary to be defined first, so as to clarify the terminology used throughout the paper.

Teaching approaches
An approach is an open-minded view towards teaching [Garcia (1989)]. The teaching approach is a “particular way of thinking”; it is a set of beliefs and assumptions about teaching and learning. The teaching methods, teaching techniques, and teaching strategies will depend on the teaching approach taken.

Teaching strategies
According to Stones and Morris, “a teaching strategy is a generalized plan for a lesson which includes structure, desired learner behaviour in terms of goals of instruction and an outline of planned tactics necessary to implement the strategy” [Stones and Morris (1972)].

A teaching strategy is a teaching approach that is used either in solving a classroom problem or in improving instruction [McClosky (1971)]. A strategy defines the basic procedure of how the content is elaborated during the teaching process [Priewe and Dullien (2010)]. Strategy usually requires some sort of planning, so it is a plan of action designed to achieve an overall aim. It is designed to help learning take place.

Teaching methods

The teaching method includes a series of actions or steps taken by a teacher to achieve certain teaching and learning objectives. According to Stones and Morris, "a method of teaching is an organization and application of teaching technique, teaching materials, teaching aids and supplementary material by the teacher, with the aim of achieving the teaching and learning objective" [Stones and Morris (1972)].

Methods in educational practice refer to activities that teachers engage in during the learning and teaching process. Methods closely relate to the objectives teachers want their students to achieve [Priewe and Dullien (2010)]. A method is a description of the way that information or behaviour is carried forward or consolidated during the instructional process. In short, a method is a way something is done, it is a systematic plan.

Teaching techniques

A technique is a detailed list of rules or a guideline for any (teaching) activity [Priewe and Dullien (2010)], that is a procedure or skill for completing a specific task. Through techniques, teachers are enabled to develop, create and implement, using their distinctive way of working and the procedures (methods) they adopt in their teaching [Garcia (1989)].

2.3 Adaptive instruction

It is not always possible to realize tutoring in the learning and teaching process. Therefore, scientists have had to find other forms of instruction to achieve approximately equal efficiency using different methods and techniques of adaptation. Teaching approaches and techniques that are directed towards the needs of individual students are called adaptive instruction [Corno and Snow (1986)] based systems. Thus, adaptive instruction applies to educational interventions intended to adapt effectively to individual differences of students in order to help them develop the knowledge and skills needed to perform certain tasks. Adaptive instruction is generally characterized as an approach to education that includes alternative methods and strategies, and flexibility in learning [Wang and Lindvall (1984)].

Any form of one-to-one tutoring can be considered to be individualized. However, if the individualized tutoring is not sufficiently flexible to students' specific needs, it cannot be considered to be adaptive. Similarly, teaching in a group environment can be adaptive if it is sensitive to the unique needs of each student, as well as the common needs of the group. The ideal individualized instruction should be adaptive, because teaching is most effective when tailored to the unique needs of each student [Park and Lee (2008)].

Adapting the teaching to individual student's needs has a long history. Even in the fourth century BC, the adaptation was considered the main factor in the success of instruction [Corno and Snow (1986)], and adaptive tutoring classes had been a common method of education until the mid-19th century [Reiser (1987)]. During the 20th century, the importance of adapting the teaching of a specialized curriculum to different grades and different students was constantly stressed [Park and Lee (2008)]. They began to implement a number of studies to determine which student characteristics should be taken into account when adapting classes to the individual student and how to adapt teaching methods and techniques so as to complement these characteristics.

Student characteristics to which instruction must be adapted are called preferences or aptitudes and accessing those preferences is called aptitude-treatment interaction (ATI). The preference or aptitude of any individual is a feature that increases or decreases the likelihood of the student's success, and treatment is a variation in the teaching method ([Cronbach and Snow (1977)], [Snow and Swanson (1992)]).

The ATI application model for courseware generation consists of eight steps: (1) identify the goals, (2) specify the task characteristics, (3) identify an initial set of student characteristics, (4) select the most important student characteristics, (5) analyse the students in the target population, (6) select the desired difference (in student performance), (7) determine how to adapt the instruction, and (8) design alternative teaching methods [Carrier and Jonassen (1988)]. This model is actually a modified approach to instructional design ([Gagne and Briggs (1979)], [Dick, Carey, and Carey (2009)]).

Implementation of the ATI model requires answers to these questions [Shute, Lajoie, and Gluck (2000)]: which features should be measured prior to the learning and teaching process, which variables should be manipulated, how progress in the learning and teaching process should be measured, and which effectiveness measures to use? A taxonomy of learning skills, developed by [Kyllonen and Shute (1988)], can help in answering these questions. This taxonomy defines a four-dimensional space that includes domain knowledge, the environment for the learning and teaching process, the desired results from the learning and teaching process, and the student attributes.

As the use of computers has increased, software systems that have applied the above principles of adaptation have begun to be developed.

2.4 Adaptive systems

Adaptive Systems (AS) are those that can change their structure, functionality, or interface so as to accommodate the different needs of individuals or groups, as well as, change their needs over time [Benyon and Murray (1993)]. Adaptive systems can be adaptable (from adaptability) and adaptive (from adaptivity) [Oppermann and Rasher (1997)]. A system that allows students to change certain parameters, causing it to adapt its behaviour accordingly, is called an adaptable system. A system that automatically adapts to the student, based on its assumption about the student, is referred to as an adaptive system. This paper concentrates solely on (the latter) adaptive systems that enable automatic adaptation.

Adaptive systems and adaptive instruction define a new class of system - Adaptive E-learning Systems – that change the process of learning, teaching and testing students' knowledge, based on the student's individual characteristics. That is, they adapt the selection and presentation of educational content according to the student’s status, needs, style, prior knowledge and preferences [Brusilovsky and Nijhawan (2002)], [Santos et al. (2002)], [Shute and Towle (2003)], [Paramythis and Loidl-Reisinger (2003)]. Sometimes they are referred to as Adaptive Educational Systems (AES) [Brusilovsky (1998)].

In e-learning systems, adaptivity is realized according to a student's goals [Kaplan, Fenwick, and Chen (1993)], [Grunst (1993)], [Vassileva (1995)]), their foreknowledge ([Milosavljevic (1997)], [Hockemeyer, Held, and Albert (1997)], [Kayama and Okamoto (1998)]), their learning style ([Gilbert and Han (2002)], (Specht & Oppermann, 1998)), and their expertise ([Pérez et al. (1995)], [Vassileva (1995)]). There are many other characteristics to which e-learning systems adapt.

The two best-known representatives of adaptive e-learning systems are Intelligent Tutoring Systems (ITS) and Adaptive Educational Hypermedia Systems (AEHS). Student characteristics to which intelligent tutoring systems adapt are oriented towards knowledge (goals, knowledge, expertise, etc.), while AHSs adapt mainly to learning styles.

2.5 Adaptivity in e-learning systems

There are two main approaches to achieving adaptivity [Shute, Lajoie, and Gluck (2000)]: macro-adaptation and micro-adaptation. The macro-adaptation occurs before the learning and teaching process. First, data about the student’s cognitive abilities have to be collected, and then used to make decisions about the type of learning environment and instruction that will best suit those abilities. In contrast, the micro-adaptation occurs during the learning and teaching process. It includes changes in what is presented, not in how it is presented. These decisions are based on the student’s current knowledge compared with the knowledge that they should have when the whole process is complete. When students show a lack of knowledge, a new or previous content is re-presented. If students demonstrate knowledge, skipping certain parts is allowed.

In the traditional classroom, the relationship between teacher and student is one-to-many (1 - ∞). In a tutoring environment this relationship is one-to-one (1-1). Adaptive e-learning systems use the many-to-one relation (∞ -1) which provides each student with multiple teaching methods.

There are many different aspects of students to which e-learning systems adapt, such as knowledge, learning styles, cognitive styles, feelings, preferences, etc. [Lin (2007)]. The two most important approaches to adaptivity implementation in e-learning systems are: adaptation to learning styles and the adaptation to cognitive characteristics.

Generally, the learning styles theory is based on the fact that people have different approaches to learning (how to acquire and process that information) and learning and the teaching process will be effective if they are given appropriate instruction adapted to their learning style [Gilbert and Han (2002)]. The best known classification of learning styles is the Felder-Silverman model [Felder and Silverman (1988)] that refers to ways in which students acquire and process information: perceptual and intuitive, visual and auditory, inductive and deductive, reflective and active, generally and sequential. There is always the possibility that a student has multiple learning styles for the same domain knowledge, or that the student prefers different learning styles in different knowledge domains. Also, learning styles change during the learning and teaching process, a factor that most of systems which use learning styles as the basis for adaptation do not take into account.

Adaptive e-learning systems that implement learning styles in the learning and teaching process are called adaptive educational hypermedia systems. They adapt only the presentation (how) to the student's learning style, but do not use that information to decide what to display to the student ([Brusilovsky, Schwarz, and Weber (1996)], [Beaumont and Brusilovsky (1995)], [Brusilovsky (1996)]). Adaptive educational hypermedia systems use adaptive presentation and adaptive navigation support based on the auditory/visual learning style. Presentation adapts the content of a document or a text style (C-Book and Hypertext) [Brusilovsky (1998)]. Adaptive navigation supports students in orientation and navigation in hyperspace by changing the appearance of visible links (ELM-ART, InterBook, WEST-KBNS and AST) [Brusilovsky (1998)]. Basic methods for adapting the content are: additional explanations (only for those students who can understand them), prerequisite explanations (repetition of those concepts for students who do not have enough prior knowledge), comparative explanations (to explain the new concepts that are similar to already adopted ones), variants of explanations (the same information, but presented differently - text, graphics, sound), and sorting (same information arranged for each student separately). The basic methods for adapting hyperlinks are: global and local control (where the student is offered a path through all the lessons or the next lesson), support for global and local orientation (a special indication of those hyperlinks which point to learned concepts, or concepts that are recommended, or concepts that are relevant but are not being made available at this stage).

Another approach used in adaptive e-learning systems is adapting to students' cognitive abilities, as adopted in the Cognitive Trait Model (CTM) [Kinshuk, Lin, and Patel (2005)]. There are four cognitive abilities: working memory capacity, inductive reasoning ability, associative learning ability, and speed of information processing. These cognitive abilities are the characteristics of the student's overall cognitive capacity, and are called cognitive traits. Adaptation in this context ensures that the cognitive capacities of students are not over-loaded, so as not to discourage them. One such system that has implemented CTM is Marginal Costing Adaptive Learning Modules (MCALM) [Lin (2007)].

Regardless of adaptation approach, it should take account of the following [Riad, El-Minir, and El-Ghareeb (2009)]: every student should learn at their own pace, adaptation should happen often, each student must successfully complete the learning and teaching process, when something is learned successfully the student should continue, and no one should have to learn what they already know.

When considering the adaptation in e-learning systems, intelligent tutoring systems in particular, it is necessary to distinguish which system components must be adapted, and on what basis (when, how and why) [Mödritscher (2008)]:

1. Adaptation information determines what form the adaptation is to take. Typically, this is the student’s knowledge. However, the adaptation may be based on any state in the system’s environment. Therefore, adaptive systems possess at least one component that observes their environment.

2. Adaptation rules are necessary for making decisions as to when to begin the adaptation process. These rules are based on the adaptation information and are considered to be triggers for adaptation.
3. Adaptation procedures indicate which system components cause the adaptation and how.
4. Adaptation targets relate to the objectives of adaptation, but not to their resources. Therefore, components that model the adaptation objectives must also observe states from the environment and evaluate the effect of adaptation.

Often the term personalization is used as a synonym for the adaptation. However, personalized systems are just a special type of adaptive system, in which the student is an adaptation target [Mödritscher, Garcia-barrios, and Gütl (2004)].These authors define five dimensions of personalization:

1. Explicit - implicit: Explicit personalization describes the adaptation to the specific student model. Implicit personalization refers to adaptation in the environmental situation without using a student model, for example, when adapting the content’s layout appropriate for the device the student is using.
2. Visible - hidden: Personalization is visible if the student recognizes the results of the personalization. Hidden personalization is not visible to the student.
3. Predictable - determined: Predictable personalization includes pre-prepared adaptation steps. Determined personalization takes place within one adaptation step.
4. Controlled - uncontrolled: Controlled personalization allows students to take control of the adaptation process at any time. Uncontrolled personalization does not allow the student to influence the process of adaptation.
5. Individual - stereotypes: Individual personalization includes personalization to one student. Stereotype personalization refers to a group of students or anonymous students (so called group-based or role-based personalization).

An adaptation method is one that is defined at the conceptual level, while an adaptation technique is an implementation of an adaptation method. In determining how to adapt, it is necessary to take account of the following: WHERE (the application area is an adaptive system), WHY (what are the goals of adaptation), TO WHAT (that is, to which student characteristics is the system adapting, for example is it knowledge, learning style, or what), WHAT (is it content or navigation that is the focus of adaptation), and HOW (what adaptation methods and techniques are used) [Brusilovsky (1996)].

2.6 Elements of courseware generation

Intelligent tutoring systems were supposed to take the lead among adaptive e-learning systems. However, this did not happen because of their inflexibility and cost of development [Mohan, Greer, and McCalla (2003)]. In order to successfully realize a flexible, dynamic, personalized courseware in ITS, it is necessary to define an architecture that provides a clear distinction between the "adaptation machine" that dynamically generates the courseware, and the "content" that is used in generating that personalized courseware [Dagger, Wade, and Conlan (2004)].

Amongst the adaptation techniques used in intelligent tutoring systems are: content sequencing, intelligent analysis of students' solutions, interactive problem solving support, example-based support for problem solving, and collaboration support. The adaptation techniques used in AEHSs are adaptive presentation and adaptive navigation support [Brusilovsky (1998)].

Behaviours that should be measured in adaptive e-learning systems, especially in intelligent tutoring systems, are those which can be predicted from specific teaching techniques. Knowledge provides a more valid and reliable basis for determining adaptation results than any other preferences or abilities [Glaser and Nitko (1970)].

Since the main reason for the existence of e-learning in general is the growth of students' knowledge, it is necessary to focus research towards adapting courseware in intelligent tutoring systems, and study the content sequencing and content adaptation (WHAT), rather than focussing on adaptive presentation (HOW), which is generally typical of adaptive educational hypermedia systems.

Whilst elementary adaptation in the learning and teaching process began twenty years ago, a reference model for courseware generation has not yet been defined [Ullrich (2008)]. There are many names for courseware generation: course (ware) sequencing, curriculum sequencing, trail generation, course planning, instructional modelling, sequencing within learning objects (or courseware elements), and instructional planning. Also, concepts such as dynamic, adaptive, personalized and intelligent courseware generation are used in the sense defined by Chapelle and Mizuno [Chapelle and Mizuno (1989)] for generating courseware that is responsive to student needs.

Regardless of the terminology used, generating courseware includes content planning and content delivery planning [Dijkstra, Krammer, and Van Merrienboer (1992)]. The content planning involves generation, sequencing and selection of content items based on the student's current knowledge, and overseeing the execution of the content plan to determine when to pre-plan (adopt an already existing plan) or when to generate a new plan. Content delivery planning, also called a teaching strategy, refers to the selection of activities and interactions that will help the student to achieve their goals [Vassileva and Wasson (1996)]. Most common teaching strategies are: learning by example, learning by reading texts, learning by doing, and simulation.

Courseware consists of courseware elements for learning and for knowledge testing. Generating courseware is the process of selecting courseware elements and their sequencing in a manner that is appropriate for the targeted group of students or individual student ([Mohan, Greer, and McCalla (2003)], [Karampiperis and Sampson (2004)]), and is considered to be the most interesting research that is related to e-learning systems ([Brusilovsky and Vassileva (2003)], [Karampiperis and Sampson (2004)]).

Generating courseware is a well-established adaptation method used in intelligent tutoring systems. The idea is to generate individualized instructional content for each student, and to select dynamically the optimal teaching method at every step of the learning and teaching process. The optimal teaching method is the one that will lead students towards the realization of their learning goals. The most common learning goal is to acquire the necessary knowledge in the shortest possible time [Brusilovsky and Vassileva (2003)].

There are different approaches for generating courseware in intelligent tutoring systems. Most systems can only generate courseware that adopts just one teaching method. Some can only change the order of questions or problems, and some can arrange the order of lessons, but the most advanced systems can generate courseware using multiple teaching methods [Brusilovsky and Vassileva (2003)].

There are three different approaches for generating courseware: static courseware generation, adaptive courseware generation, and dynamic courseware generation [Brusilovsky and Vassileva (2003)]. Traditional intelligent tutoring systems adapt the learning and teaching process using appropriate pedagogical strategies during the presentation of domain knowledge [Oppermann and Rasher (1997)]. The courseware model in most intelligent tutoring systems is a static array or tree. All elements are created in advance by the teacher. The entire courseware is generated at once, which reduces flexibility.

The idea of adaptive courseware generation is the creation of customized content before students start using it. Dynamic courseware generation observes student progress. If the student performance does not match what is expected, the content is dynamically re-generated. During the re-generation, the courseware model is static. This approach takes into account the student's current knowledge, goals, and time frame; it changes its level of difficulty and adapts to the student's progress [Brusilovsky and Vassileva (2003)].

In all systems that generate courseware, there are courseware elements known also as: domain knowledge elements, concepts, fragments of knowledge, themes, learning objects, knowledge objects [Brusilovsky and Vassileva (2003)]. The idea of using small, re-usable chunks is also present in instructional design [Wiley (2000)]. These chunks of knowledge may represent a larger or smaller part of the knowledge domain, depending on the knowledge and those who define them.

There are several techniques for implementing courseware adaptation [Bhaskar et al. (2010)]:

1. A learning path graph is a directed acyclic graph that defines all possible learning paths that match a given learning objective [Karampiperis and Sampson (2005)].
2. A concept path graph is a directed acyclic graph that represents a set of sequencing rules which determine the sequence of concepts ([Pukkhem, Evens, and Vatanawood (2006)], [Carchiolo et al. (2003)]).
3. A concept map is used for the graphical presentation of domain knowledge-based courseware [Chang et al. (2008)].
4. An ontology represents a set of abstract concepts and semantic relationships between them [Gascuena, Fernandez-Caballero, and Gonzalez (2006)].
5. A learning activity graph is a directed graph that is used to organize content within a learning task [Zhu and Cao (2008)].
6. A Bayesian network is a directed graph whose nodes are uncertain variables and edges are causal relationships between variables. Each node has an associated table of conditional probabilities that depend on the student's characteristics ([Márquez et al. (2008)], [Anh, Ha, and Dam (2008)]).

The most complex of these techniques is the Bayesian network that predicts the probability that the student will learn the new knowledge based on their characteristics that were established prior to their involvement in the learning and teaching process. The probability is modified according to the student's performance, by selecting and adapting the teaching method. It was shown that the hardest part in applying the Bayesian model is the determination of "a priori" probabilities based on the results of an initial test [Park and Lee (2008)].

Shute and Zapata-Rivera proposed a cyclic four-phase process of adapting courseware in e-learning systems: (a) collecting information about the student, (b) creating and maintaining a student model, (c) selecting courseware elements based on the student model, and (d) presenting courseware elements based on the student model [Shute and Zapata-Rivera (2008)].

3 Adaptive systems with courseware generation – an historical overview

There now follows a description of knowledge-based systems that generate adaptive courseware. In these systems, the student's domain knowledge develops during the learning and teaching process.

There are many classifications of cognitive styles and learning styles, as well as contradictory evidence about which approach is best. It is clear that students show a preference towards one way of learning and teaching, but it is not clear how stable is this preference (does it change according to content, does it change according to student mood) and how reliable is it for determining the way in which the student prefers to be taught [Magoulas, Papanikolaou, and Grigoriadou (2003)]. Brusilovsky, although a great supporter and originator of the idea of adaptive hypermedia, states that the information about the student, obtained by checking the student's knowledge, is more reliable than drawing conclusions on the basis of the student's navigation sequence [Brusilovsky (1996)].

Bloom's knowledge taxonomy [Bloom (1956)] provides a clear and stable base for developing adaptive e-learning systems.

The focus of this review paper was to find in the literature those systems that automatically and dynamically generate adaptive courseware, with an emphasis on the adaptation of content, not the interface. Automatic courseware generation means that courseware elements for learning and testing knowledge are created by the system itself. Dynamic courseware generation means that courseware is created at the moment of execution. Thus, the keyphrases that were used as the basis for the search of sources in relevant scientific paper databases (CurrentContents, WebOfScience, CiteSeerX, INSPEC, ScienceDirect, SCOPUS) are: adaptive e-learning systems, intelligent tutoring systems, courseware generation, courseware sequencing, automatic courseware, dynamic courseware, adaptive courseware, and automatic generation of courseware. These keyphrases were also used in a combination with the word "course".

The search results themselves were interesting (many papers are repeated in several searches) [Tab. 1]:

- in total, there were 5924 papers that satisfied at least one search query in at least one database
- 21% (or 1260) papers related to adaptive e-learning systems
- 67% (or 3975) papers related to intelligent tutoring systems
- 12% (or 689) papers related to adaptive course(ware)
- an average value for the search relating to adaptive e-learning systems is 51, and all the search queries that resulted in a higher number were rejected

- an average value for the search relating to intelligent tutoring systems is 111, and all the search queries that resulted in a higher number were rejected
- an average value for search relating to the adaptive course(ware) is 115, and all the search queries that resulted in a higher number were rejected
- finally, 472 papers relating to adaptive e-learning systems, 597 papers for intelligent tutoring systems and 166 papers for the adaptive course(ware) remained

Keywords	**CC**	**WoS**	**CiteSeerX**	**INSPEC**	**Science Direct**	**SCOPUS**	**TOT**	**AV**
adaptive e-learning systems + courseware (course) generation	27 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=generati*	42 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=generati*	43 "adaptive e-learning" AND (courseware OR course) AND generation	68 adaptiv* and learning and (courseware or course) and generati*	23 "adaptive e-learning" AND (courseware OR course) AND generation	125 "adaptive e-learning" AND (courseware OR course) AND generation	328	55
adaptive e-learning systems + courseware (course) sequencing	20 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=sequenc*	30 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=sequenc*	27 "adaptive e-learning" AND (courseware OR course) AND sequencing	67 adaptiv* and learning and (courseware or course) and sequenc*	20 "adaptive e-learning" AND (courseware OR course) AND sequencing	64 "adaptive e-learning" AND (courseware OR course) AND sequencing	228	38
adaptive e-learning systems + automatic + courseware (course)	28 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=automat*	39 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=automat*	28 "adaptive e-learning" AND (courseware OR course) AND automatic	100 adaptiv* and learning and (courseware or course) and automat*	27 "adaptive e-learning" AND (courseware OR course) AND automatic	72 "adaptive e-learning" AND (courseware OR course) AND automatic	294	49
adaptive e-learning systems + dynamic + courseware (course)	49 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=dynam*	87 TS=(adaptiv*) AND TS=learning AND (TS=courseware OR TS=course) AND TS=dynam*	37 "adaptive e-learning" AND (courseware OR course) AND dynamic	127 adaptiv* and learning and (courseware or course) and dynam*	32 "adaptive e-learning" AND (courseware OR course) AND dynamic	112 "adaptive e-learning" AND (courseware OR course) AND dynamic	444	74
e-learning SUBTOT	**124**	**198**	**101**	**362**	**102**	**373**	**1260**	**210**
e-learning AVG	**31**	**50**	**25**	**91**	**25**	**93**		**53**
intelligent tutoring systems + courseware (course) generation	10 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=generati*	13 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=generati*	169 "intelligent tutoring" AND (courseware OR course) AND generation	55 intelligent and tutoring and (courseware or course) and generat*	395 "intelligent tutoring" AND (courseware OR course) AND generation	460 "intelligent tutoring" AND (courseware OR course) AND generation	1102	184

intelligent tutoring systems + courseware (course) sequencing	6 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=sequenc*	12 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=sequenc*	114 "intelligent tutoring" AND (courseware OR course) AND sequencing	43 intelligent and tutoring and (courseware or course) and sequenc*	122 "intelligent tutoring" AND (courseware OR course) AND sequencing	162 "intelligent tutoring" AND (courseware OR course) AND sequencing	459	77
intelligent tutoring systems + automatic + courseware (course)	11 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=automat*	14 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=automat*	116 "intelligent tutoring" AND (courseware OR course) AND automatic	93 intelligent and tutoring and (courseware or course) and automat*	311 "intelligent tutoring" AND (courseware OR course) AND automatic	322 "intelligent tutoring" AND (courseware OR course) AND automatic	867	145
intelligent tutoring systems + dynamic + courseware (course)	10 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=dynam*	17 TS=(intelligent) AND TS=tutoring AND (TS=courseware OR TS=course) AND TS=dynam*	123 "intelligent tutoring" AND (courseware OR course) AND dynamic	79 intelligent and tutoring and (courseware or course) and dynam*	389 "intelligent tutoring" AND (courseware OR course) AND dynamic	431 "intelligent tutoring" AND (courseware OR course) AND dynamic	1049	175
intelligent tutoring systems + automatic generation	7 TS="automat* generat*" AND TS="intelligent tutoring"	12 TS="automat* generat*" AND TS="intelligent tutoring"	235 "intelligent tutoring" AND "automatic generation"	68 automat* generat* AND intelligent tutoring	56 "intelligent tutoring" AND "automatic generation"	129 "intelligent tutoring" AND "automatic generation"	507	85
intelligent tutoring systems + dynamic generation	3 TS="dynam* generat*" AND TS="intelligent tutoring"	6 TS="dynam* generat*" AND TS="intelligent tutoring"	112 "intelligent tutoring" AND "dynamic generation"	23 dynam* generat* AND intelligent tutoring	14 "intelligent tutoring" AND "dynamic generation"	45 "intelligent tutoring" AND "dynamic generation"	203	34
ITS SUBTOT	**47**	**74**	**657**	**361**	**1287**	**1549**	**3975**	**663**
ITS AVG	**8**	**12**	**110**	**60**	**215**	**258**		**111**
adaptive courseware (course)	8 TS="adaptive courseware" OR TS="adaptive course"	18 TS="adaptive courseware" OR TS="adaptive course"	17 "adaptive courseware" OR "adaptive course"	86 "adaptive courseware" OR "adaptive course"	37 "adaptive courseware" OR "adaptive course"	523 "adaptive courseware" OR "adaptive course"	689	115
TOTAL	**179**	**290**	**775**	**809**	**1426**	**2445**	**5924**	**987**

Table 1: Number of papers in relevant scientific databases

An analysis of the remaining papers (from 1235 papers, many are repeated in the different searches) showed that most of these sources only mention the concept of intelligent tutoring systems and the generation and sequencing of courseware, and generally discuss adaptive educational hypermedia systems. Adaptive educational hypermedia systems were eliminated from the results, focussing solely on systems that have implemented knowledge based adaptation. Recently, scientists have focussed their research solely on adaptive educational hypermedia systems and, especially, on learning style adaptation (ADAM [Wang, Wang, and Lin (2010)], Lecomps5 framework [Limongelli et al. (2011)], EDUCA [Cabada, Barrón Estrada, and Reyes García (2011)], Protus 2.0 [Vesin et al. (2012)]). That is the reason why there are no recent papers included in the literature review. The following table [Tab. 2] presents the most commonly indexed adaptive e-learning systems. This shows the number of papers in the database that have references to the system.

	CC	**WoS**	**CiteSeerX**	**INSPEC**	**Science Direct**	**SCOPUS**	**Total**
GTE **[Van Marcke (1990)]**	3	5	8	16	4	4	**40**
ECSAI **[Gavignet (1991)]**	0	0	1	0	2	0	**3**
DCG **[Vassileva (1992)]**	1	1	17	1	2	0	**22**
ELM-ART **[Brusilovsky, Schwarz, and Weber (1996)]**	0	0	172	0	44	0	**216**
AST **[Specht et al. (1997)]**	0	0	6	1	0	0	**7**
ACE **[Specht and Oppermann (1998)]**	0	0	4	0	8	1	**13**
KBS **[Henze and Nejdl (1999)]**	0	0	47	0	12	0	**59**
ATLAS **[Macías and Castells (2001)]**	0	0	0	0	0	0	**0**
INSPIRE **[Grigoriadou et al. (2001)]**	0	1	11	0	7	0	**19**
WINDS **[Specht et al. (2001)]**	0	0	2	0	0	0	**2**
NETCOACH **[Weber, Kuhl, and Weibelzahl (2002)]**	0	0	37	0	8	2	**47**
APeLS **[Conlan et al. (2002)]**	2	4	45	1	3	3	**58**
ACCT **[Dagger, Wade, and Conlan (2004)]**	1	1	7	0	0	1	**10**
ASM **[Karampiperis and Sampson (2005)]**	0	0	1	0	0	0	**1**
iCLASS **[O Keeffe et al. (2006)]**	0	0	1	0	0	7	**8**
PAIGOS **[Ullrich (2008)]**	1	1	5	0	2	2	**11**
ADOPTA **[Bontchev et al. (2009)]**	0	0	1	0	0	11	**12**
TOTAL	**8**	**13**	**365**	**19**	**91**	**32**	**528**

Table 2: The most significant adaptive e-learning systems – relevant database citations

Therefore, only systems that are mentioned in most scientific databases, and are considered the most important in this area by the largest number of authors, were chosen. These were analysed in terms of how each system generates courseware elements, how they are sequenced, and how the knowledge is presented to the student. Included in the analysis is the type of domain knowledge, the student model, the approach to courseware generation, the adaptivity and how the student knowledge is tested.

Some of the recent adaptive e-learning systems are included regardless of their number of citations [Tab. 3].

System	Paper
ELP1+ELP2	**[Essalmi et al. (2010)]**
ISCARE	**[Muñoz-Merino et al. (2012)]**
UZWEBMAT	**[Özyurt, Özyurt, and Baki (2013)]**
Oscar CITS	**[Latham, Crockett, and McLean (2014)]**
TECH8	**[Dolenc and Aberšek (2015)]**

Table 3: Recent adaptive e-learning systems

Each system is briefly described and the student characteristics to which the system adapts is explicitly mentioned, as well as, the level of adaptation. Such an analysis has not been done so far. In this paper adaptive systems are described uniformly as follows: (1) a brief description of the system's purpose and its authors; (2) the domain knowledge formalization; (3) the student model; (4) courseware generation; (5) adaptivity; (6) the student characteristic to which the system adapts; (7) level of adaptation; and (8) knowledge testing. The biggest challenge for the authors of this review was to derive descriptions of features, as many of the original papers do not explicitly describe them.

4 Comparative analysis of adaptive systems with courseware generation

In this section, the main differences in the approaches to adaptive courseware generation are identified. The following major elements will be compared: domain knowledge, the student model, courseware generation, the features that adapt, the level of adaptation and the knowledge testing strategy.

The biggest challenge in conducting this analysis was to unify the terminology, as different authors use different terminology for the same purpose. In order to describe each system uniformly, a common terminology is defined thus:

- Graph or network nodes are also called units, concepts, pages, topics
- A concept is also called a knowledge item
- Educational contents include introduction, explanation, example, exercise, test, text, demonstration, interactive environment, theory presentation, activity, definition, description, task, question, learning objects
- Learning path (as a sequence of educational contents) is also called courseware plan, content plan, concept path

- Courseware are also called pages of storyboard, learning path, learning scenario
- Courseware elements are also called tasks, units, lessons, sections, subsections, concepts, learning objects, modules, concept narratives (descriptions), pages
- Learning object are observed as paragraphs, exercises, objective test items
- Teaching strategies include learning by example, learning by reading texts, learning by doing

4.1 Historical overview of courseware generation systems

Here, a brief description of the selected courseware generation systems along with their authors and corresponding papers that describe them the most, are given. This historical dimension shows a time span of over twenty years.

The idea of adaptive courseware started with a Generic Tutoring Environment (GTE), an environment for development of intelligent courseware based on generic knowledge that includes tasks, methods and objects ([Van Marcke (1990)], [Van Marcke (1992)], [Van Marcke (1998)]).

A year later, an Environnement de Conception de Systeme d'Apprentissage (ECSAI), a learning environment generator with the intelligent sequencing of instructional units, appeared ([Gavignet (1991)], [Grandbastien and Gavignet (1994)], [Grandbastien (1999)]).

Although not the first one in this chronological sequence, the Dynamic Courseware Generator (DCG) is considered to be the most distinct one. The DCG distinguishes the domain knowledge concepts, represented by a graph (content planning), and the educational contents (presentation planning) ([Vassileva (1992)], [Vassileva (1995)], [Vassileva (1998)]).

A system that has the largest number of citations is the Episodic Student Modelling Adaptive Remote Tutor (ELM-ART), an intelligent web environment for learning LISP [Brusilovsky, Schwarz, and Weber (1996)]. It was founded on the basis of ELM-PE (ELM programming environment) that supported programming based on examples, intelligent analysis of problem solving, advanced testing and debugging [Weber and Moellenberg (1995)]. Both systems are based on the episodic modelling of students. The conclusions about the student's knowledge were based on which pages the student might have seen or read.

Two years later, an Adaptive Statistics Tutor (AST), a web-oriented courseware for statistics was developed [Specht et al. (1997)].

Two years after the ELM-ART and six years after the DCG were developed, there emerged an Adaptive Courseware Environment (ACE) [Specht and Oppermann (1998)] that combined the adaptive navigation support from ELM-ART with the presentation planning from DCG.

The last system from the 20th century was the KBS Hyperbook, an adaptive hyper book for an Introduction to Computing course [Henze and Nejdl (1999)].

The first decade of the 21st century was very productive, especially the first two years. In 2001, an Authoring Tool for Adaptive Software Design (ATLAS), a graphical tool for designing interactive dynamically adaptive courseware was developed [Macías and Castells (2001)], [Macías and Castells (2002)]). The ATLAS enabled an understanding of the relationships between structure, content, courseware presentation and the student model. The teacher creates, modifies and assigns tasks and content. While creating a new courseware, the teacher must make a conceptual decomposition of the tasks, and establish rule-based links between tasks.

Also in 2001, an adaptive hypermedia system called Intelligent System for Personalized Instruction in a Remote Environment (INSPIRE) appeared [Grigoriadou et al. (2001)], [Papanikolaou et al. (2003)]).

Another system that has origins in ELM-ART is the Web-based Intelligent Design and Tutoring System (WINDS) [Specht et al. (2001)], an authoring tool for content management which is closely associated with ACE. It ensures the creation of individualized courseware.

As with ACE, the NetCoach is an authoring system for creating Web-oriented adaptive courseware [Weber, Kuhl, and Weibelzahl (2002)] and was also based on ELM-ART.

The first system that presents courseware using candidate groups and narratives (descriptions) was the Adaptive Personalized e-Learning Service (APeLS) [Conlan et al. (2002)]. This approach has a content model, a student model and a narrative model.

Based on ideas from AEpLS, the Adaptive Course Construction Toolkit (ACCT), provides a set of tools for creating courseware whether adaptive or not [Dagger, Wade, and Conlan (2004)], [Dagger, Wade, and Conlan (2005)]). It contains tools for generating domain knowledge ontologies, descriptions, tests and repositories of educational contents. The ACCT exports courseware to the APeLS together with the domain knowledge and descriptions.

Karampiperis and Sampson have suggested a methodology for adaptive sequencing (Adaptive Sequencing Methodology - ASM), which uses statistical methods for finding the best learning path [Karampiperis and Sampson (2005)].

After working on the systems APeLS and ACCT, Conlan and Wade continued their research within the framework of Intelligent Distributed Cognitive-based Open Learning System for Schools (iClass) [O Keeffe et al. (2006)]. The iClass generates courses that adapt intelligently according to the student's cognitive features. Unlike APeLS, iClass separates pedagogical information from the domain knowledge structure.

A new approach to courseware generation was created in PAIGOS, a model for generating courseware that has great pedagogical knowledge, and is independent of learning theory [Ullrich (2008)]. The courseware generation in PAIGOS is based on hierarchical task network planning (HTN planning), which was first implemented in the environment for e-learning Formación Humana (FORHUM) [Mendez, Ramirez, and Luna (2005)]. ActiveMath is the intelligent e-learning courseware generator that uses PAIGOS [Melis et al. (2009)].

One of the most recent approaches is used in the ADaptive technOlogy-enhanced Platform for eduTAinment (ADOPTA), a framework for generating adaptive courseware in e-learning systems [Bontchev et al. (2009)]. Courseware in this environment is called a storyboard.

A full personalization strategy of e-learning scenarios is implemented in both E-Learning personalization levels 1 and 2 (ELP1+ELP2) [Essalmi et al. (2010)]. The main objective here is to allow teachers to choose and apply the personalization strategy which matches the students' characteristics and the specifics of the courses.

The first personalization level ELP1 allows the learning contents and structure of the course to be personalized according to a given (specified within ELP2) personalization strategy (applied to the selected learning scenario). The second personalization level, ELP2, allows the personalization strategy to be defined by choosing a subset of personalization parameters.

An Information System for Competition based on pRoblem solving in Education (ISCARE) [Muñoz-Merino et al. (2012)] is an intelligent tutoring system, based on the Swiss-system tournament, that allows students to compete to improve their learning process. The competition is based on different tournaments and rounds. In each round, students are assigned in groups of two, which compete one against another, and each pair receives different questions that students have to solve in a limited amount of time.

In UZWEBMAT [Özyurt et al. (2012)], an e-learning environment to teach probability in a 10th grade mathematics course, delivers content according to the student's learning style, and adapts the content according to the student's knowledge using an expert system.

An Oscar Conversational ITS (CITS) is a sophisticated ITS that uses a natural language interface to enable learners to construct their knowledge through discussion [Latham, Crockett, and McLean (2014)]. Oscar CITS aims to simulate a human tutor by dynamically detecting and adapting to a student's learning style while directing the conversational tutorial. The Oscar CITS is independent of a particular model of learning styles and of the subject domain being taught.

TECH8 [Dolenc and Aberšek (2015)] is an intelligent and adaptive e-learning system that is designed modularly, based on a system for collecting a range of metadata and variables that are vital for the teaching process. Prepared in such a way, the proposed system supports individualization and differentiation; because of this, it can be adapted to each individual's level of knowledge and understanding of the subject matter.

4.2 The domain knowledge formalization

Domain knowledge, at first, was broadly represented in the form of *graphs* or *networks*. The main difference in each approach was what the nodes in those structures represent.

In ECSAI, nodes are units organized in a hierarchical network. The network of units also constitutes the domain knowledge in ELM-ART.

The DCG nodes are concepts that are formalized in a "and/or" domain knowledge concept graph. This is the simplest structure of domain knowledge and is achieved using only one type of connection – "a prerequisite of", which enables the sequencing of content [Brusilovsky and Vassileva (2003)]. Each node and each edge is associated with a set of educational contents (e.g., introduction, explanation, example, exercise or test) that describe different features of the concept and have a different role in the learning and teaching process. The structure of domain knowledge is used to create a courseware plan (a subgraph of domain knowledge) in order to achieve objectives (the learning of the concepts). This plan is called a "content plan" and the process is called "content planning".

ACE also adopts the domain knowledge concept graph where each concept is associated with different types of educational contents (text, example, demonstration, interactive environment, and test). A similar approach is adopted in INSPIRE, where each concept, from a concept network, is also associated with educational content (theory presentation, test, example, exercise, activity, and definition).

NetCoach requires its domain knowledge to be in the form of a concept network, where concepts represent pages that have educational content that is to be presented to the student. There are two types of relations in that concept network: prerequisite and conclusion. The teacher defines concepts that are prerequisites for learning a concept. Since prerequisite concepts have their own prerequisites, they are indirect prerequisites of a certain concept. A concluding concept is one whose prerequisite is a certain concept. Each domain knowledge concept is linked with questions. One question can be related to several concepts. Sets of questions provide for the evaluation of the student's knowledge about a concept.

The KBS Hyperbook System uses dependency graphs in which nodes are concepts (knowledge items), while edges are dependencies between concepts.

Since 2004, the *ontological approach* to the formalization of domain knowledge has been used. In APeLS, ACCT, ASM, iClass and ADOPTA, domain knowledge is expressed as an ontology.

APeLS and ACCT represent domain knowledge in the form of a concept ontology (called subject matter concept space), as a collection of high level concept descriptions, relationships and interrelationships in a content-independent way.

A concept path graph, in ASM, is a simple directed acyclic graph that represents the concept structure of the domain knowledge ontology. Concepts are selected from the concept graph path according to the relationship between the learning goals hierarchy and the domain knowledge ontology.

In ADOPTA, educational content is broken down into learning objects, which can be descriptions, tasks or questions. Learning objects are organized in an ontological semantic graph with two kinds of links: *is_a* for links between descriptions and *has_a* for links between tasks and questions and descriptions. It is necessary to form as many different kinds of learning objects that are suitable for any learning style, allowing teachers to generate different learning paths for different learning styles.

Several systems have no description of domain knowledge reported: GTE, AST, ATLAS, WINDS, ELP1+ELP2, ISCARE, UZWEBMAT, Oscar ITS, and TECH8, while in PAIGOS, domain knowledge is expressed only by references to on-line educational contents.

4.3 The student model

An *overlay model* [Carr and Goldstein (1977)] is generally used for student modelling. In the overlay model, the student knowledge is, in fact, a weighted subset of the expert knowledge.

ECSAI uses a simple overlay model (without weights). In ELM-ART, for each page that is visited, corresponding units in the overlay student model are marked as visited. After a test or problem solving task, all learned concepts from units are marked as known, and the process of making conclusions about the student's knowledge begins. INSPIRE's overlay student model remembers information that describes student interaction, which represents the attitude of students towards learning; it remembers the general information about the student (name, occupation, gender, learning style) and it is visible to the student who can control it. ASM uses the overlay model to determine the student's knowledge level and monitors the number of points achieved in tests, as well as, the number of attempts to solve those tests.

GTE uses a weighted overlay student model that includes estimates of how well the student understands the domain knowledge. A student's knowledge is presented in terms of topics which have been classified "not learned", "pretty well learned" and "well-learned"; these are the graded weights in the overlay model. The weight is determined after each knowledge test. The student model also remembers the objects that have already been given to the student, as well as the student's attributes (motivation, learning style, knowledge and control. The student's attributes are not changed automatically, but they are determined by the student or teacher.

The student model in DCG is a numerical overlay with the concept structure. That is, the student's knowledge of each concept is represented by a number within a specified interval. If a student cannot reach the threshold for a given concept, a new content plan is generated that avoids the difficult concept.

NetCoach uses a multi-layered overlay student model. The first layer describes whether the student has visited a page that contains a specific concept. The second layer contains information about answers to the questions related to a certain concept. The third layer describes whether it can be concluded that the student knows a concept based on the relationship between that concept and concepts that the student already knows. The fourth layer describes whether a student has designated a certain concept as known.

In some systems there is a *Bayesian probabilistic student model* (AST, KBS) and an *episodic student model* [Weber (1996)] (ACE, WINDS). Only ASM adopts a *stereotype model* which is used to represent student learning styles according to the Honey and Mumford model [Honey and Mumford (1992)]. This stereotype model is also used for presenting the learning material (visual, textual, auditory and mixed). More complex student modelling approaches allow more complex adaptation.

In the Bayesian probabilistic student model in the KBS Hyperbook System, nodes are concepts. Dependencies between concepts are represented as conditional probabilities.

In AST, every interaction with the system has an impact on the student model. The impact depends on the educational content and its parameters. Depending on the severity and importance, the reliability value of a concept in a Bayesian probabilistic student model increases. The reliability value indicates the system’s confidence as to how well the student understands what is being presented. The value takes into account the percentage of correct tests. Based on the importance and the value of the total weight, the student model is updated after each test. Severe tests with high importance have the greatest impact on the reliability value of the specific unit. If a student exceeds the limit set for the unit, the system concludes that the student knows the unit and its requirements.

The episodic student model in WINDS remembers all the educational events. These events are called episodes and include the student's actions, evaluations of their work by the teacher and the system, as well as any conclusions that the system has about the student’s knowledge level.

The student model in ACE combines the overlay model with the episodic approach. The student model consists of three parts: a profile, a knowledge model and an interest model. A profile contains information about the student’s language, media, interface settings, etc., and can always be changed. The knowledge model contains information about the units the student has worked on and those units which are marked as learned. Learned units have a reliability value that depends on the student's experience with the unit. The interest model includes interest clusters and dynamically sets hypotheses about the interests that students have. Hypotheses are based on rules that are associated with the student's episodic characteristics and certain assumptions about the student's interests.

The student model in ATLAS includes personal characteristics such as the student's age and language, and the relation between student and domain knowledge. The student model consists of a set of simple and complex attributes in the form of a tree, which the teacher can modify and use to define the conditions that will determine the courseware structure.

The student model in APeLS contains foreknowledge and learning objectives, and facilitates a personalized delivery of contents based on the student's experience and learning goals (sets of attributes that the user should acquire in the learning and teaching process).

The student model in ADOPTA includes student goals and preferences, learning style and knowledge. Details are not provided by ADOPTA's authors.

Each system that claims to record student goals in their student model, does not give examples of those goals; the literature is vague.

ACCT, iClass, PAIGOS, as well as, ELP1+ELP2, ISCARE, UZWEBMAT, Oscar CITS, and TECH8 have no student model described.

4.4 Courseware generation and adaptation

There are several stages in the courseware generation process: (1) the generation of courseware elements, (2) the selection and sequencing of courseware elements and (3) the presentation of courseware elements. Each phase is briefly described and its implementation in different adaptive e-learning systems elaborated.

4.4.1 The generation of courseware elements

The first stage is the generation of courseware elements. That can either be done by the teacher or the elements can be generated automatically. In most systems, courseware elements are *generated by the teacher* (GTE, ECSAI, AST, KBS Hyperbook, ATLAS, INSPIRE, WINDS, NetCoach, APeLS, ACCT, ADOPTA, ELP1+ELP2, ISCARE, UZWEBMAT, Oscar CITS, TECH8).

Courseware elements in GTE and ATLAS are tasks, a set of activities that need to be achieved during the learning and teaching process. In GTE tasks can be: "give an example", "give an opposite example", "give an exercise", "give an overview", etc. Tasks are generic because the same could occur in a variety of knowledge domains and could be given to different students. In ATLAS the courseware is hierarchically organized. Tasks are defined by specifying values of the following attributes: name, description, type (theoretical, practical, example), complexity (indivisible and complex), and requirements (foreknowledge). Tasks also include a list of multimedia elements (text, images, video, audio, animation, etc.) used to generate HTML pages.

Courseware elements in ECSAI are units that are a collection of content items, practice items, and assessment items that are combined to support a single learning objective. Each unit is assigned with a description that includes: label, type (presentation, examples, exercises, etc.), text description, prerequisites (completed lessons, student's understanding of certain domain knowledge elements) and post conditions (changes in the student model after completing a unit).

AST has different types of courseware elements that can be lessons, sections, subsections and concepts. Each concept has several descriptors: text, examples, demonstrations, tests and interactive playgrounds (e.g., a spreadsheet simulation in which the student requests data analysis and watches a formula filling up with the computed values). There are three levels of text: from basic information to very detailed text with messages from experts. In addition, each courseware element has prerequisites (units which the student must know) and consequences (the possible impact on other units). Tests and prerequisites are determined according to their importance for the courseware element.

The courseware in KBS Hyperbook consists of units that correspond to parts of the book and are semantically interconnected. They are indexed by concepts. On every page of the hyper book there are one or more major concepts. For each concept there is a unit for which the particular concept is the main one. There are other units that involve the same concept, but they are not as important. The courseware also consists of units that contain a description of the project (exercises or examples of solved problems). These units are indexed by those concepts that the student needs in order to be successful in realizing the project.

INSPIRE's courseware elements are units that include learning goals, outcome concepts and those educational contents that are associated with the outcome concepts and their prerequisite concepts. The learning goal is a set of domain knowledge concepts. Concepts that are the most important for achieving the goal are called the outcome concepts. To achieve the goal, the student must learn all the outcome concepts. Prerequisites and related concepts are associated with each outcome concept. Preconditions must be learned together with the educational content of the outcome concept. Related concepts are primitive concepts that are used in an educational content or outcome concept.

WINDS contains four types of courseware elements: (1) a unit which is the highest level element and can have only subunits, (2) subunits that are the basis for courseware structuring, (3) learning objects which are the basic pieces of information with templates for various educational purposes, and (4) concepts that are the basic terms in the dictionary that allow heterogeneous content to be connected providing individual paths through the educational contents. The learning objects are: (1) paragraphs which consist of content blocks and may have different pedagogical functions, (2) exercise, and (3) objective type items which evaluate the student's knowledge.

In APeLS, courseware is organized into sections, modules and units. Courseware is a series of steps through the content in which each step consists of sets of educational contents that have the same learning goal (they belong to the same candidate group). Each candidate group contains learning objects that meet the same content requirements. Learning objects in the candidate group can differ technically, educationally or by any other feature that can be adapted. The separation of content from its narratives (descriptions) promotes re-usability of learning objects.

In ACCT, courseware elements are concept narratives. The narrative model includes the semantics of the adopted pedagogical strategy. It describes the logic for the selection and delivery of learning activities and concepts. The concept narrative allows the teacher to apply aspects of pedagogical strategy to certain parts of the adaptive courseware.

In ADOPTA, courseware elements are learning objects that the teacher places in pages of the storyboard (in the courseware). For each learning object, the teacher defines the knowledge level coefficient that is used by the mechanism that adaptively selects content. This selection is based on the students' knowledge which they have demonstrated on the last test (a control page).

In UZWEBMAT, while preparing courseware elements (learning objects) special attention is paid to ensure that learning objects take into consideration the characteristics of each learning style. For example, figures, flow charts, pictures and animations are appropriate for those learning visually. Voiced instructions, warnings and feedback are for those who prefer to learn audibly. Similarly, learning objects are also constructed to exploit interactive animations for those who prefer to learn kinaesthetically.

In some systems there are *indications of automatism in the generation of courseware elements*, but are still heavily influenced by the teacher (ELM-ART, ACE), while *complete automation* is present in the DCG, ASM, iClass, and PAIGOS systems. For those systems, the generation process is not described in detail.

In ELM-ART, courseware elements are units that are hierarchically organized into lessons, sections, subsections and terminal pages. Terminal pages may contain new concepts or problems that need to be addressed.

The courseware structure in ACE is the conceptual network of units that can be sections (which may include other units) or concepts. Each learning unit might include the prerequisite units that students must know before they begin learning a particular unit. Prerequisite units are weighted according to their importance.

In DCG, courseware elements are HTML pages related to the domain knowledge concepts. The courseware generation takes into account the student's knowledge and respects the diversity in the manner and pace of acquiring knowledge.

In PAIGOS, courseware elements are a structured series of references to educational contents. The same educational content can be used in multiple courseware elements and support the achievement of different learning objectives.

The courseware in TECH8 consists of learning steps and an assessment. Each learning step has a branched structure, with multiple formative assessments. The formative assessments are didactically designed in such a way that, with the help of the answers provided, TECH8 can figure out exactly, any gap in the student's knowledge, and on that basis can provide the student with additional content/knowledge.

4.4.2 The selection and sequencing of courseware elements

The second phase of courseware generation is the selection and sequencing of courseware elements. This process can be *static* (executed only one time before students begin the learning and teaching process) or *dynamic* (can be repeated several times during the learning and teaching process).

Most of the systems found in this search (ECSAI, DCG, ELM-ART, AST, ACE, KBS Hyperbook, ATLAS, INSPIRE, NetCoach, APeLS, ACCT, ASM, ADOPTA, ISCARE, UZWEBMAT, TECH8) have their courseware elements *dynamically selected and sequenced*.

Once the student has specified their learning goals, the ECSAI determines the first teaching unit which has to be presented and shows it to the student. It changes the student model according to the student's behaviour during these interactions, and then presents the next unit. The process continues until all objectives are achieved or until all teaching units are presented.

When the student model and learning goals are known, a content planner in DCG generates paths through the domain knowledge graph and, in that way, connects concepts familiar to the student with learning objectives. These paths are a blueprint for presentation planning. Afterwards, a presentation planner chooses educational contents associated with the selected domain knowledge concepts and defines the order and way in which those contents will be presented to the student.

In ELM-ART, each unit contains static slots for both the text that will be displayed and for the information that can be used for connecting units with concepts. Units can also have dynamic slots for tests that contain questions that the student needs to answer, in addition to the problem description. During the process of making conclusions about the student, all concepts that were prerequisites for a unit are considered to be closed. The information from the dynamic slots is used to generate automatically, an optimal learning path for the student.

AST has a default teaching strategy (learning by example, learning by reading texts, and learning by doing) for each type of concept, though teachers can change it if necessary. Also, there are rules for each strategy that allow adaptive selection of teaching strategies depending on the student's characteristics and the type of concepts that are taught. The system monitors which combination and sequence of educational content a student often uses, and changes its teaching strategy accordingly. AST determines which units the student should learn next, based on the Bayesian probabilistic overlay model and the requirements for possible next units. The system first throws away those units for which the student has not fulfilled the prerequisites, and then observes the reliability and weight of units for which the student has fulfilled the prerequisites. The next unit selected is the one in which the student has reliably fulfilled the most important conditions.

In ACE, teachers must explicitly define which teaching strategies are to be used for teaching which types of concepts. A component for adaptive sequencing tries to keep students on a path that is determined by the student's current knowledge. Furthermore, the sequencing adapts to the interests specified by the student. Each student's actions can have an impact on the dynamically generated optimal. For example, the completion of the test for a particular section allows the student to skip all the concepts it covers.

In the KBS Hyperbook, students select a learning goal and the system proposes project units that they have to work on in order to achieve their goal (adaptive project selection). The system can also suggest learning goals that are consistent with the student's knowledge (adaptive goal selection); it proposes projects and then generates the courseware. Units that are already known are discarded from the courseware, and all others are sequenced according to their complexity. If a student has no prior knowledge needed for working on a project, then units related to the missing elements of knowledge are added at the beginning of the courseware.

In ATLAS, HTML pages, presented to the student, are dynamically generated. Descriptions of components that will appear on pages are included as values in HTML fields. These descriptions must include the type of multimedia element that will be located on the page, as well as its position on the page. A complex task can be broken down using decomposition rules. Each rule has a name, and contains information about the task that has to be divided into subtasks; there is also a subtask list and a word that describes the sequence of subtasks (AND, ANY, XOR, OR).

Based on the learning objective that the student chooses, INSPIRE generates courseware elements (lessons) compliant with the student's learning style and knowledge. The students can affect the process of courseware element generation by expressing an opinion about their characteristics or about the courseware element content.

In NetCoach, the learning goal is a set of concepts that students must learn. All prerequisites are automatically determined and appropriate courseware elements (pages) are proposed. Based on concept descriptions, all courseware elements are generated individually in accordance with a student model.

An adaptive engine in APeLS co-ordinates the content model, the student model and the narrative model; at execution time, it dynamically creates adaptive courseware. An adaptive engine selects a concept narrative from a candidate group that is the best for delivery. The narrative model contains adaptation rules that teachers define. Those rules that govern the personalized courseware generation are separate from the content and are included in the courseware.

In order to make courseware adaptive in ACCT, the teacher must assign descriptions (narrative attributes) to concepts and learning activities. The description includes a feature of adaptation (knowledge, learning style, etc.), an adaptation technique (adaptive annotation, adaptive navigation support, etc.) and guidelines or usability description. The teacher determines which adaptation technique will be used for each feature.

In ASM, a learning paths graph is a simple directed acyclic graph that contains all possible learning paths (sequences of educational contents) that lead to achieving the goal. Based on the suitability function that assigns weights to edges in the learning paths graph, and on an algorithm for finding the shortest paths, the most appropriate learning path from the learning paths graph is selected. The chosen path is called an optimal learning path and it inherits relations from the domain knowledge structure and those relationships that exist between educational contents. The suitability function determines which educational contents are appropriate for a particular student, based on statistical analysis. It compares the characteristics of a learning object with the characteristics of students, and vice versa.

The chosen learning path in ADOPTA is the one with the maximum weight. Only the appropriate courseware elements are presented. At the end of each learning path, the achieved weight of that path is calculated and this determines whether a student can continue or must return to the beginning of the path.

ISCARE provides an adaptive methodology in which the different pairs of assignments are adapted for each round, trying to minimize the differences in score between the different pairs of students.

The expert system integrated into UZWEBMAT determines the route to be followed by the learner within the system in the widest sense. If a student does not complete a courseware element and reach an adaptivity point, the student is directed to the same courseware element but with a different learning style. If the student accomplishes the courseware element with the new learning style, the next courseware element adopts the new learning style. A similar approach is used in Oscar CITS.

TECH8 regularly assesses and analyses students' achievements and prepares the best selection of subject matter and the learning path for subsequent units. Simultaneously at this stage, the student's cognitive abilities are determined, according to which all future learning paths are adjusted (early differentiation). If a student is placed into a lower group at this unit (classified as a student with lower cognitive capabilities), additional content with additional explanation is automatically switched on in the following steps, so that the student may understand better.

Static selection and sequencing of courseware elements is present in the GTE, iClass and ELP1+ELP2 systems, while WINDS and PAIGOS have static selection and sequencing "determined by the learner's competencies and learning goals".

In GTE, tasks describe what has to be done, while teaching methods describe how it should be done. The GTE's adaptivity is reflected in the selection, from all the available teaching methods, of the most appropriate one for each task. Teaching methods divide tasks into subtasks. The final result of the task decomposition is a tree called the task structure. While GTE has a content model, which consists of themes, it is still a static skeleton around which presentations are built. Instead of dynamically deciding which theme to present, a full path of themes is generated.

A learning path in iClass consists of courseware elements (concepts) which are determined according to the chosen teaching strategy. The teacher determines the limits of adaptation. Improved adaptation is realized by applying different teaching strategies on the same domain knowledge. This means that the most appropriate teaching strategy is adaptively selected, regardless of what is being taught.

In ELP1+ELP2, the second level ELP2 allows teachers to select a subset of personalization parameters for each course. Then, the teachers combine the selected personalization parameters and decide how the learning material will be composed according to each possible value of the personalization parameters. The combination defined by the teachers is then used by the first level ELP1 to provide personalized courses.

In WINDS, an authoring system enables the teacher to define courseware by hierarchically linking pre-defined learning elements. The learning objects have a pre-defined sequence within a unit. Learning objects consist of content blocks that have a pre-defined sequence and a pre-defined pedagogical role. While each of these levels is fixed, relationships (such as prerequisites) and a student model can combine to provide adaptation as the student progresses through the course material.

Courseware generation in PAIGOS can be stopped at the level that determines the type of educational content, but does not specify exactly which educational content shall be taken into account. Specific educational contents are selected as late as possible, that is, when a student wants to see them.

4.4.3 The presentation of courseware elements

The third stage of courseware generation is the presentation of courseware elements. Presentation is the stage where the adaptivity of courseware is also realized. Most systems used *adaptive hypermedia techniques*, such as adaptive navigation support and adaptive annotation (ECSAI, ELM-ART, AST, ACE, KBS Hyperbook, INSPIRE, WINDS, NetCoach, ADOPTA).

ECSAIWeb [Sanrach and Grandbastien (2000)] is a web version of ESCA, which allows the design of intelligent tutoring systems where the presentation and navigation support are adaptive. The same is true for INSPIRE.

ELM-ART uses an adaptive annotation technique based on an extension of the "traffic light" metaphor. In this metaphor, a green dot in front of a link indicates recommended readings, a red dot indicates that the student might not have enough knowledge to understand the information contained at the end of the link, and a yellow dot indicates that there is no new knowledge at the end of the link. There is also the possibility of following the optimal learning path that is determined by the student's current knowledge. The following page in the optimal learning path corresponds to the one that is marked as "suggested". A similar strategy is used in AST which has adaptive navigation support ("traffic lights"), adaptive sequencing and adaptive testing. The KBS Hyperbook also uses adaptive navigation support to adapt lists of units in the courseware.

The text in ACE can contain hyperlinks to concepts that the student has not yet learned. If the text is presented to the student for the first time, then it will not contain hyperlinks to related, but unlearned concepts. After working in the system for a while, ACE will present all the hyperlinks to the learned concepts and those concepts that the student is ready to learn. ACE implements two methods of adaptive navigation support: adaptive annotation and incremental linking of hyperlinks. Adaptive annotation of hyperlinks provides the student with additional information about the content behind a hyperlink (using different colours). Annotation is adapted to the student, taking into account the student's knowledge and the relations between units that must be learned (concepts that were visited, concepts of which the student has no prior knowledge, preferred concepts, and units that are not recommended, but do not require any further knowledge).

In WINDS, the courseware structure and the student model allow adaptive navigation and presentation. Since learning objects are linked together, different adaptation techniques can be used: direct guidance, link sorting, link hiding, and link annotation. Learning elements consist of content blocks that have a pre-defined sequence and a pedagogical role, which allow applications to use various adaptation methods such as: additional explanation, prerequisite explanation, comparative explanation, explanation variants, and link sorting.

NetCoach uses adaptive hyperlink annotation. The next page that is recommended to the student is dynamically generated based on learning objectives and the student's knowledge. Students receive a warning if they select a page for which they do not have all the prerequisites (alerts can be switched off).

During the courseware presentation in ADOPTA, adaptive navigation support, adaptive annotation and adaptive content selection are used. The teacher must define the courseware parameters so that it can adapt to the student's learning style and test results.

Approaches different from adaptive hypermedia techniques are used during the presentation of courseware elements in the systems GTE, DCG, APeLS and ASM.

The dynamic selection of teaching methods is crucial for GTE's adaptation. When a particular task has to be executed, GTE selects the most appropriate teaching method that is assigned for this task. This selection is based on a value calculated from conditions that describe which teaching method is applicable in a particular context.

In DCG, the content planner itself cannot determine how to present the selected content to the student. Therefore, DCG uses a “presentation plan” for each concept. Teaching methods allow the system to present dynamically the contents associated with certain concepts in a way that is adapted to the student (which educational contents to select and how to sequence them). During the courseware presentation, if the students correctly answer the questions in the test, they continue with the learning and teaching process according to a defined plan that does not change. If the student shows a lack of knowledge, it is necessary to pre-plan the courseware. Pre-planning is initially done either at the presentation level by changing the sequence of educational contents, or by changing the teaching methods for presenting a concept. If a student shows a lack of knowledge again, the content planner generates a new set of concepts that lead to the target concept according to the current student’s knowledge.

In APeLS, presentation planning is limited to selecting a candidate group. A candidate group is pre-defined with a variety of different structures and formats for educational contents.

In ASM, a learning paths graph is made by replacing each concept in a concepts graph path with the appropriate set of educational contents.

Special attention was given to the introduction of content, which is followed by pictures, animations and a range of interactive elements. The content is prepared in such a way that the students can increase their knowledge, eliminate gaps in knowledge and progress from simpler subjects to more complex ones.

4.5 The feature of adaptation

The main mechanisms for adaptation are based on *learning style* (AST, ATLAS, INSPIRE, WINDS, APeLS, ACCT, ASM, ADOPTA, ELP1+ELP2, UZWEBMAT, Oscar CITS) and *learning objectives* (KBS Hyperbook, NetCoach, ACCT, iClass). Several systems adapt courseware according to the student’s preferences or interests (DCG, ACE, WINDS, iClass), cognitive style (DCG) or student’s behaviour (ECSAI). For some systems, the feature of adaptation is not explicitly stated (GTE, PAIGOS).

For a large number of systems they adapt according to the student’s *knowledge*. This adaptation does not interfere with the changing of the courseware element content, but rather it refers to the selection and sequencing of courseware elements (ECSAI, DCG, ELM-ART, AST, ACE, KBS Hyperbook, INSPIRE, WINDS, NetCoach, APeLS, ACCT, ASM, iClass, ADOPTA, ELP1+ELP2, ISCARE).

4.6 The level of adaptation

Courseware adaptation can be achieved either by selecting and sequencing the courseware elements or by adapting the presentation of courseware elements. It is crucial for adaptation to include the smallest possible courseware granule that can be manipulated. The lower the level of adaptation, the "finer" and "more sensitive" it is, and ultimately, the more successful is the adaptation.

Systems found in this research generally use *units* (ECSAI, ELM-ART, AST, ACE, KBS Hyperbook, ELP1+ELP2), *pages* (NetCoach), *lessons* (INSPIRE), *tasks* (ATLAS, ISCARE), *themes* (GTE), *learning objects* (WINDS, iClass, ADOPTA, UZWEBMAT, Oscar CITS) or *candidate groups* (ApeLS, ACCT) as a level of adaptation. Only two systems (DCG, ASM) use the smallest element possible for the adaptation level – a domain knowledge *concept*.

4.7 Knowledge testing

Testing students' knowledge is the central, but time-consuming, feature provided in e-learning systems. In the majority of e-learning systems, the teacher is the one who has to devise the questions that check the students' knowledge. A disadvantage of that approach is that it is hard work for teachers to enter manually the question, determine the evaluation method, select the correct and incorrect answers, and write the feedback.

So, while researching the knowledge testing in adaptive systems, it is important to distinguish whether the system itself generates the questions (*automatic*) or the *teacher* enters questions *manually*. In some systems there is no knowledge testing (GTE, ECSAI, KBS Hyperbook, ATLAS, APeLS, ACCT, ELP1+ELP2, ISCARE, Oscar CITS). In others the questions are created by the teacher (DCG, ELM-ART, AST, ACE, INSPIRE, WINDS, NetCoach, ADOPTA, UZWEBMAT, TECH8). In ASM, iClass and PAIGOS, the knowledge assessment is mentioned, but it is not described.

There is no system that completely and automatically generates questions based on its domain knowledge. Only ACE gives a short initial knowledge test to students that is dynamically generated and includes all the prerequisite first-level concepts and their associated tests. In ACE, each test can be associated with multiple units, and its weight is determined by the importance of these units. The severity of the test will influence how a student's response influences the student model. Throughout the courseware, the student's knowledge is checked so that the system can dynamically adapt to the student's changing knowledge.

In DCG, the students can test their knowledge about any concept at any time. Exercises and tests are presented with a set of smaller units that contain pre-stored correct answers, help, explanation, etc.

ELM-ART uses three types of question: yes/no, multiple choice and essay. For each question it is necessary to specify the precise explanation for the correct answer. The students answer questions until they have answered correctly a given number of questions. Only then can the system record that the concepts from the unit have been learned.

In AST, there are four types of questions that are used: yes/no, multiple choice, essay and fill-in-the-blank. Each test is associated with one or more concepts based on their importance, and has a value for the overall weight.

NetCoach allows the presentation of tests in different question formats: multiple choice, gap-filling, and essay. The test's feedback contains a specification for the correct answer and why it is correct; feedback is defined by the teacher.

In PAIGOS, knowledge is tested using the automatically generated tests that have been placed at control points within the courseware. Tests are automatically generated by selecting questions that relate to learning objects found in pages that the student has been learning. The teacher sets the test's passing score.

An adaptive assessment module was integrated into UZWEBMAT. Item Response Theory (IRT) was used instead of classical testing theory within UZWEBMAT for the end of subject tests and the end of unit tests.

Initial tests (DCG, AST, ACE) or *questionnaires* (ECSAI, AST, ACE, ATLAS, APeLS) are used for student model initialization. Student model initialization is very important for successful adaptation of courseware and it should be conducted as soon as a student starts the learning and teaching process.

In AST, all new students must fill out a questionnaire about their preferences, learning style and goals. Students can indicate what type of content and teaching strategies suit them most. They can also determine the level of detail in the text.

In ACE's introductory questionnaire, students are asked about their preferences relating to the content that they will be taught and about their common interests. Students can specify teaching strategies that suit them. They can also assess their expertise in the domain knowledge (novice, experienced, expert).

Before APeLS can create a personalized courseware, it must have adequate information about the student. This information is obtained through a questionnaire about prior knowledge and learning style. The student can access the same questionnaire anytime during the learning and teaching process; this causes the student model to be modified and the personalized courseware is re-generated.

In TECH8, the initial learning step includes elements of diagnostic assessment, the majority of the formative assessment and part of the summative assessment. Initially, all students begin with the same subject matter, and the system adjusts to their needs as the learning progresses. The student's initial classification can change during the course of study. After the first unit, an intelligent agent analyses the student's progress and alerts the student to any mistakes. At the same time, it gives the student recommendations on how to improve learning. This agent is important, mostly because it can automatically analyse the student's learning path and warn them of the most frequent mistakes that occur during learning.

A comparison of systems that generate adaptive courseware is given in [Tab. 4].

5 Conclusion

This literature review of adaptive courseware generation has analysed a representative segment of the published material through summary, classification, and comparison of prior research studies, reviews of literature, and theoretical articles. This paper has considered the critical points of current knowledge including substantive findings, as well as, theoretical and methodological contributions to this topic. This "state of the art" review has identified major methodological flaws or gaps in research, inconsistencies in theory and findings, and areas or issues pertinent to future study. These aspects are now addressed.

No.	System	Courseware elements generation (teacher/ automatic)	Selection and sequencing (static/ dynamic)	Adaptation	Test generation (teacher/ automatic)
1	GTE	teacher	static	applicability of teaching methods	none
2	ECSAI	teacher	dynamic	student knowledge and behaviour, adaptive presentation, adaptive navigation support	none
3	DCG	automatic	dynamic	student knowledge and cognitive style	teacher
4	ELM-ART	teacher partly automatic	dynamic	student knowledge, adaptive annotation (t)	teacher
5	AST	teacher	dynamic	student knowledge and learning style, adaptive navigation support (t), adaptive sequencing, adaptive testing	teacher
6	ACE	teacher partly automatic	dynamic	student knowledge, interests and preferences, adaptive annotation, incremental linking of hyperlinks	teacher
7	KBS	teacher	dynamic	student knowledge and goal, adaptive navigation support, adaptive annotation	none
8	ATLAS	teacher	dynamic	student profile and actions	none
9	INSPIRE	teacher	dynamic	student knowledge and learning style, adaptive presentation adaptive navigation support	teacher
10	WINDS	teacher	static	adaptive presentation adaptive navigation support	teacher
12	NetCoach	teacher	dynamic	student knowledge and learning style, adaptive annotation	teacher
11	APeLS	teacher	dynamic	student knowledge and learning style, descriptions and candidate groups	none
13	ACCT	teacher	dynamic	student knowledge, goal and learning style, descriptions and candidate groups	none
14	ASM	automatic	dynamic	student knowledge and learning style	none
15	iClass	automatic	static	learning goal, student and teacher preferences	none
16	PAIGOS	automatic	static partly dynamic	student knowledge, motivation and ability level	teacher
17	ADOPTA	teacher	dynamic	student knowledge and learning style, adaptive navigation support	teacher
18	ELP1+ELP2	teacher	static	student knowledge and learning style	none
19	ISCARE	teacher	dynamic	student knowledge	none
20	UZWEBMAT	teacher	dynamic	student knowledge and learning style	teacher
21	Oscar CITS	teacher	dynamic	learning style	teacher
22	TECH8	teacher	dynamic	student knowledge	teacher

Table 4: Comparison table of systems that generate adaptive courseware

This paper has not considered the following aspects of learning using adaptive courseware systems. For example, students are generally restricted by the system as to which courseware elements should come next; this is useful for those students who have difficulty making decisions. But it might be inappropriate for those students who do not understand a topic learned, or for those who want to investigate why they are learning that topic. A cause for concern is the amount by which the teacher can observe each student's progress; some students want to keep their learning activity private and may resist using the system to the full. Nevertheless, such “snooping” may help to identify those students who are struggling, thus enabling teachers to intervene before it is too late. Some students need to know why an answer is wrong, and they will deliberately select wrong answers to confirm that they understand why the wrong answer is incorrect. The student should not be penalised (and marked down) for such investigations. With adaptive courseware, strict adherence to the learning and teaching process is very important, thus denying the student the opportunity for explorative learning. Furthermore, not all students want to learn, or be forced to learn, all that is contained in a course. They may prefer to learn all those parts that are easiest to learn and forego those that need an excessive amount of time to master. However, an adaptive system does enable students to monitor how much of the course has been learnt.

An adaptive courseware and its elements are very hard to produce if it is performed manually by teachers. The difficulty lies in the fact that teachers have to define a great number of applicable teaching strategies, all the educational contents that are to be learnt, the testing regime, not to mention the sequencing and presentation appropriate for each student. The automatic extraction of questions from the knowledge content, using natural language processing techniques, would be useful.

This review has highlighted a number of barriers that obstruct future developments in adaptive courseware generation. More importantly, it has identified concrete guidelines for the current understanding of the adaptivity concept; it has uncovered a range of strategies that can be employed to overcome these barriers. These barriers can be overcome by a model that fulfils the following demands: automatic generation of courseware elements based only on the domain knowledge structure, dynamic selection and sequencing of courseware elements in order to increase adaptivity primarily driven by the knowledge itself and, finally, automatic generation of questions and knowledge tests derived only from the domain knowledge structure.

The model that encompasses all these demands should follow recent trends with the domain knowledge formalized using an ontological approach. It should combine an overlay model with a Bayesian probabilistic model for both student and stereotypes. This would provide for a more complex student modelling approach, which would enable more complex adaptation.

The teacher should have no effect on the generation of courseware elements. The generation of courseware elements should be fully automated based on the domain knowledge ontology. These elements should vary in complexity, enabling the student’s knowledge to govern the adaptivity.

The student’s knowledge provides an unambiguous and stable base for realizing adaptivity, with Bloom’s knowledge taxonomy as an underpinning concept.

Therefore, the knowledge level should be defined according to Bloom's taxonomy. In this way, in the proposed model adaptivity realization is different from the one used in adaptive educational hypermedia systems. Adaptation could be realized, for example, using statement templates for learning.

Courseware adaptation can be achieved by selecting and sequencing courseware elements or by adapting the presentation of courseware elements. It is crucial for adaptation to include the smallest possible courseware or domain knowledge granule that can be manipulated. The lower the level of adaptation, the "finer" and "more sensitive" it is, and ultimately, the more successful is the adaptation. Therefore, the smallest possible elements for adaptation should be used. Since the domain knowledge concept is an "atomic particle" of knowledge and in this sense indivisible, it should be used as the level of adaptation. In this way, the proposed approach would maximize adaptivity.

Automatic knowledge testing excludes any interference from the teacher. Questions should be automatically generated and assigned to knowledge tests that are also evaluated automatically. Questions should be generated based on concepts and relations from the domain knowledge ontology using question templates (for creating objective type tasks). Question difficulty should also be consistent with the student's knowledge level (using Bloom's taxonomy).

Acknowledgement

This paper describes the results of research being carried out within project 177-0361994-1996 Design and evaluation of intelligent e-learning systems within the program 036-1994 Intelligent Support to Omnipresence of e-Learning Systems, funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

We thank the anonymous referees who helped us during the process of refining this paper.

References

[Albert 2001] Albert, D.: "E-learning Future-The Contribution of Psychology"; In Catching the Future: Women and Men in Global Psychology, Proceedings of the 59th Annual Convention, International Council of Psychologists, Winchester, England, (2001), 30–53

[Anderson 1988] Anderson, J.R.: "The expert module"; In Polson M. C., Richardson J. J. (Eds.) Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates Publishers, London (1988), 21–53

[Anh, Ha, and Dam 2008] Anh, N.V., Ha, N.V., Dam, H.S.: "Constructing a Bayesian Belief Network to Generate Learning Path in Adaptive Hypermedia System"; Journal of Computer Science and Cybermetics, 24, 11 (2008), 12-19

[ASTD 2001] ASTD: "A vision of E-learning for America's workforce: Report of the Commission on Technology and Adult Learning"; Commission On Technology And Adult Learning, (2001)

[Beaumont, and Brusilovsky 1995] Beaumont, I., Brusilovsky, P.: "Educational applications of adaptive hypermedia"; , In K. Nordby, P. Helmersen, D. J. Gilmore, & S. A. Arnesen (Eds.), Human-Computer Interaction, Proceedings of Interact'95, Lillehamer, Norway, London, Chapman & Hall (1995), 410–414

[Beck, Stern, and Haugsjaa 1996] Beck, J., Stern, M., Haugsjaa, E.: "Applications of AI in Education"; Crossroads, 3, 1 (1996), 11–15

[Benyon, and Murray 1993] Benyon, D., Murray, D.: "Adaptive systems: From intelligent tutoring to autonomous agents"; Knowledge-Based Systems, 6, 4 (1993), 197–219

[Bloom (1956)] Bloom, B.S. (Ed.) “Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain”, New York, Toronto: Longmans, Green (1956)

[Bhaskar, Das, Chithralekha, and Sivasathya 2010] Bhaskar, M., Das, M.M., Chithralekha, T., Sivasathya, S.: "Genetic Algorithm Based Adaptive Learning Scheme Generation For Context Aware E-Learning"; IJCSE International Journal on Computer Science and Engineering, 2, 4 (2010), 1271–1279

[Bontchev, Vassileva, Chavkova, and Mitev 2009] Bontchev, B., Vassileva, D., Chavkova, B., Mitev, V.: "Architectural design of a software engine for adaptation control in the ADOPTA e-learning platform"; , Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, ACM (2009), 1–6

[Brusilovsky 1996] Brusilovsky, P.: "Methods and Techniques of Adaptive Hypermedia"; User Modeling and User Adapted Interaction, 6, 2-3 (1996), 87–129

[Brusilovsky 1998] Brusilovsky, P.: "Adaptive educational systems on the world-wide-web: A review of available technologies"; In the Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, (1998)

[Brusilovsky, and Nijhawan 2002] Brusilovsky, P., Nijhawan, H.: "A framework for adaptive e-learning based on distributed re-usable learning activities"; In Driscoll, M. and Reeves, T. C. (Eds.) Proceedings of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, AACE, (2002), 154-161.

[Brusilovsky, and Vassileva 2003] Brusilovsky, P., Vassileva, J.: "Course sequencing techniques for large-scale web-based education"; International Journal of Continuing Engineering Education and Life Long Learning, 13, 1 (2003), 75–94

[Brusilovsky, Schwarz, and Weber 1996] Brusilovsky, P., Schwarz, E., Weber, G.: "ELM-ART: An intelligent tutoring system on World Wide Web"; Intelligent Tutoring Systems, Springer (1996), 261–269

[Burns, and Capps 1988] Burns, H.L., Capps, C.G.: "Foundations of intelligent tutoring systems: An introduction"; In Polson M. C., Richardson J. J. (Eds.) Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates Publishers, London (1988), 1–19

[Cabada, Barrón Estrada, and Reyes García 2011] Cabada, R.Z., Barrón Estrada, M.L., Reyes García, C.A.: "EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network"; Expert Systems with Applications, 38, 8 (2011), 9522–9529. doi:10.1016/j.eswa.2011.01.145

[Carbonell 1970] Carbonell, J.R.: "AI in CAI: An artificial-intelligence approach to computer-assisted instruction"; Man-Machine Systems, IEEE Transactions on, 11, 4 (1970), 190–202

[Carchiolo, Longheu, Malgeri, and Mangioni 2003] Carchiolo, V., Longheu, A., Malgeri, M., Mangioni, G.: "Automatic generation of learning paths"; In Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems ICECS 2003, IEEE, Vol. 3 (2003), 1236–1239

[Carr, and Goldstein 1977] Carr, B., Goldstein, I.P.: "Overlays. A theory of modeling for computer-aided instruction, AI Lab Memo 406"; Massachusetts Institute of Technology, Cambridge, Massachusetts, (1977)

[Carrier, and Jonassen 1988] Carrier, C.A., Jonassen, D.H.: "Adapting courseware to accommodate individual differences"; In Jonassen D.H. (Ed.) Instructional designs for microcomputer courseware, Hillsdale, NJ: Erlbaum, (1988), 203–226

[Chang, Chang, Heh, and Liu 2008] Chang, M., Chang, A., Heh, J.S., Liu, T.C.: "Context-aware learning path planner"; WSEAS Transactions on Computers, 7, 4 (2008), 316–325

[Chapelle, and Mizuno 1989] Chapelle, C., Mizuno, S.: "Student's strategies with learner-controlled CALL"; Calico Journal, 7, 2 (1989), 25–47

[Conlan, Wade, Bruen, and Gargan 2002] Conlan, O., Wade, V., Bruen, C., Gargan, M.: "Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning"; , in P. D. Bra, P. Brusilovsky, and R. Conejo (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science 2347; Springer Berlin Heidelberg (2002), 100–111

[Corno, and Snow 1986] Corno, L., Snow, R.E.: "Adapting teaching to individual differences among learners"; In Handbook of research on teaching, 3, (1986), 605–629

[Cronbach, and Snow 1977] Cronbach, L., Snow, R.: "Aptitudes and Instructional Methods: A Handbook for Research on Interactions"; New York, NY, USA: Irvington, (1977)

[Dagger, Wade, and Conlan 2004] Dagger, D., Wade, V., Conlan, O.: "Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit (ACCT)"; In the Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2004, De Bra P. and Nejdl W. (Eds.), Berlin: Springer Verlag, Eindhoven, The Netherlands, (2004) 55–64

[Dagger, Wade, and Conlan 2005] Dagger, D., Wade, V., Conlan, O.: "Personalisation for all: Making adaptive course composition easy"; Educational Technology & Society, 8, 3 (2005), 9–25

[Dick, Carey, and Carey 2009] Dick, W., Carey, L., Carey, J.O.: "The Systematic Design of Instruction"; Merrill/Pearson (2009)

[Dijkstra, Krammer, and Van Merrienboer 1992] Dijkstra, S., Krammer, H.P., Van Merrienboer, J.G.: "Instructional models in computer-based learning environments"; Springer-Verlag New York, Inc. Secaucus, NJ, USA (1992)

[Dolenc, and Aberšek 2015] Dolenc, K., Aberšek, B.: "TECH8 intelligent and adaptive e-learning system: Integration into Technology and Science classrooms in lower secondary schools"; Computers & Education, 82, (2015), 354–365

[Essalmi, Ayed, Jemni, Kinshuk, and Graf 2010] Essalmi, F., Ayed, L.J.B., Jemni, M., Kinshuk, Graf, S.: "A fully personalization strategy of E-learning scenarios"; Computers in Human Behavior, 26, 4 (2010), 581–591

[Felder, and Silverman 1988] Felder, R.M., Silverman, L.K.: "Learning and teaching styles in engineering education"; Engineering education, 78, 7 (1988), 674–681

[Gagne, and Briggs 1979] Gagne, R.M., Briggs, L.J.: "Principles of instructional design"; Holt Rinehart and Winston (1979)

[Garcia 1989] Garcia, M.B.: "Focus on Teaching"; Rex Bookstore, Inc. (1989)

[Gascuena, Fernandez-Caballero, and Gonzalez 2006] Gascuena, J.M., Fernandez-Caballero, A., Gonzalez, P.: "Domain ontology for personalized e-learning in educational systems"; In the Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, (2006), 456–458

[Gavignet 1991] Gavignet, E.: "Environnement de conception de système d'apprentissage: une modélisation de la connaissance pédagogique, PhD thesis, de l'Université de Nancy 1"; (1991)

[Gilbert, and Han 2002] Gilbert, J.E., Han, C.Y.: "Arthur: A personalized instructional system"; Journal of Computing in Higher Education, 14, 1 (2002), 113–129. doi:10.1007/BF02940953

[Glaser, and Nitko 1970] Glaser, R., Nitko, A.J.: "Measurement in Learning and Instruction"; In Thorndike R. L. (Ed.), Educational measurement (2nd ed), Washington, DC: American Council on Education. (1970), 625–670

[Grandbastien 1999] Grandbastien, M.: "Teaching expertise is at the core of ITS research"; International journal of artificial intelligence in education, 10, 3-4 (1999), 335–349

[Grandbastien, and Gavignet 1994] Grandbastien, M., Gavignet, E.: "ECSA: An Environment to Design and Instantiate Learning Material"; In de Jong T. and Sarti L. (Eds.) Handbook of Design and Production of Multimedia and Simulation-based Learning Material, Kluwer Academic, Publishers, Dordrecht, The Netherlands (1994), 31-44

[Grigoriadou, Papanikolaou, Kornilakis, and Magoulas 2001] Grigoriadou, M., Papanikolaou, K., Kornilakis, H., Magoulas, G.: "INSPIRE: an intelligent system for personalized instruction in a remote environment"; In the Proceedings of Third workshop on Adaptive Hypertext and Hypermedia, Vol. 14. (2001), 13–24

[Grunst 1993] Grunst, G.: "Adaptive hypermedia for support systems"; In Schneider-Hufschmidt, M., Khme, T., Malinowski, U. (Eds.): Adaptive user interfaces: Principles and practice, North-Holland, Amsterdam, (1993), 269–283

[Henze, and Nejdl 1999] Henze, N., Nejdl, W.: "Adaptivity in the KBS Hyperbook System"; In the Proceedings of the 2nd Workshop on User Modeling and Adaptive Systems on the WWW, Toronto, Canada, (1999), 67-74

[Hockemeyer, Held, and Albert 1997] Hockemeyer, C., Held, T., Albert, D.: "RATH - A Relational Adaptive Tutoring Hypertext WWW-Environment Based on Knowledge Space Theory"; In C. Alvegård (Ed.), Proceedings of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering, Göteborg, Sweden (1997), 417–423

[Honey, and Mumford 1992] Honey, P., Mumford, A.: "The manual of learning styles"; Maidenhead, Peter Honey (1992)

[Kaplan, Fenwick, and Chen 1993] Kaplan, C., Fenwick, J., Chen, J.: "Adaptive hypertext navigation based on user goals and context"; User Modeling and User-Adapted Interaction, 3, 3 (1993), 193–220. doi:10.1007/BF01257889

[Karampiperis, and Sampson 2004] Karampiperis, P., Sampson, D.: "Adaptive instructional planning using ontologies"; In the Proceedings of the IEEE International Conference on Advanced Learning Technologies, (2004), 126–130

[Karampiperis, and Sampson 2005] Karampiperis, P., Sampson, D.: "Adaptive learning resources sequencing in educational hypermedia systems"; Educational Technology & Society, 8, 4 (2005), 128–147

[Kayama, and Okamoto 1998] Kayama, M., Okamoto, T.: "A mechanism for knowledge-navigation in hyperspace with neural networks to support exploring activities"; In the

Proceedings of Workshop Current Trends and Applications of Artificial Intelligence in Education at the 4th World Congress on Expert Systems, Mexico City, Mexico, (1998), 41-48

[Khan 2001] Khan, B.H.: "A framework for Web-based learning"; Web-based training, (2001), 75–98

[Kinshuk, Lin, and Patel 2005] Kinshuk, Lin, T., Patel, A.: "Supporting the Mobility and the Adaptivity of Knowledge Objects by Cognitive Trait Model"; In Spector J. M., Ohrazda C., Van Schaack A., and Wiley D. (Eds.), Innovations in instructional technology: Essays in honor of M. David Merrill, Mahwah, NJ: Lawrence Erlbaum, (2005), 29–41

[Kyllonen, and Shute 1989] Kyllonen, P.C., Shute, V.J.: "Learning indicators from a taxonomy of learning skills"; In Ackerman P. L., Sternberg R. J., and Glaser R. (Eds.) Learning and individual differences, New York: Freeman, (1988), 117-163

[Latham, Crockett, and McLean 2014] Latham, A., Crockett, K., McLean, D.: "An adaptation algorithm for an intelligent natural language tutoring system"; Computers & Education, 71, (2014), 97–110. doi:10.1016/j.compedu.2013.09.014

[Limongelli, Sciarrone, Temperini, and Vaste 2011] Limongelli, C., Sciarrone, F., Temperini, M., Vaste, G.: "The Lecomps5 framework for personalized web-based learning: A teacher's satisfaction perspective"; Computers in Human Behavior, 27, 4 (2011), 1310–1320

[Lin 2007] Lin, T.: "Cognitive Trait Model for Adaptive Learning Environments"; PhD Thesis, Massey University, Palmerston North, New Zealand, (2007)

[Macías, and Castells 2001] Macías, J., Castells, P.: "An authoring tool for building adaptive learning guidance systems on the web"; Active Media Technology, (2001), 268–278

[Macías, and Castells 2002] Macías, J., Castells, P.: "Interactive Design of Adaptive Courses"; Computers and Education, (2002), 235–242

[Magoulas, Papanikolaou, and Grigoriadou 2003] Magoulas, G.D., Papanikolaou, Y., Grigoriadou, M.: "Adaptive web based learning: accommodating individual differences through system's adaptation"; British Journal of Educational Technology, 34, 4 (2003), 511–527

[Márquez, Ortega, Gonzalez-Abril, and Velasco 2008] Márquez, J.M., Ortega, J.A., Gonzalez-Abril, L., Velasco, F.: "Creating adaptive learning paths using Ant Colony Optimization and Bayesian Networks"; In the Proceedings of the IEEE World Congress on Computational Intelligence IJCNN 2008, (2008), 3834–3839

[McClosky 1971] McClosky, M.G.: "Teaching Strategies and Classroom Realities."; First Edition edition. Englewood Cliffs, NJ: Prentice Hall, January (1971)

[Melis, Goguadze, Libbrecht, and Ullrich 2009] Melis, E., Goguadze, G., Libbrecht, P., Ullrich, C.: "Activemath - a learning platform with semantic web features"; In Dicheva, D., Mizoguchi, R. and Greer, J. (Eds.) Semantic Web Technologies for e-Learning, IOS Press, The Future of Learning, (2009), 159 - 177

[Mendez, Ramirez, and Luna 2005] Mendez, N.D.., Ramirez, C.J., Luna, J.A..: "IA planning for automatic generation of customized virtual courses"; In Proceedings of ECAI Frontiers In Artificial Intelligence And Applications, Vol. 117, Valencia, Spain, IOS Press (2005), 138–147

[Milosavljevic 1997] Milosavljevic, M.: "Augmenting the User's Knowledge via Comparison"; In Jameson A., Paris C., and Tasso C. (Eds.), User Modeling, International Centre for Mechanical Sciences, 383, Springer Vienna (1997), 119–130

[Mödritscher 2008] Mödritscher, F.: "Adaptive E-Learning Environments: Theory, Practice, and Experience"; Verlag Dr. Müller (2008)

[Mödritscher, Garcia-Barrios, and Gütl 2004] Mödritscher, F., Garcia-Barrios, V.M., Gütl, C.: "The Past, the Present and the Future of adaptive E-Learning: An Approach within the Scope of the Research Project AdeLE"; In the Proceedings of the International Conference on Interactive Computer Aided Learning ICL2004, (2004)

[Mohan, Greer, and McCalla 2003] Mohan, P., Greer, J., McCalla, G.: "Instructional planning with learning objects"; Knowledge Representation and Automated Reasoning for E-Learning Systems, (2003), 52–58

[Muñoz-Merino, Fernández Molina, Muñoz-Organero, and Delgado Kloos 2012] Muñoz-Merino, P.J., Fernández Molina, M., Muñoz-Organero, M., Delgado Kloos, C.: "An adaptive and innovative question-driven competition-based intelligent tutoring system for learning"; Expert Systems with Applications, 39, 8 (2012), 6932–6948

[O Keeffe, Brady, Conlan, and Wade 2006] O Keeffe, I., Brady, A., Conlan, O., Wade, V.: "Just-in-time generation of pedagogically sound, context sensitive personalized learning experiences"; International Journal on e-Learning, 5, 1 (2006), 113-127

[Ohlsson 1987] Ohlsson, S.: "Some principles of intelligent tutoring"; In Lawler, R.W., Yazdani, M. Ablex (Eds.) Artificial Intelligence and Education, Vol. 1, Norwood, NJ (1987), 203–237

[Oppermann, and Rasher 1997] Oppermann, R., Rasher, R.: "Adaptability and adaptivity in learning systems"; Knowledge Transfer, 2, (1997), 173–179

[Özyurt, Özyurt, and Baki 2013] Özyurt, Ö., Özyurt, H., Baki, A.: "Design and development of an innovative individualized adaptive and intelligent e-learning system for teaching–learning of probability unit: Details of UZWEBMAT"; , Expert Systems with Applications, 40, 8 (2013), 2914–2940

[Özyurt, Özyurt, Baki, Güven, and Karal 2012] Özyurt, Ö., Özyurt, H., Baki, A., Güven, B., Karal, H.: "Evaluation of an adaptive and intelligent educational hypermedia for enhanced individual learning of mathematics: A qualitative study"; , Expert Systems with Applications, 39, 15 (2012), 12092–12104

[Papanikolaou, Grigoriadou, Kornilakis, and Magoulas 2003] Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H., Magoulas, G.D.: "Personalizing the Interaction in a Web-based Educational Hypermedia System: the case of INSPIRE"; User Modeling and User-Adapted Interaction, 13, 3 (2003), 213–267

[Paramythis, and Loidl-Reisinger 2003] Paramythis, A., Loidl-Reisinger, S.: "Adaptive learning environments and e-learning standards"; In the Proceedings of the 2nd European Conference on e-Learning, Glasgow, Scotland (2003), 369-377

[Park, and Lee 2008] Park, O., Lee, J.: "Adaptive instructional systems"; In Spector, J.M., Driscoll, M.P., Van Merrienboer, J., and Merril , M.D. (Eds.), Handbook of Research on Educational Communications and Technology, 3rd ed. (2008), 469-484

[Pérez, Lopistéguy, Gutiérrez, and Usandizaga 1995] Pérez, T., Lopistéguy, P., Gutiérrez, J., Usandizaga, I.: "HyperTutor: From hypermedia to intelligent adaptive hypermedia"; In ED-MEDIA'95, World conference on educational multimedia and hypermedia, (1995), 529–534

[Priewe, and Dullien 2010] Priewe, J., Dullien, S.: "Teaching tools : a digital handbook for academic lectures"; (2010), retrieved on 09/09/2015 from http://vi.unctad.org/teaching-tools-a-digital-handbook-for-academic-lecturers

[Pukkhem, Evens, and Vatanawood 2006] Pukkhem, N., Evens, M.W., Vatanawood, W.: "The Concept Path Combination Model for Supporting a Personalized Learning Path in Adaptive

Educational Systems"; In the Proceedings of the 2006 International Conference on e-Learning, e-Business, Enterprise Information Systems, e-Government, and Outsourcing (EEE'06), (2006), 10-16

[Reiser 1987] Reiser, R.A.: "Instructional technology: A history"; In Gagne R.M. (Ed.), Instructional technology: Foundations. Hillsdale, NJ: Lawrence Erlbaum Associates, (1987), 11–48

[Riad, El-Minir, and El-Ghareeb 2009] Riad, A.M., El-Minir, H.K., El-Ghareeb, H.A.: "Review of e-Learning Systems Convergence from Traditional Systems to Services based Adaptive and Intelligent Systems"; Journal of Convergence Information Technology, 4, 2 (2009), 108-131

[Rickel 1989] Rickel, J.W.: "Intelligent computer-aided instruction: A survey organized around system components"; Systems, Man and Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics, 19, 1 (1989), 40–57

[Sanrach, and Grandbastien 2000] Sanrach, C., Grandbastien, M.: "ECSAIWeb: A Web-Based Authoring System to Create Adaptive Learning Systems"; In Brusilovsky, P., Stock, O., and Strapparava C. (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems, Berlin, Heidelberg: Springer Berlin Heidelberg (2000), 214–226

[Santos, Anido, Llamas, and Rodríguez 2002] Santos, J., Anido, L., Llamas, M., Rodríguez, J.: "On the Application of the semantic Web Concepts to Adaptive E-learning"; EurAsia-ICT 2002: Information and Communication Technology, (2002), 536–543

[Self 1974] Self, J.A.: "Student models in computer-aided instruction"; International Journal of Man-Machine Studies, 6, 2 (1974), 261–276

[Shute, and Psotka 1996] Shute, V.J., Psotka, J.: "Intelligent Tutoring Systems: Past, Present and Future"; In Jonassen, D. (Ed.) Handbook of Research on Educational Communications and Technology, New York, NY: Macmillan (1996), 570-600

[Shute, and Towle 2003] Shute, V., Towle, B.: "Adaptive e-learning"; Educational Psychologist, 38, 2 (2003), 105–114

[Shute, and Zapata-Rivera 2008] Shute, V.J., Zapata-Rivera, D.: "Adaptive technologies"; Handbook of research on educational communications and technology, (2008), 277–294

[Shute, Lajoie, and Gluck 2000] Shute, V.J., Lajoie, S.P., Gluck, K.A.: "Individualized and group approaches to training"; Training and retraining: A handbook for business, industry, government, and the military, (2000), 171–207

[Sleeman, and Brown 1982] Sleeman, D., Brown, J.S.: "Introduction: Intelligent Tutoring Systems: An Overview"; Intelligent Tutoring Systems, Sleeman, D.H., Brown, J.S. Academic Press, Burlington, MA (1982), 1–11

[Snow, and Swanson 1992] Snow, R.E., Swanson, J.: "Instructional psychology: Aptitude, adaptation, and assessment"; Annual Review of Psychology, 43, 1 (1992), 583–626

[Specht, and Oppermann 1998] Specht, M., Oppermann, R.: "ACE-adaptive courseware environment"; The New Review of Hypermedia and Multimedia, 4, 1 (1998), 141–161

[Specht, Kravcik, Pesin, and Klemke 2001] Specht, M., Kravcik, M., Pesin, L., Klemke, R.: "Authoring adaptive educational hypermedia in WINDS"; In the Proceedings of ABIS2001, Dortmund, Germany, (2001), 1–8

[Specht, Weber, Heitmeyer, and Schöch 1997] Specht, M., Weber, G., Heitmeyer, S., Schöch, V.: "AST: Adaptive WWW-courseware for statistics"; In the Proceedings of Workshop'

Adaptive Systems and User Modeling on the World Wide Web' at 6th International Conference on User Modeling, UM97, Chia Laguna, Sardinia, Italy, (1997), 91–95

[Stankov, Grubišić, and Žitko 2004] Stankov, S., Grubišić, A., Žitko, B.: "E-learning paradigm & Intelligent tutoring systems"; Annual 2004 of the Croatian Academy of Engineering, (2004), 21–31

[Stones, and Morris 1972] Stones, E., Morris, S.: "Teaching Practice: Problems and Perspectives"; London: Routledge, (1972)

[Suppes 1967] Suppes, P.: "Some theoretical models for mathematics learning"; Journal of Research & Development in Education, 1, 1 (1967), 5–22

[Uhr 1969] Uhr, L.: "Teaching machine programs that generate problems as a function of interaction with students"; In the Proceedings of the 24th ACM National Conference, (1969), 125–134

[Ullrich 2008] Ullrich, C.: "Pedagogically founded courseware generation for web-based learning: an HTN-planning-based approach implemented in PAIGOS"; Lecture Notes In Computer Science, Vol. 5260, (2008), 111-167

[Urban-Lurain 1996] Urban-Lurain, M.: "Intelligent tutoring systems: An historic review in the context of the development of artificial intelligence and educational psychology", (1996), Retrieved on 09/09/2015 from http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm

[Van Marcke 1990] Van Marcke, K.: "A generic tutoring environment"; The European Conference on Artificial Intelligence, (1990), 655–660

[Van Marcke 1992] Van Marcke, K.: "Instructional expertise"; In the Proceedings of the Second International Conference on Intelligent Tutoring Systems, Springer (1992), 234–243

[Van Marcke 1998] Van Marcke, K.: "GTE: An epistemological approach to instructional modelling"; Instructional Science, 26, 3 (1998), 147–191

[Vassileva 1992] Vassileva, J.: "Dynamic CAL-courseware generation within an ITS-shell architecture"; In Lecture Notes in Computer Science, Vol. 602, Proceedings of the International Conference on Computer Assisted Learning ICCAL, (1992), 581–591

[Vassileva 1995] Vassileva, J.: "Dynamic courseware generation: at the cross point of CAL, ITS and authoring"; In the Proceedings of ICCE, (1995), 290–297

[Vassileva 1998] Vassileva, J.: "DCG+ GTE: Dynamic courseware generation with teaching expertise"; Instructional Science, 26, 3 (1998), 317–332

[Vassileva, and Wasson 1996] Vassileva, J., Wasson, B.: "Instructional planning approaches: From tutoring towards free learning"; In the Proceedings of EuroAIED96, (1996), 1–8

[Vesin, Ivanović, Klašnja-Milićević, and Budimac 2012] Vesin, B., Ivanović, M., Klašnja-Milićević, A., Budimac, Z.: "Protus 2.0: Ontology-based semantic recommendation in programming tutoring system"; Expert Systems with Applications, 39, 15 (2012), 12229–12246

[Wang, and Lindvall 1984] Wang, Lindvall, C.M.: "Individual Differences and School Learning Environments"; Review of Research in Education, 11, (1984), 161–225

[Wang, Wang, and Lin 2010] Wang, C., Wang, D.-Z., Lin, J.-L.: "ADAM: An adaptive multimedia content description mechanism and its application in web-based learning"; Expert Systems with Applications, 37, 12 (2010), 8639–8649

[Weber 1996] Weber, G.: "Episodic Learner Modeling"; Cognitive Science, 20, 2 (1996), 195–236

[Weber, and Moellenberg 1995] Weber, G., Moellenberg, A.: "ELM-Programming-Environment: A Tutoring System for LISP Beginners"; In Wender, K. F., Schmalhofer, F., and Böcker, H.-D. (Eds.) Cognition and Computer Programming, Norwood, NJ: Ablex (1995), 373–408

[Weber, Kuhl, and Weibelzahl 2002] Weber, G., Kuhl, H.C., Weibelzahl, S.: "Developing adaptive internet based courses with the authoring system NetCoach"; Hypermedia: Openness, Structural Awareness, and Adaptivity, (2002), 222–223

[Wenger 1987] Wenger, E.: "Artificial Intelligence and Tutoring Systems"; Morgan Kaufmann Publishers, Inc., California, USA (1987)

[Wiley 2000] Wiley, D.A.: "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy"; The instructional use of learning objects, 2830, 435 (2000), 1–35

[Woolf 1992] Woolf, B.: "AI in Education"; In Shapiro, S. (Ed.), Encyclopedia of Artificial Intelligence, John Wiley & Sons, Inc., New York, (1992), 434-444.

[Zhu, and Cao 2008] Zhu, F., Cao, J.: "Learning activity sequencing in personalized education system"; Wuhan University Journal of Natural Sciences, 13, 4 (2008), 461–465