

## ADAPTIVE E-LEARNING CONTENT DESIGN AND DELIVERY BASED ON LEARNING STYLES AND KNOWLEDGE LEVEL\*

Dessislava Vassileva

**ABSTRACT.** The possibility of adaptive learning content delivery in e-learning systems is one of the important factors for highly improving their quality. Therefore, the application field of adaptive e-learning is relevant and significant. This article presents the main results of a PhD thesis examining various aspects of this area. The aim of the dissertation is to propose a model and a platform architecture of an adaptive e-learning system and a corresponding prototype to be designed, implemented and tested in experimental conditions. On one hand, the developed prototype will assist a learner in accessing and using learning resources which are adapted according to his/her personal characteristics (in this case his/her learning style and level of knowledge). On the other hand, it will facilitate the author of the learning content and course instructor in the creation of appropriate learning objects and applying them to the suitable pedagogical strategies.

---

*ACM Computing Classification System* (1998): K.3.1, K.3.2.

*Key words:* adaptive e-learning systems, models of adaptive hypermedia systems, authoring tool, instructor tool, adaptive engine, narrative storyboard courseware.

\*The work reported in this paper is supported by the ADOPTA project funded by the Bulgarian National Science Fund under agreement no. D002/155.

This article presents the principal results of the Ph.D. thesis Adaptive software systems for e-learning by Dessislava Vassileva (Faculty of Mathematics and Informatics at Sofia University), successfully defended at the Specialized Academic Council of FMI on 05 July, 2011.

**1. Introduction.** In years of increasingly fast changing technology, lifelong learning becomes a necessity. Its requirements can be satisfied most adequately way by e-learning methods. The reason for this is that, unlike the classical classroom approach, e-learning provides an opportunity to conduct asynchronous teaching. Asynchronous learning is learning where students and teachers are not bound to a specific place and time [31]. On the other hand, e-learning has some disadvantages compared to traditional learning. Some of them are the lower efficiency of the learning process due to lack of direct contact and the impossibility of applying a personal approach of teaching to each student. Adaptive e-learning systems (AES) aim to overcome the last problem.

In the last fifteen years, creation and delivery of modules for adaptive learning content has become an important part in the design of advanced platforms for e-learning [33]. In order to be effective, a learning process should not only rely on developing training materials and placing them in an accessible place on a website as in traditional e-learning. For an effective training process insufficient learning material is accessible via Internet, which fuels the invasion of adaptive e-learning. Knowledge materials should be tailored to various characteristics of the learner such as specific goals, preferences, knowledge, and learning style, so that appropriate teaching strategies can be used [46]]. The aim of adaptive e-learning is precisely this, and therefore the interest in this area and its popularity continue growing.

As adaptive software systems for e-learning, AES cover basic requirements of both e-learning systems and adaptive software and follow their basic principles. The main trends identified in systems for designing and delivering educational content for e-learning are as follows:

- learning content is composed of *learning objects*. Learning objects represent a new paradigm for creating teaching materials. In the old paradigm, training content is organized into lessons and courses that meet predetermined objectives of the course or lesson. In the new paradigm the curriculum is divided into smaller, autonomous units that can be used both separately and combined (statically or dynamically) with others. Among the properties that a learning object must possess one may find modularity, interoperability, reusability, and accessibility.
- educational content and its learning objects must be described by *metadata*. Metadata provide information about a resource. They describe its context, properties, purpose and characteristics. Metadata can describe an object independently of its level of aggregation—combination of resources, a resource or component of a larger object. The aim of using metadata is

to improve and facilitate extraction of information. Furthermore, they can support interoperability, the integration of an object and its identification [44]. The most popular standards for metadata used in e-learning are these two: Dublin Core [19] and IEEE Learning Object Metadata (LOM) [23].

- packaging of learning content must be consistent with an established appropriate standard. The main objective of such a standard is to specify a fixed data structure used for training and communication protocols. Its efforts are aimed at facilitating reuse of learning objects and giving an opportunity for interoperability between systems for e-learning [5, 45].

The main characteristics that an adaptive software system must have are the following:

- its behavior should be adaptable according to changes in the working environment or parts/components of the adaptive system itself;
- constituent modules should be developed to maximize compliance with the changes in external environments;
- tools should be provided for monitoring and controlling its work, means for changing parameters and using a closed circle of actions to improve productivity and to optimize the interaction with users.

Concepts' adaptation and adaptability are very often used as interchangeable terms, but their principle of action is different. Systems that allow the user to manually change some parameters and thus to adapt their behavior are called adaptable [42]. Systems that are adapted automatically based on actions of the user and on independent conclusions about the user's needs are called adaptive [42]. The adaptability and adaptation of a system mainly refer to environment (adaptability or adaptation to the hardware and/or software platform respectively) or the user (reflecting the user's specific requirements). This article will focus on the last one.

AES are adaptive in the context of e-learning. They are focused mainly on adaptation of content and its presentation. Adaptability in e-learning systems consists in the possibility of:

- meeting the user's needs and preferences;
- dependence on the user's behavior;
- using results obtained by a user for further adaptation.

These data (preferences, behavior, performance) are collected into a so-called user model (UM). According to [52] "An adaptive e-learning system is an

interactive system that personalizes and adapts e-learning content, pedagogical models, and interactions between participants in the environment to meet the individual needs and preferences of users if and when they arise”.

There are different types of AES such as macro-adaptive learning systems, computer-supported collaborative learning, intelligent tutoring systems (ITS) [12] and adaptive hypermedia systems (AHS). The most widespread of these are the last two—ITS and AHS. ITS are systems that use additional techniques from artificial intelligence. The advantage of these systems over others is that they automatically customize the learning process. Their main purpose is to simulate different aspects of teaching. Inspired by the ITS, in the early 1990s researchers developed AHS. They try to combine adaptive learning systems and hypermedia-based systems. AHS adapt to user characteristics, and for this they usually employ a user model. The instructor has to choose different solutions for the educational content, its navigation and its presentation, and to define various adaptive criteria.

According to Brusilovsky [13], AHS is any system using hypertext and/or hypermedia which is focused on the student and applies his/her model to adapt various aspects of the system to the user. In terms of definition ITS and AHS overlap, as they have the same purpose, but different options for achieving it, and these options can be combined and can be reproducible between both systems.

AHS are used for implementation of educational systems, e-commerce applications, information systems. They can be useful anywhere where hypertext and hypermedia are used. The most popular adaptive hypermedia systems are web-based systems [2]. The article presents a Web based AHS called ADOPTA (ADaptive technOlogy-enhanced Platform for eduTAinment) for content delivery, adaptive to individual learner style and performance accessed by measuring the learner’s knowledge level [55]. After providing a short overview of other work related to this research, the paper described the conceptual system model, proposed a methodology for mastering and delivery of adaptive content, a software architecture of the adaptive platform and, finally, some of the practical experiments performed by using the platform for adaptive e-learning provided to B.Sc. students in software engineering at Sofia University, Bulgaria.

**2. Related work.** The main factor for the popularization of AES is that they enable use of standardized models for data sharing and reuse of learning objects. They should include standardized data models to ensure interoperability of content and services. Usually, AES use several established models of data. These models are based on different categories of adaptation of the learning process

[43]. We will discuss briefly each of them and then present some of the most used reference models for creating AES.

**2.1. User model.** The *user model* describes the data system for the current state of the student's knowledge, preferences, etc. It is often divided into two parts:

- common part—refers to the various characteristics of the learner such as preferred learning style, cultural background, preferences and many others, depending on the type of system. Some of these characteristics can be determined by the student directly or indirectly—for example, by completing the test;
- specific part—here the learner's knowledge and progress are presented. Usually the *user model* has no direct control over this part except when an initial setup is necessary. This part stores all actions of the learner as visited pages, completed tasks, and answers to tests questions. These actions can be tracked and used to reduce or increase the students' level of knowledge for any of the concepts involved.

Sometimes for this model researchers use other terms such as learner model or student model, but these refer to the same concept. One of the first proposals for learner model is made by Wenger [57] and includes all the performance and knowledge of the user which can affect the learning and progress of students. However, it is impossible to collect all these data, because this would take a long time and mostly because it is not clear how they will be used in the adaptation process.

In some studies [16, 35] researchers express the opinion that the way of thinking, goals, interests and knowledge levels are the main attributes that determine the definition of different types of learners. It is believed that these four types of data are highly correlated with the cognitive style of each learner [39]. Moreover they may determine students' learning style.

Brusilovsky (2002) proposes using seven attributes in AES as shown in Fig. 1, which concern a given learning style. This appears very important for AES design.

**2.2. Domain model.** The *domain model* describes the content of information in an e-learning system. It includes information on conceptual level, describing the structure of a domain, and individual fragments or pages. Some authors use only the first part of this model—structural level [14]. The first part of the model—the structure—is usually modeled using concepts, i.e., abstract representations of information units of a domain. As well, the domain model

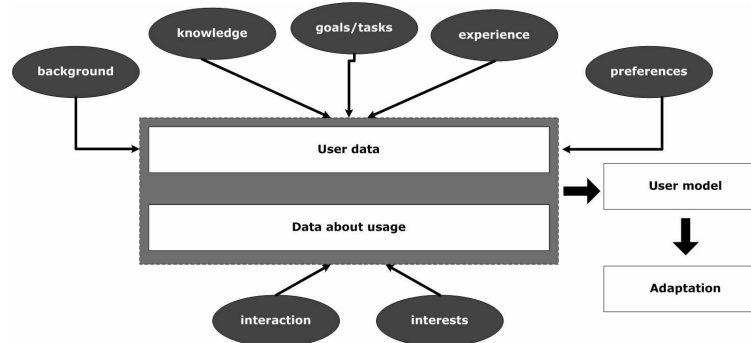


Fig. 1. Structure of the user model according [15]

reflects the relationship between concepts (preconditions, results, and others depending on the system) and even the definition of how information fragments (units of explanatory text, images, examples and other content) are associated to concepts and how they are stored. This is the highest level of organization of content, especially of logical links between nodes in the network of content concepts.

The second part of the model—the content—consists of text documents, images or other content and description of how to store and invoke them (some authors include content directly, while others are simply reluctant to rely on the repositories).

The simplest type of structure of this model is only a set of concepts for a subject area [13, 14], which contains the real content of elements pertaining directly to them without any special relationship. A more advanced structure of the domain model is a network which contains nodes corresponding to concepts of an area and links reflecting several kinds of relationships between concepts. This network presents the structure of a subject area covered by the hypermedia systems.

**2.3. Adaptation model.** The *adaptation model* specifies the logic used in the implementation of adaptation decisions. Usually it is not a separate part of the system and this is a problem for modern AES. According to the most widespread practice, adaptation instructions are to be entered and stored with the content. The most used standard SCORM supports setting of an adaptive sequence of learning activities, but it again assumes that the logic will be an integral part of the personalized content and unit and the logic only provides definitions for the external behavior rather than a formulation of internal representation.

The most popular method for presenting the logic uses adaptation rules.

These rules are expressed with the classical construction *if-then*, where in the *if* section conditions are listed, and in *then*, actions. The form of conditions and actions depends entirely on a given system. Conditions can range from simple to complex expressions. Some actions can be performed not only by satisfying certain conditions, but by executing another action, such as visiting a site.

Relatively old systems like HyperTutor [40] and SYPROS [27] use a set of pedagogical rules to decide which concepts and units must be visible at any given moment and which are not. In old AES, rules are included in the program code, while modern systems usually store them together with the content. For instance, AHA! [21] stores adaptive rules in the same file with the concept and definition of the page. This limits rules to being applied only on a set of concepts and prevents them from working within the scope of a course.

We could distinguish two completely independent sets of rules:

- rules modeling the learner—they reflect activities and results of the learner in his/her model (usually as a set of values);
- rules adapting the content—on them depends the preparation of content adapted to the individual learner. They usually do not use actions of learners as input but already processed data generated by the rules of the first group.

**2.4. Reference models of AES.** Modern AES follow some of existing well-proven reference models such as:

- Dexter Hypertext Reference Model [29]—the purpose of this model is to provide a methodological basis for comparison of systems and to develop standards for the exchange and interoperability. It divides an AHS into three layers:
  - Storage layer—describes network of nodes and links, which is the essence of hypertext documents;
  - Run-time layer—describes mechanisms to support student interaction with hypertext;
  - Within-component layer—it contains content objects which are internal and specific to the implementations.

The focus of the model is on the first layer and on mechanisms specifying presentation and linking of individual nodes.

- AHAM Reference model [20]—improves the above model. This model en-

ables support of adaptive hypermedia applications by separating the first layer (data store) of the Dexter model of following three submodels—domain model, user model and adaptation model (Fig. 2). More, it adds options to support adaptation based on the user model and independent of the duration of the session. The purpose of the AHAM Reference model is to describe the functionality of an AHS.

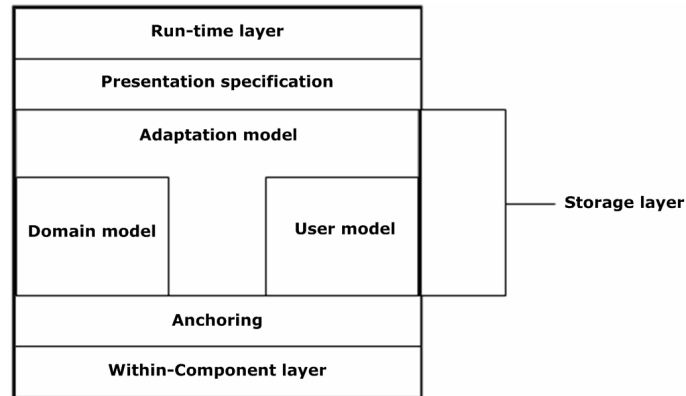


Fig. 2. AHAM Reference model [20]

- Goldsmiths Adaptive Hypermedia Model (GAHM) [41]—abstract model developed by Goldsmiths College, University of London. Provides a formal approach for modeling personalized, adaptive hypertext based systems. The model contains three groups of functions:
  - H-Region—models non-personalized hypermedia based on interaction. In this model, pages are presented as a formal specification. The semantics of this specification is represented by reference to abstract machine operations, which are formalized;
  - P-Region—presents the adaptation of learners to the content of a hypermedia;
  - A-Region—models the adaptation of a system to the content of a hypermedia.

The main drawbacks of the above models consist in their support of adaptability only to students' knowledge. They do not support adaptability to the learner's objectives and learning styles, which has established itself in recent years as one of the most effective concepts in modern AHS. Furthermore, exist-



ing models of AHS insert specific teaching strategies in specific content, which hampers the implementation of interoperability between different e-learning systems [45]. Also this prevents the use of the same pedagogic strategies for different learning materials.

Another restriction of the examined reference models is maintenance of metadata for educational content that is provided solely by AHAM and describes only relations between concepts. In this model, there are no metadata that describe the content itself (or fragments, which are equivalent to learning objects). This leads to a lack of opportunity for effective search and exchange of learning objects, and maintenance standards for description of metadata such as LOM.

**2.5. Existing implementations of AES.** In recent years the field of AES marks significant progress with the rise of many new implemented applications that reflect the latest trends in this area and the improvement of old ones. Among the most widely used and well documented AES are the following:

- Adaptive Course Generation System (ACGs) [56]—an authoring system for adaptive course, designed and created by the Faculty of Information Technology, College of Technology, Vietnam National University, Hanoi [56]. It constitutes a basic AES offering web courses with adaptive navigation. ACGs is implemented using agent-based technology via the system Bee-gent (Bonding and Encapsulation Enhancement Agent) [34]. Agents are used for monitoring of students, updating the domain model, delivering and formatting of curriculum, preparation of tests and evaluation of results. The focus of the adaptation is on goals and knowledge. ACGs has the opportunity to read SCORM compliant training packages but it cannot export them;
- AHA! [22, 21]—here the user model is presented as a set of knowledge reflecting different concepts. The domain model consists of fragments, pages and concepts. Pages are represented as XML files consisting of fragments that can be included conditionally and contain hypertext links and information about different concepts and their relations. Concepts are defined only in the pages and there is no comprehensive system reflecting the relationship between them. Each page is created by the author of the learning content. The presentation of the content page is determined at runtime, so that certain fragments are selected depending on the condition with which they are associated. The focus of the adaptation is on knowledge, interests and goals. There are tools developed specifically for AHA!, enabling SCORM courses to be integrated into the system [49]. The system interface is simple and it is not well developed;

- PERSO [17]—based on the recognition and processing of natural language. The system uses sophisticated techniques to understand the information and requirements entered by the students, and create learning content on this basis. PERSO used two approaches. One of them serves to represent semantic space and evaluation of semantic similarity between the student and the correct answers (given by the trainer) by latent semantic analysis (LSA). The other approach is based on previous decisions on similar problems and generates a new solution using case-based reasoning (CBR). It is used by the system in order to determine which concept to display to the student based on his/her knowledge. Knowledge is represented by semantic networks—graphs in which vertices represent parts of the knowledge, and arcs, connections between them. PERSO consists of five components: curriculum, student model, analyzer, CBR system and planning module (Planer). The focus of the adaptation is on knowledge. PERSO does not support standards and processes and it analyzes natural language and therefore this system has limited potential for internationalization.

**2.6. Conclusions.** According to the overview of AES given above, we can conclude that some fundamental shortcomings of existing models of AHS and AES are related to lack of maintenance of adaptation to learning style and the insertion of specific teaching strategies into learning content.

Specifying a pedagogical strategy in learning content hampers the implementation of interoperability between different e-learning systems and prevents the use of the same teaching strategies for different training materials. On the other hand, use of learning styles in a variety of e-learning systems gives very good results [4, 37, 25], which proves that it can be one of the most effective concepts in modern AHS. Therefore it is important for adaptive courses to comply with this concept.

In older AHS, rules are included into the program code, while in contemporary systems they are usually stored together with content, however, this is not an optimal solution. In order to be effectively maintained and reused, both content and rules should be stored separately from content.

Another problem of AES which support standards for packaging of learning objects is that authors have to devote too much time on their description (LOM contains 83 fields to be filled). It demotivates teachers from using a similar system or describing learning objects and leads to lack of interest among most of them.

The systems discussed above have no friendly graphical interface that is able to track how the course will be presented to different learners. Most systems

provide a scheme of relationships between concepts and learning objects but it is not clear how it will conduct the training process.

Therefore, it can be concluded that is essential to create a new principal model of adaptive e-learning system, which should eliminate the disadvantages stated above. Next, the model should serve as a base for construction of a software platform which will be used for practical experiments with e-learning adaptive to learning styles and knowledge level.

**3. Conceptual model of AHS for e-learning.** In view of the shortcomings of the AHAM model, this chapter will offer a new conceptual model of AHS which eliminates them. It follows an approach based on metadata and offering a clear separation of the sequence of learning activities from learning content and the adaptation engine. Fig. 3 presents its so-called triangular conceptual model (TCM) [54]. As shown in the figure, the new conceptual model has common features with the AHAM model [20] but improves it by dividing AHAM into three main independent submodels, namely learner model, domain model and adaptation model. They support characteristics missing in other models of AHS such as learning styles, ontological structure of learning activities, their metadata and adaptive rules. Besides these three main submodels, the new model includes an adaptation engine, which is responsible for adaptive content delivery to students and adaptive control of the learning process.

Thereby we obtain a new hierarchically organized model for building AHS to manage the learning process. This hierarchical model consists of two levels.

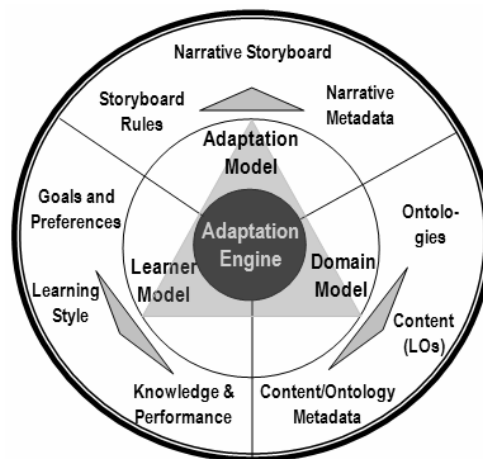


Fig. 3. Structure of TCM [6]

On the first level the model is based on precise separation of the three main submodels, while on the second level each of these submodels is divided into three other submodels [54]. All submodels should be defined using XML schemas that represent characteristics of the learner, content and teaching strategies. These schemes should be designed and used concurrently for interoperability and consistency with other e-learning systems.

In contrast to the AHAM, TCM supports different learning styles such as these of Honey and Mumford, which include the following categories of students: theorist, activist, pragmatist and reflector [32]. The model includes metadata about learning content, adaptation rules and rules for packaging of content according to standard SCORM [47]. These additional features make the process more efficient and provide opportunities for more reliable interoperability.

**3.1. Learner model.** The purpose of the learner model is to store information about students. This information is structured in the following three submodels:

- *Goals and Preferences*—stores information which indicates what courses in the system a particular learner wants to visit, what are his/her preferences such as font type, size, color and other parameters associated with the interface. Furthermore, the preferred courses can be ranked by priority.
- *Learning style*—it contains information about the specific way of learning of each student and the teaching approach appropriate for him/her. These data are used by the adaptive machine to provide the most complete and effective training.
- *Knowledge and performance*—stores results of tests, essays, projects, tasks, and more. These results are indicators of the student's progress and hence of the effectiveness of the learning process. Furthermore, this model contains the learner's knowledge and knowledge obtained from other sources outside the AHS and declared by the learner.

Unlike other approaches, in the learner model (LM) goals and preferences are separated from knowledge and results. The first of the two submodels (goals and preferences) is static in terms of participation of the adaptive engine in the process of defining goals and preferences. The second model (knowledge and results) is very dynamic and takes part in the whole learning process and its monitoring. The third submodel (style of learning) takes a central place in LM. Depending on the style model, characteristics of learners may have varying degrees in the relevant learning styles. These learning styles may be as defined by the model of Honey and Mumford [32]—activist, theorist, pragmatist, reflec-

tor, or by the VARK model of Fleming's learning styles—visual, auditory and kinesthetical, etc. Learning style is included as an individual submodel and can be used to select the best content for a student possessing a combination of learning styles. Such a combination of learning styles forms a polymorphous learning style and is used because in most cases the students pertain to different degrees to several of them and never to only one of these styles (Fig. 4). While learning style can be defined at the beginning of the training personally for each student by appropriate preliminary tests, other tests (assessment) must be solved in the process of learning to evaluate past or acquired knowledge.

To describe the polymorphic learner profiles, conceptual characteristic properties will be defined, such as the characteristics of learning styles, training preferences and psychological profiles. Each of the conceptual characteristics describing the learner has a weight factor  $W_{si}$  (set to zero or any integer or a percentage between 0% and 100% inclusive for each style). This factor determines the level of importance or availability of a concept (character) in the learner model, as shown in the learning style in Fig. 4. Thus conceptual characteristics which are not relevant or not available receive zero weight. Each user is to a certain extent identified with different concepts. This can be expressed in numerical form as a percentage. This set of values forms an N-dimensional vector characterizing the user.

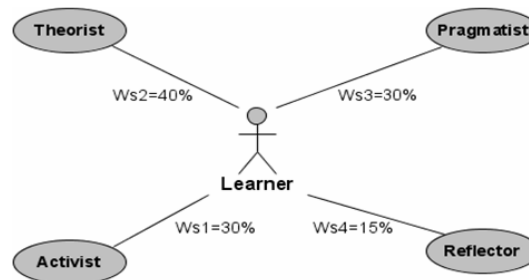


Fig. 4. Example of a polymorphic profile of a learning style

It is important to note that the learner model is not oriented to one of the existing models of learning styles (such as the models defined by Honey and Mumford, Gregoric-Mind, Dunn and Dunn, etc. [32]) and it can be used for each learning style of these models.

**3.2. Domain model.** The domain model is used for storage, organization and description of the learning content. It is composed of the following three parts:

- *Content*—in form of learning objects according to standard SCORM ([51]);
- *Ontologies*—the goal is for the learning content to be organized in a semantic way [9];
- *Metadata*—describes learning objects and ontologies according to the standards LOM [24] and OMV [30] respectively.

This model supports different types of educational content—not just standard narrative content (like lectures) but also educational activities such as tasks, projects, topic for writing an essay, test questions, educational games, and more. The ontology should be specified by the author of the course at an early stage, so that during the creation of educational materials a logical taxonomy for the knowledge area is formed (i.e., ontology of the subject area). Thus, learning objects are developed by the author and placed in the pages of the course by the instructor.

The purpose of defining the semantics of learning content during its creation is to facilitate stakeholders in reviewing and requiring it. Ontology may be represented not only by a tree—as in Protégé [38]—but also by a graph, which adds more semantic power and completeness and facilitates the presentation of multiple inheritance and references of one learning object to another. The domain model and in particular its submodel—*ontologies*, supports the following two types of relationships between nodes in ontological graphs:

- *is-a* relation—corresponds to typisation of learning objects in terms of their definition. This type of relationship is present as subtyping among classes in the paradigm of object-oriented programming.
- *has-a* relation—represents relationships between types such as aggregation, association and dependency.

**3.3. Adaptation model.** The adaptation model (AM) is responsible for the semantics of teaching strategies used in a course and, as well, for describing the logic of selection and delivery of learning activities/concepts. It includes the following submodels: *narrative metadata*, *narrative storyboard* and *storyboard rules*.

The *narrative storyboard* submodel supports a narrative graph, which contains working paths (WP)—different for each learning style. This graph consists of control points (CPs) and WPs connecting every two CPs. In a narrative graph, the instructor can create different WP for different learning styles as shown in fig. 5. Learning objects are located on pages of narrative content, which represent the nodes of the graph. Moreover, AM has to submit a scheme of rules that control

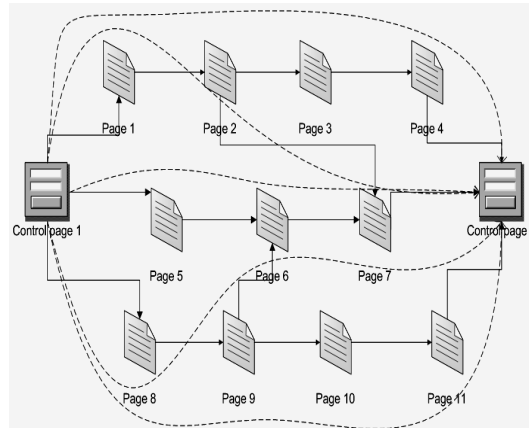


Fig. 5. Example for a narrative graph of a course

learning process. These rules determine the sequence of pages in a course depending on the input data in LM. The third submodel—narrative metadata—defines parameters of the rules such as a threshold of the test results for the transition to the next control point or return to the previous one, annotated links between pages, visibility of learning objects depending on their complexity, etc.

**3.4. Adaptation engine.** The delivery of pages with learning content to students is controlled by an adaptation engine (AE) by selecting the most suitable WP (using adaptive navigation) and content (with adaptive selection of content and annotated links), whose presentation is consistent with the corresponding profile of the learner. Instead of dynamically choosing a page (i.e., a node of the graph) with its contents, AE selects the most suitable WP of the graph for a student with a learning style on one hand, and knowledge and results on the other. For this purpose CPs are defined as nodes of the graph, where AE assesses students' knowledge/results and/or receives data from the learners themselves on the level of satisfaction according to his/her goals and preferences. When a student begins a new course, the adaptive engine locates the WP that is best for him/her in the graph for the corresponding course. The best path for the student is the one with the greatest weight. For a learner with given learning styles, the best path is calculated by the following formula:

$$(1.1) \quad \max_{(k)} \left\{ \frac{\sum_i W_{WPk}(c_i) * W_{c_i}(l)}{\|W_{WPk}(c)\| * \|W_c(l)\|} \right\}, \text{ where:}$$

- $k$  is the number of WP from current CP to next
- $c_i$  is the value of support of learning style  $i$
- $W_{WP_k}(c_i)$  is the weight of working path  $WP_k$  for learning style  $c_i$
- $W_{c_i}(l)$  is the level at which student  $l$  belongs to learning style  $c_i$
- $\|W_{WP_k}(c)\|$  is the length of the vector consisting of weights of working path  $WP_k$  for each learning style
- $\|Wc(l)\|$  is the length of the vector composed of the level at which student belongs to each of the learning styles.

If you present the weights of different WPs and the polymorphic learning style as vectors, the maximum scalar product between them divided by the product of their lengths will correspond to the cosine of the angle between these two vectors and therefore of the learner profile. Thus, the more similar the two vectors are, the more appropriate will the relevant WP be for that learning style. If there are several vectors of WPs with the same angle as the vector of polymorphic learning style, then the vector with greatest length will be selected, i.e., the vector with higher values for the learning styles.

The formula for updating the weight of WP, after solving test in CP  $k+1$ , is the following:

$$(1.2) \quad W_{WP_k}(c_i) = W'_{WP_k}(c_i) + \frac{W''_{WP_k}(c_i) + (R - P) * wc_i(l)}{N}, \text{ where:}$$

- $c_i$  is the value of support of learning style  $i$
- $WP_k$  is the corresponding WP from CP  $k$  to CP  $k+1$
- $W'_{WP_k}(c_i)$  is the weight of  $WP_k$  for  $c_i$  originally defined by the trainer
- $W''_{WP_k}(c_i)$  is the difference between the current weight and the weight initially set by the trainer for  $WP_k$  for  $c_i$
- $W_{WP_k}(c_i)$  is the new weight of  $WP_k$  for  $c_i$
- $R$  is the result of the test for learner  $l$  in CP  $k+1$
- $P$  is a parameter of correction. It can be equal to the pass threshold set for a test or have a greater value. It depends on the discretion of the teacher/instructor about the appropriateness of the WP for a learner with a given assessment result. The purpose of this parameter is the weight of the WP to be reduced when the test result is less than the expected one.
- $W_{c_i}(l)$  is the level at which learner  $l$  belongs to learning style  $c_i$



- $N$  is the number of students who passed through the path  $WP_k$  until now.

The adaptation engine can configure some parameters of the learning process such as:

- number of questions that appear in each CP
- WPs from which the adaptation engine chooses the most appropriate one
- level of complexity of learning objects that is to be displayed
- stopping adaptive navigation and structural adaptation

**3.5. Formalization.** The goal of this part is to present a formal description of an adaptation process based on TCM. There are some approaches which can be used for this description such as:

- Object Constraint Language [48]—used in the Munich Reference Model [36];
- descriptive language—used in the specification of GAHM;
- predicate logic—used in the Dexter Hypertext Reference Model ([28, 29]).

In this paper, for the formal model description of TCM, in addition to the above approaches, predicate logic [1] is chosen. It is an extension of propositional logic with separate symbols for predicates, subjects and quantifiers and uses a wholly unambiguous formal language, which enables clearer understanding of adaptive processes.

According to TCM, it could be assumed that an AES can be presented with a quadruple (LM, DM, AM, AE). Each element of this quadruple reflects to a submodel from the third level of TCM plus its core—an adaptive engine. For each one element of the set (LM, DM, AM, AE) predicates are defined describing basic functionalities of the relevant submodel:

- $LM$  presents the *Learning model*. Learning styles can be defined with constants. The predicate, which indicates the level of single learner's belonging to a learning style, is as follows:
  - $user\_learning\_style(user\_id, learning\_style, value)$ , where  $learning\_style = \{style1, style2, \dots, styleN\}$ .

The predicates representing the students' knowledge and test results are as follows:

- $user\_knows\_domain(user\_id, domain\_id)$ —returns *true*, if a user with identifier  $user\_id$  knows domain area with identifier  $domain\_id$ ;

- *user\_knows\_subject*(*user\_id*, *subject\_id*)—returns *true*, if a user with identifier *user\_id* knows subject with identifier *subject\_id*;
- *user\_knows\_learning\_object*(*user\_id*, *lo\_id*)—similar to the above predicate; this one returns *true*, if a user with identifier *user\_id* knows learning object with identifier *lo\_id*;
- *user\_performance*(*user\_id*, *subject\_id*, *control\_point\_id*, *value*) where *value*={*pass*, *fail*, *notReach*}—the value of a test result can be:
  - ◇ *pass*—if a student has passed it successfully in the control point *control\_point\_id*,
  - ◇ *fail*—if a student has not passed it successfully in the control point *control\_point\_id*
  - ◇ *notReach*—if a learner has not reached the control point *control\_point\_id*.
- DM defines predicates related to the domain model. Some of these predicates are the following:
  - *domain\_lo*(*domain\_id*, *lo\_id*)—returns *true* if a learning object *lo\_id* is part of domain *domain\_id*.
  - *parent\_lo*(*lo\_parent\_id*, *lo\_child\_id*)—returns *true* if a learning object *lo\_child\_id*, inherits the learning object *lo\_parent\_id* in the corresponding ontological graph.
  - *test\_question\_lo*(*lo\_id*, *test\_question\_id*)—presents test questions and learning objects associated with them.
  - *test\_answers*(*test\_question\_id*, *answer\_id*, *value*)—presents test questions, answers associated with them and their values.
- AM contains predicates associated with features of MA, such as:
  - construction of narrative graph
    - ◇ *cp\_path\_4\_graph*(*subject\_id*, *path\_id*)—consists of all WPs for a course or subject and returns *true* if the WP *path\_id* belongs to subject *subject\_id*;
    - ◇ *cp\_path\_4\_cp\_path*(*path\_id*, *page\_id*)—determines whether WP

*path\_id* contains page *page\_id*;

- ◇ *annotation\_cp(learning\_style, control\_point\_id, value)*—presents an annotation (in the element *value*) for the control point *control\_point\_id* according to a value of the learning style such as *learning\_style*={*style1, style2, . . . , styleN*}.
- content of a course
  - ◇ *lo\_4\_subject(subject\_id, lo\_id)*— consists of all learning objects for a course and returns *true* if a course *subject\_id* contains learning object *lo\_id*;
  - ◇ *lo\_4\_page(page\_id, lo\_id)*—returns *true* if a page *page\_id* contains the learning object *lo\_id*;
  - ◇ *control\_point\_4\_subject(subject\_id, control\_point\_id)*—similar to the above predicate; contains all control points (*control\_point\_id*) for a particular course (*subject\_id*).
- sequence of pages and annotation of links:
  - ◇ *link\_pages(current\_page\_id, next\_page\_id)*—presents a sequence between two neighboring pages;
  - ◇ *link\_pages\_annotation(learning\_style\_id, link\_id, annotation)*—presents an annotation for link *link\_id* and learning style *learning\_style\_id*;

AE defines predicates related to the adaptive engine such as:

- *next\_cp\_path(user\_id, subject\_id, previous\_cp\_id)*—defines the WP of a student *user\_id* to the next CP;
- *sub\_precondition(subject\_new\_id, subject\_old\_id)*—describes preconditions (which courses with *subject\_old\_id* the learner needs to pass) for starting the new course *subject\_new\_id*;
- *precondition\_subject(subject\_new\_id)*—returns *true* if all preconditions are satisfied.
- *user\_precondition(user\_id, subject\_id)*—checks if a user *user\_id* has fulfilled all the necessary preconditions for starting the course *subject\_id*.

Predicates defined in this way describe the main features of TCM and may be used for defining rules for adaptation. These rules may be presented by defining relations between predicates. Adaptation rules can be divided into three main groups depending on their purpose:

- starting rules—describe a learner’s knowledge and the starting conditions for beginning of new course.

If a user knows all learning objects of a particular domain/course, then he/she knows this domain/course—(1), (2):

$$(1). \forall user_i \exists domain_j (\forall lo_k domain\_lo(domain_j, lo_k) \wedge user\_knows\_learning\_object(user_i, lo_k)) \rightarrow user\_knows\_domain(user_i, domain_j)$$

$$(2). \forall user_i \exists subject_j (\forall lo_k lo\_4\_subject(subject_j, lo_k) \wedge user\_knows\_learning\_object(user_i, lo_k)) \rightarrow user\_knows\_subject(user_i, subject_j)$$

If a learner knows all courses required by a precondition for a particular course, then he/she can start learning it—(3), (4):

$$(3). \forall subject_j \exists user_i (user\_knows\_subject(user_i, subject_j) \wedge sub\_precondition(subject_k, subject_j)) \rightarrow user\_precondition(user_i, subject_k)$$

$$(4). \forall user_i (user\_precondition(user_i, subject_j)) \rightarrow next\_cp\_path(user_i, subject_j, null)$$

- rules for the graph crawling

If a learner does or does not pass successfully the test at a control point, she/he continues respectively forward (5) or backward (6):

$$(5). \exists k (user\_performance(user_i, subject_j, control\_point_k, pass)) \rightarrow next\_cp\_path(user_i, subject_j, control\_point_k)$$

$$(6). \exists k (user\_performance(user_i, subject_j, control\_point_k, fail)) \rightarrow next\_cp\_path(user_i, subject_j, control\_point_{k-1})$$

- rules for updating the learner model

If the learner passes all of a control point’s for a particular course then he/she knows this course—(7):

$$(7). \forall k (user\_performance(user_i, subject_j, control\_point_k, pass)) \rightarrow user\_knows\_subject(user_i, subject_j)$$

If the learner passes a particular control point's test, then she/he knows the learning objects contained in the selected control point path—(8):

$$(8). \forall k \exists i (user\_performance(user_i, subject_j, control\_point_t, pass) \wedge page\_A\_cp\_path(path_m, page_d) \wedge lo\_A\_page(page_d, lo_k) \rightarrow user\_knows\_learning\_object(user_i, lo_k)$$

According to the description of the proposed conceptual model of AHS it can be concluded that it allows delivery of adaptive educational content. This delivery shall be adapted according to the learner's knowledge and learning styles through the weights of WPs in a narrative storyboard graph and weights of learning objects put in pages of this graph and representing its nodes. To use TCM successfully, it is necessary to develop a methodology for designing learning content and an instructional design for its adaptive delivery.

**4. Methodology for constructing adaptive content and managing the process of adaptation.** Designing learning content and placing it in a course are two iterative and incremental processes, which are consistent with the LM. After the educational content is created by the author, it must be linked appropriately in the course by the instructor and adapted according to the purpose, knowledge and learning style of each student. These two processes have influence mostly on the efficiency of adaptive e-learning. The process of creating learning content refers to the domain model of TCM, while course design refers to the adaptation model of TCM. In this section, workflow methodologies adapted according to these two models will be proposed.

**4.1. Methodology for creating learning content.** Content authors are responsible for the creation and management of learning content. The main characteristics which learning content must meet are as follows:

- the content has to be properly structured—its has to be placed in a suitable ontology and ontological graph. Each ontological graph presents a formal description of a particular domain using a conceptual scheme. Thus, knowledge can be described by a graph or hierarchical structure of objects associated with the presentation of links and relations between them. Objects are concepts/knowledge, which can be both basic and specific. In order to describe the sequence of acquisition of knowledge, more general concepts should be placed higher up in the hierarchical structure of the ontology, as

opposed to more specific ones. The links in the ontological graph can be of two main types: *is\_a*—semantic sub-typing and *has\_a*—a reference. When creating an ontology, its author/s has/have to annotate it according to the standard for metadata ontologies OMV [30].

- content structuring enables reuse in different courses and contexts—i.e., it has to be with the lowest possible granularity, and thus it can be used in different places and with different goals.
- it has to be detailed according to fields of the standard LOM [24]. This makes it easier to find relevant content from the instructor or other systems for Internet-based learning.

Following the domain model of TCM, the learning content has to be presented by learning objects. These learning objects are grouped into ontological graphs. Four main types of learning objects that constitute ontological graphs can be distinguished. These are as follows:

- *narrative content*—may contain formatted text, video, audio, graphics, references to other educational sites. The author should specify the sequence and level of complexity of each object. The level of complexity is relevant to establishing a LO and helps the instructor in choosing the appropriate content. The objects of this type in the ontology are associated among themselves by a connection of type *is\_a*;
- *test question*—may be relevant to one or more learning objects (of type *narrative content*), but must be connected with at least one of them in order to participate in the learning process. Like the previous type of learning object, a *test question* contains hypermedia content in the form of a question. Questions can be of type one of many (1: M) and many of many (N: M) and they must contain at least two answers. For each answer the author sets as an integer (possibly negative) number of points that its selection carries. The questions are used by the AE. It generates automatically a test in control points based on learning objects visited by each student and evaluates automatically his/her answers. In the ontological graph this type of learning object can be connected only with the learning objects of type *narrative content* through a connection type *has\_a*;
- *task*—contains a textual description of requirements for a task or a description of a problem that the student must solve. The result is evaluated by the teacher or instructor and input by him/her into the system. This result influences only the final grade, i.e., it is not taken into account in the process

of adaptation and adaptive delivery of content. As well as the former type of learning objects in the ontological graph, a learning object of type *task* may be relevant to one or more learning objects of type *narrative content* and a *task* is connected with a *narrative content* with a link of type *has\_a*. Even if it is not connected to any other learning object, it can be used to create an adaptive course;

- *essay*—specifies one or more essay topics and describes the requirements, evaluation criteria and guidelines. It is estimated outside of the system and the result is not taken into account by the AE. Again, like the two types of learning objects described above, it may be relevant to one or more learning objects of type *narrative content* and connects with them within the ontological graph by connection of type *has\_a*;
- *others*—besides the types described above, there may be defined other types such as project or game. Their results are assessed and links between this type of learning objects and objects of type *narrative content* will be of type *has\_a*.

Fig. 6 shows the distribution of learning objects in accordance with Honey and Mumford's established and commonly used family of learning styles [32]. The family includes four predefined styles:

- *activist*—he/she is enthusiastic about new ideas, experiments and seeks challenges. An activist prefers to be actively involved with performing a task directly, rather than listening to lectures and detailed descriptions;
- *reflector*—he/she does not proceed to action, but prefers to observe a situation from different perspectives and gather as much information about it. A reflector likes to analyze and work with examples and detailed plans to accomplish a task, project or problem.
- *theorist*—he/she is opposite of the *activist*. A theorist is interested in research, formalization, concepts and logical theories. He/she works well with symbols and abstract concepts and is fully oriented to scientific research.
- *pragmatist*—he/she is opposite of the *reflector*. A challenge for students with such prevailing style is to apply theoretical ideas into practice. Most important for them is to acquire practical skills. A pragmatist is attracted to work on real projects and he/she does not go deep into study and analysis of abstract ideas, concepts and theories.

Honey and Mumford's model is based on Kolb's theory [50], according to which education has two dimensions—perception (axis *y* in Fig. 6) and informa-

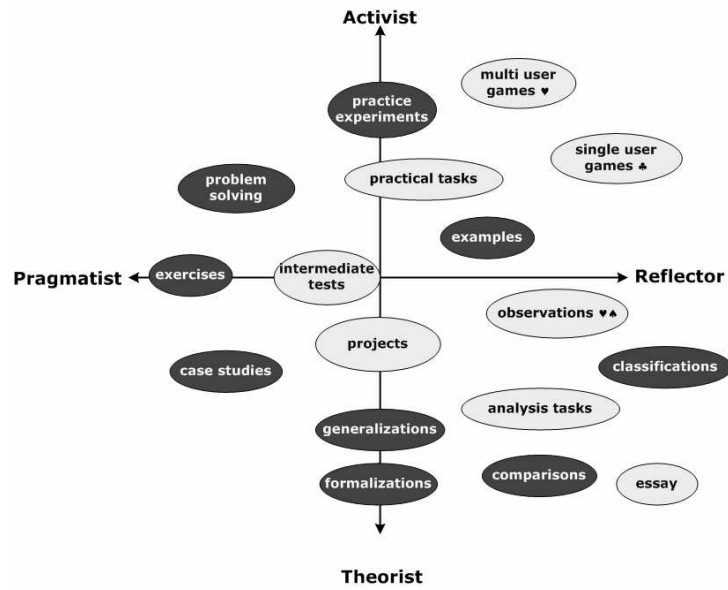


Fig. 6. Distribution of LOs types according their suitability for learning styles [10]

tion processing (axis  $x$  in Fig. 6). Each of these dimensions is bipolar and thus four styles can be represented by a two-dimensional coordinate system.

According to Honey and Mumford, a preferred style is dominant but there always are elements from the other three styles added to it. This should be taken into account by teachers using different pedagogical strategies which satisfy different learning styles.

**4.2. Design of narrative graph for an adaptive course.** When designing and constructing a narrative graph, the instructor must take into account LM on the one hand and DM on the other. His/her duty is to select, group and distribute learning objects according to the learner's current knowledge, abilities and learning style.

At the beginning of the course the teacher has no information about the learner's knowledge and therefore he/she should be guided solely by learning styles. For this purpose, narrative graphs must have several initial nodes (or narrative pages) or start with one that gives basic information, and then split. To cover the four learning styles of Honey and Mumford, the narrative graph must have at least two WPs in which educational strategy is aimed primarily at two poles of the axis  $x$  or  $y$  and includes elements from the other axis (Figs 7 and 8). For example, at least one WP has to be designed for an activist and another



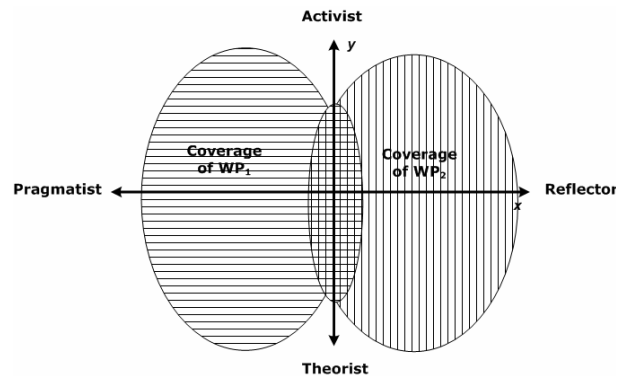


Fig. 7. Distribution of coverage of the four learning styles of Honey and Mumford by WPs for axis  $y$

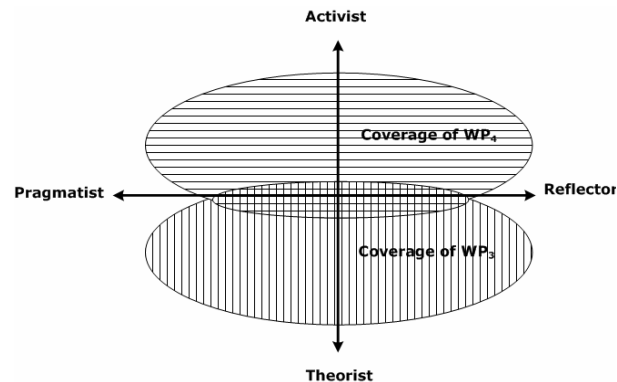


Fig. 8. Distribution of coverage of the four learning styles of Honey and Mumford by WPs for axis  $x$

one for a theorist, as both include elements for pragmatists and reflectors.

After defining the first CP, adaptation of educational content can be added both for learning styles and for the level of acquired knowledge. For this purpose, narrative pages can be composed of learning objects with different difficulty. The difficulty can be represented in the form of several degrees (e.g., very easy, easy, mean, difficult, and very difficult). In this case, for each of them there must be determined when a learning object is visible for a given learner depending on the learner's test result in the CP. For example, if a learning object is assigned a level of visibility 67, it will be seen by students visiting the page in which the learning object resides and having score over 67% in the last visited CP. Thus, students

who have a higher level of knowledge will have the opportunity to deepen it, while others with lower scores in the last CP will not be hampered with materials that are not up to their learning abilities.

The educational content of a course is distributed into narrative pages that, together with the controlling pages, are nodes of the narrative graph. It may consist only of learning objects defined in different ontological graphs of the domain model. At a CP, a test is generated for given learner consisting of randomly selected questions related in the ontological graph to the learning objects already provided to that learner. Thus, the content of a test cannot be controlled when designing a course, but only in designing teaching materials. Moreover, students who have visited various learning objects (of varying difficulty) will receive different questions (of varying difficulty). The instructor should create a valid narrative graph that does not have cycles and each WP ends with CP.

Adaptation of educational content is managed on two levels of TCM:

- adaptive navigation—on this level the student is guided in the narrative graph in accordance with his/her profile and available pedagogical strategies;
- adaptive content selection—here suitable content is presented to the individual learner based on the learner's test results shown in a CP.

For these two levels of adaptation, settings can be made during the creation of learning content and course or later during the learning process. In creating a course, a threshold should be defined for each CP of the narrative graph. If a student's test result is above this threshold, he/she continues forward in the narrative graph. Otherwise, it is assumed that the student has not mastered the minimum necessary to continue and he/she shall be returned to the previous CP. There, the AE should offer him/her another WP suggesting that previous WP was probably not chosen adequately. At a later stage, the instructor is supposed to analyze the reasons for the failure (the selection and/or arrangement of content and/or test questions or the learner) and make corrections where necessary. In the narrative graph, several WPs must be established between each two CPs (as already discussed, there must be at least two). For each of them weights must be defined. Each weight reflects how suitable the respective WP is for a given polymorphic learning style. For example, in Fig. 9 weight  $W_{WP1.1} = (40, 50, 20, 70)$  may be defined for the path WP1.1. As seen before, the weight is represented by four values that correspond to the degree of relevance for each of the four learning styles—*activist*, *reflector*, *theorist* and *pragmatist*.

It should be noted that any learner's learning style is polymorphous, i.e., it is represented by an ordered quadruple like the weight of the WP. Based on

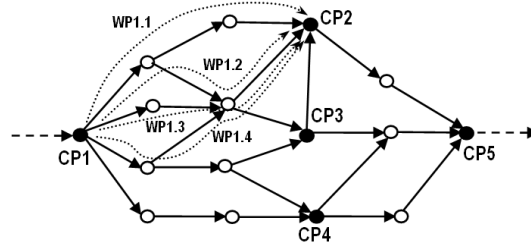


Fig. 9. Example for a narrative graph in which paths are noted with WP1.1–WP1.4 and control points with CP1–CP5

weights for the WP defined by the trainer and depending on how much a student belongs to a learning style, AE calculates the path most suitable for him/her by using the formula (1.1).

### 5. Type of roles, workflow and software architecture of AES.

Based on the described methodology for creating learning content, adaptive course and management of its delivery to students, several types of roles and workflow may be defined and, hence, the software architecture of the ADOPTA AES that supports TCM. They will be topic of this section.

**5.1. Types of roles in an adaptive system for e-learning based on TCM.** The main actors (roles) in the process of building an adaptive course and its delivery to students under the proposed methodology and TCM are as follows:

- *instructor (teacher)*—designs a course by a narrative graph, which describes various pedagogical strategies appropriate for different learner profiles. For each graph, the instructor sets rules for crawling. He/she uses learning materials created by the author and distributes them among narrative pages depending on a given instructional scenario. Another important task here is to monitor the learning process and to make necessary corrections depending on students' results;
- *author of learning content*—responsible for creation, grouping and description of metadata of learning materials (objects);
- *learner*—follows course content delivered by the AE and solves tests at CPs. Before starting the first course in the system, the learner must complete (solve) a special test for determining his/her learning style. In branching pages of a WP in the course graph, the learner may choose to follow another link to another page and to leave that WP, despite the recommendations of

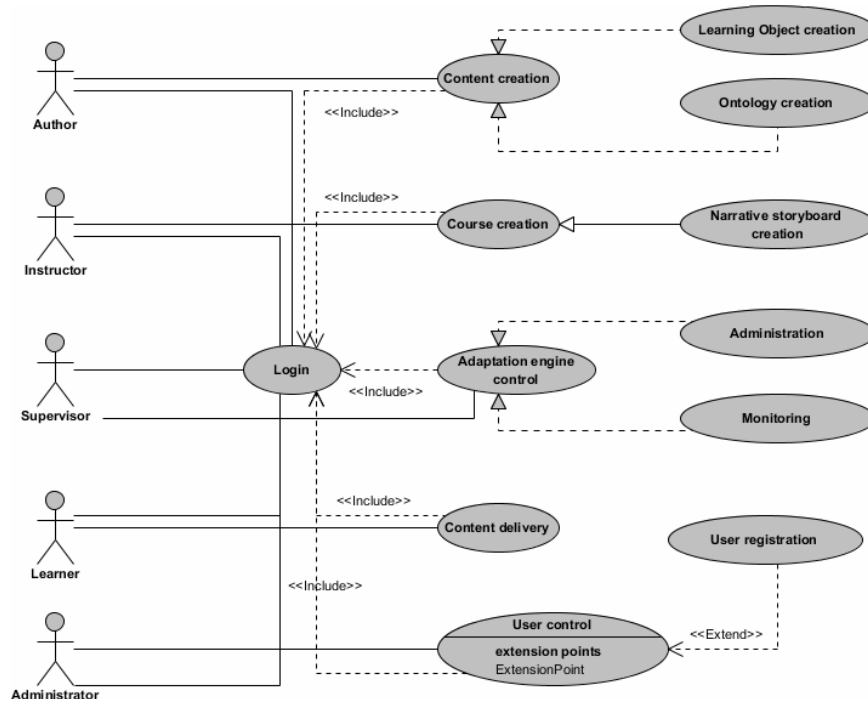


Fig. 10. Type of roles and UML diagram of main use cases

the AE. After leaving the recommended WP, the student can either return back to the old WP, or to continue in the WP chosen by himself/herself;

- *administrator*—supervises all participants' rights and access to training courses.

The main use cases for the roles described above are presented in Fig. 10 through an UML diagram.

**5.2. General workflow of AES based on TCM.** The work process for creation and delivery of an adaptive course expands the non-adaptive one by adding new processes associated with the LM. In Fig. 11 this workflow is shown by an UML activity diagram. It presents the main stages of creating an adaptive course and their sequence.

First, the author creates educational content for the adaptive course. Then it is used by the instructor and finally delivered to the learner. In addition, the instructor monitors the adaptive e-learning process, makes analysis, conclusions and improves the narrative storyboard.

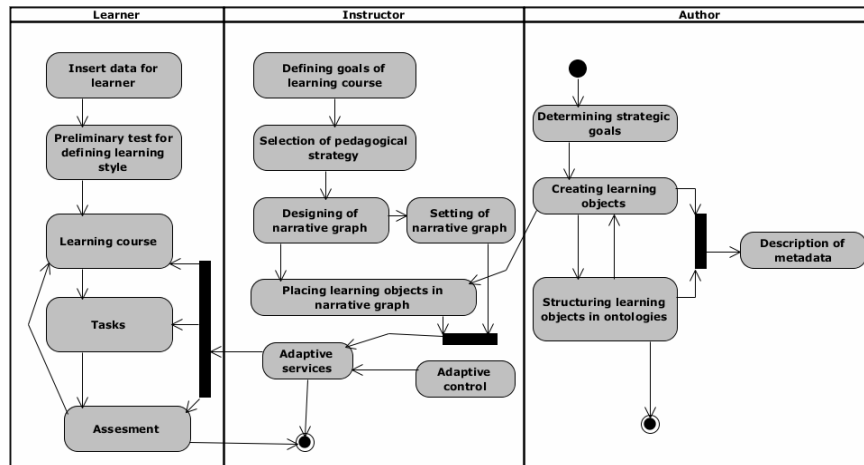


Fig. 11. UML activity diagram of the process for creating adaptive e-learning course

As shown in the figure, three main swim lanes of activities (for the instructor, author and student) are compared to the three basic submodels of TCM. Proceeding from this and the distribution of key players, a delivery system of adaptive content based on TCM may be developed modularly and each of its participants will use a separate independent unit. In this way, any change in any of the submodels of TCM will have an effect on its corresponding module, whereas other modules will be affected as little as possible. Thus the following basic modules may be defined:

- authoring tool—used for the creation of educational content and its description by metadata. This tool is used only by the author, who in most cases plays the role of instructor as well;
- instructor tool—for designing narrative graphs of adaptive learning courses. Here the instructor sets adaptive rules for transition from one node to another, weights of WPs and visibility of learning objects;
- adaptation engine—used by the person responsible for the management of a course, which in almost all cases is the instructor. He/she can set different levels of management of adaptability including complete stop of adaptation;
- admin tool—used by the administrator of the adaptive system; consists of a set of tools for managing users and courses.

Fig. 12 shows a diagram of the general workflow for each role and its corresponding module.

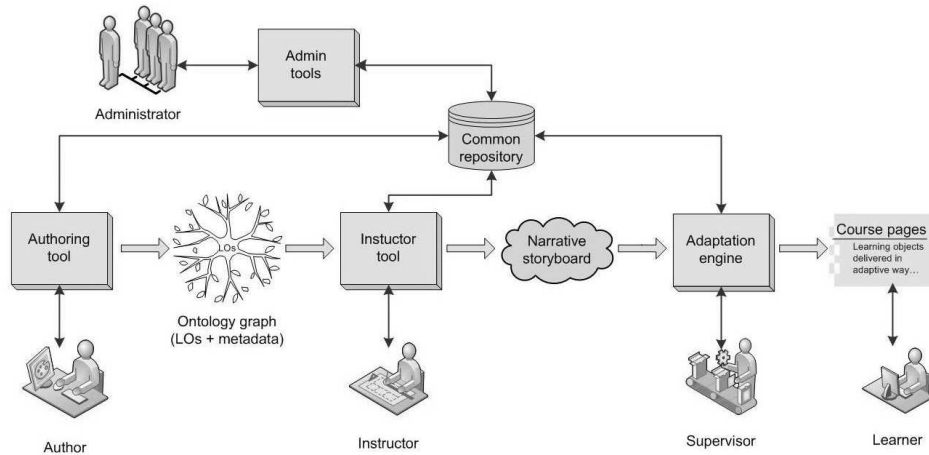


Fig. 12. Principal workflow of adaptive courseware design and delivery [7]

**5.3. Software architecture of ADOPTA.** The software architecture of the ADOPTA platform consists of three main layers—web layer, business layer and persistence layer as shown in Fig. 13. Between the business layer and web layer, however, there is another layer—communication layer. Because it contains no business logic, it only serves as a link between the web and business layers.

The common database defines a common data layer. Thus, objects that are used by several subsystems are reused as part of a common library. Besides reuse, the same logic allows any changes in this layer to remain transparent to others.

ADOPTA persistent semantic units (Adopta Persistence Entities in Fig. 13) are common to all modules of Java objects based on JPA (Java Persistence API) specification for persistent semantic units [53]. They are used for an object representation of persistent data.

Persistent session beans (Fig. 13) are shared and divided into groups (Authoring Tool, Instructor Tool, and Adaptive Engine Persistence Session Beans) for the relevant subsystem. They are used for communication with the common database, i.e., to read, create and modify objects. As a part of the common data layer, here the logic for reading and manipulating of same objects by different subsystems is reused.

The business session layer is a tier which contains the business logic of each subsystem. Each subsystem has a separate layer that solves only the task for which it is intended. Compatible with the previous layer, the business layer is built in accordance with EJB 3.0 technology (Enterprise JavaBeans 3.0) [26].

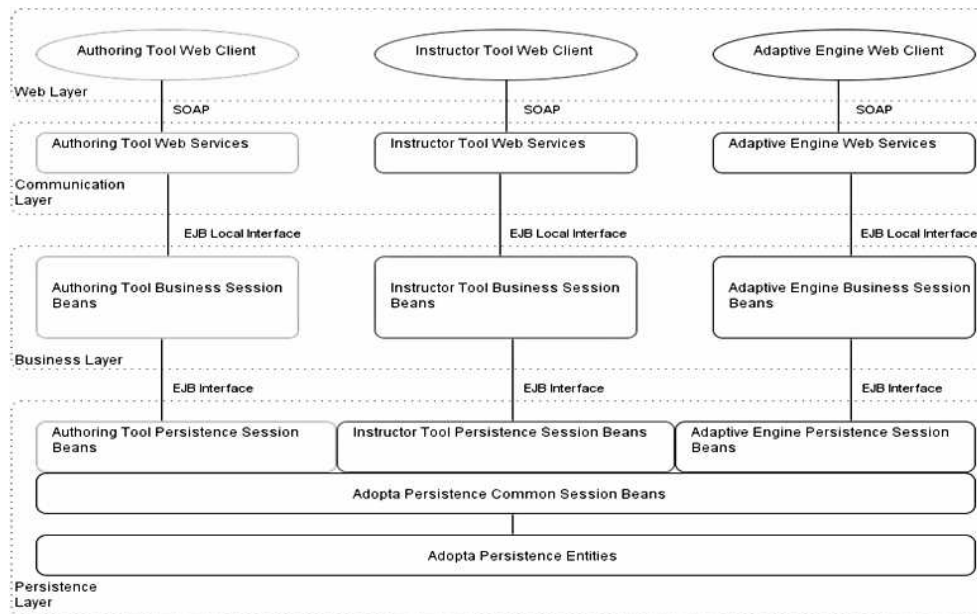


Fig. 13. Software architecture of ADOPTA [8]

The communication layer accomplishes communication between the web layer and the business layer. It is built again based on EJB 3.0 technology and the services which it provides are available in form of Web services. Each subsystem implements Web services that are specific to it. The separation of the connection between the application server and the web client in a separate layer allows easy modification of the way of communication between these two layers, without this affecting business logic.

The client tier is a Web based Adobe Flex client that consumes Web services provided by the communication layer. The Flex technology allows unloading the server layers from dedicated additional resources for visualization and manipulation of the visual data. The layer is implemented in a way incurring only minor changes when possible changes in the communication layer are introduced.

The advantage of the presented architecture is that each subsystem is completely independent from the others. This allows the application to be installed in a distributed environment where each subsystem is located on a separate machine. On the other hand, a clear separation of the subsystems in no way restricts reusing common functionality.

**6. Experimental evaluation of ADOPTA.** In order to investigate the usability of ADOPTA within a field trial, an adaptive learning course and a questionnaire on the effectiveness of its implementation have been developed. For creating the course, methodologies described above were used together with pedagogical strategies for building adaptive training course content. The developed adaptive course was used for teaching bachelor students in Software Engineering at the St. Kliment Ohridski University of Sofia, in the subject “*Modern markup languages (XML)*”. The questionnaire was completed by students who took the course and results were analyzed and summarized.

**6.1. Creating learning content.** Educational content for an adaptive course on *XML* was designed in accordance with the described methodology. According to it, an ontological graph was created with the short name of the course, in this case “*XML*”. To this ontology graph there were added learning objects representing the content of the course. At the top level (as shown in Fig. 14) are located two of the basic learning objects (“*What is XML*” and “*XML Intro*”), which define *XML* in two ways: one is more suitable for activists (the definition is presented schematically in Fig. 15a), and the other is more suitable for theorists (includes the systematic description of Fig. 15b).

These two elements of the ontology are inherited from two types of learning objects. They complement two definitions given in them – with an example

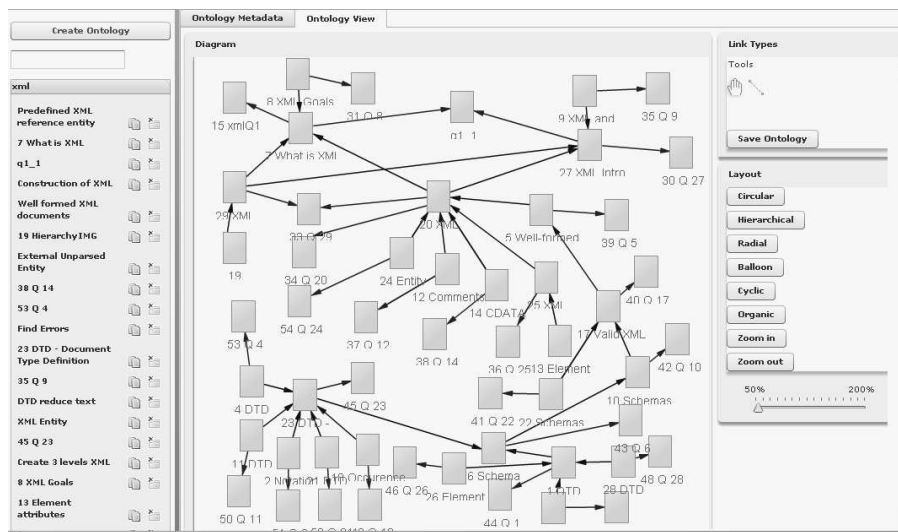


Fig. 14. View of ontology created with the authoring tool



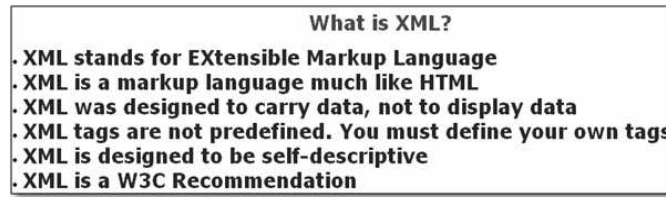


Fig. 15a. Definition of XML suitable for activists

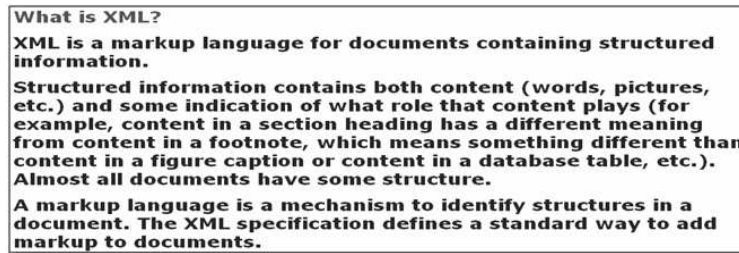


Fig. 15b. Definition of XML suitable for theorists

(*XML Example*) and with a clarification (*XML Hierarchy*) – and others that contain more specific information such as *XML Syntax*.

The learning objects described above are related within the ontological graph “XML” with semantic links of type *is\_a*, which show the sequence of acquisition of knowledge and thus form a hierarchical structure. Another specific issue for these objects is that they are of type *narrative content*, i.e., they present learning content in the form of text, video, audio, graphics and references to other learning objects. Each successive level consists of learning objects that contain more specific information than the previous level and for whose study it is necessary to know the learning objects from the upper level. Another principle which is to be followed for the content composition states that educational materials should be appropriate for learning styles—activist and theorist (as shown in the example of introducing XML—definitions appropriate for an activist and for a theorist are given). Keeping this principle in the creation of learning objects of type narrative content, lower levels of ontological graph “XML” were built (Fig. 14). They contain educational content about syntactic rules for creating well-formed XML (such as how XML should be structured, what the syntax of XML elements may include, etc.) and for validation of XML by means of *Document Type Definition*. All learning objects of this type number 48.

In order to reach an equal distribution of educational material for each of

the four learning styles, there were added learning objects of type of *task*, *essay* and *game*, which are appropriate for the learning styles pragmatist and reflector. Learning materials for pragmatist, according to profile of this learning style, are more practical, such as problem solving or task (Fig. 16a), and for reflector they are focused on analysis, classification, comparison and verification (Fig. 16b). Objects of such type are associated with the reference relation type *has\_a* with objects of type *narrative content*. An object of this type is usually associated with two objects of the type *narrative content*, presenting educational materials appropriate respectively for activist and for theorist. For example, the learning object of type *task* named *Add Element to XML* is related to the element of ontology *What is XML*, and to *XML Intro*. Thereby the hierarchical structure of ontology becomes a graph. The XML elements of types *task* or *essay* included in the ontology number 21. Additionally, there have been added two learning objects of type *game* that implement well-known games—anagram and hangman—and contain educational course content.

Other learning objects that are associated with a reference relation type *has\_a* with objects of type *narrative content* are those of the type *test question*. They number 30 and are designed to be used for assessing students' knowledge about appropriate object to which they relate.

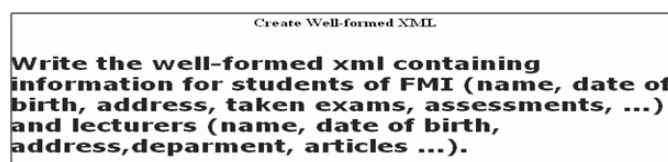


Fig. 16a. Learning object of type task target to pragmatists

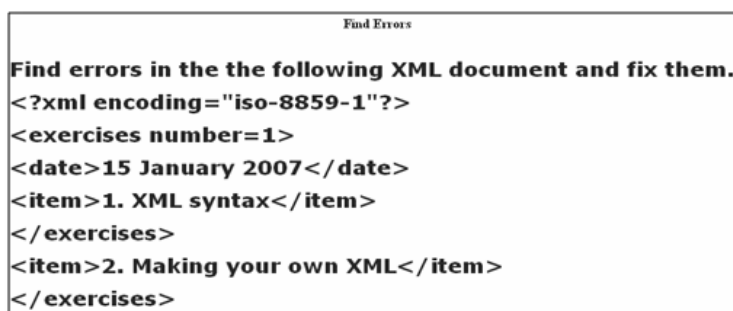


Fig. 16b. Learning objects of type task target to reflectors

**6.2. Design of an adaptive course.** The instructional scenario of the developed adaptive course is consistent with the described methodology for creating an adaptive course. It has two stream lines and WPs, which follow them. The course content in one of these lines is designed for theorists while the other line is more appropriate for the anti-pole learning style, i.e. for activists. In this way, those students who have a higher value for activist than for theorist will be guided by the adaptive engine to the WP which is designed for this style of learning.

Fig. 17 shows the narrative graph of the developed course until the first CP. It has two start pages, labeled *P1* and *P2* in the figure. They constitute starting points of the two main lines in the narrative graph. Page *P1* includes two learning objects—*What is XML* and *XML Example*, i.e., its content is oriented to theorists. Page *P2* also consists of two learning objects—*XML Intro* and *XML Example*—designed for activists. Considering the description of these learning objects above in this chapter, we can conclude that the pages *P1* and *P2* represent the same content in different ways appropriate to particular learning styles. As shown in Fig. 17, both main WPs starting from *P1* and *P2* are divided symmetrically into two other internal WPs (one is marked with a dashed line and the other with a continuous line), through the pages *P1.1*, *P1.2* (for *P1*), and *P2.1* and *P2.2* (for *P2*). These pages contain learning objects of type *task*—*Add Element to XML* (in *P1.1* and *P2.1*) and learning object of type *essay*—*Short List of XML Languages* (in *P1.2* and *P2.2*). Afterwards the so formed two internal WPs of the two main WP merge again in page *P3* and *P4* respectively. This

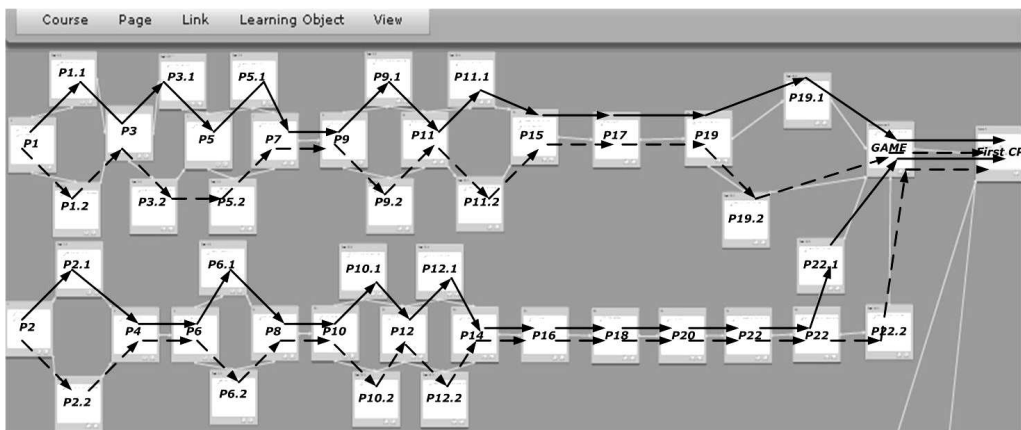
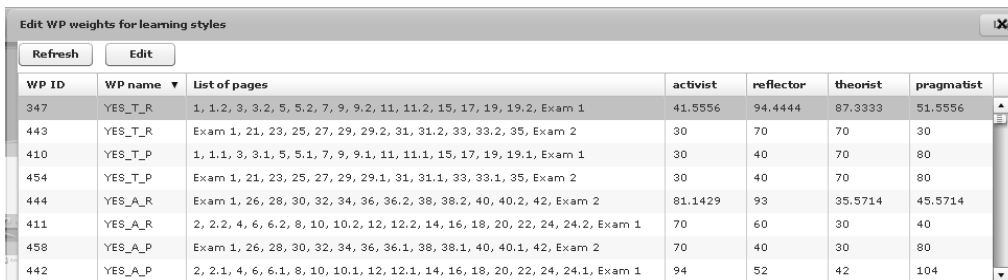


Fig. 17. View of a narrative graph created with ADOPTA authoring tool

principle of construction of the narrative graph is followed to its end. Thus, for each of the two main WPs two sets of internal WPs have been constructed. One of them contains all WPs with appropriate content for activists—pragmatists and theorists—pragmatists (marked in Fig. 17 with a continuous line), and the other, all WP for activists—reflectors and theorists—reflectors (indicated with a dashed line). The aim is to add learning objects to the basic sets of WPs (for activist and theorist), designed to be appropriate for a pragmatist and a reflector. Therefore in some of the transitions in the narrative graph from one page to another students, depending on their polymorphic style of learning, will go through learning objects (e.g., of type *task*), prepared either for pragmatist or for reflector.

After the first CP, the narrative graph is extended to the next CP in a similar way. Again two main lines are followed, which distribute and combine learning objects in pages appropriate to one of two learning styles—activist and theorist. In the pages through which the main WPs is divided into two symmetrical internal WPs, there are placed learning objects for pragmatist or reflector. In the second CP, the adaptive course ends. Before it and before the first CP, there is a page containing a game (anagram or hangman, respectively), through which all learners are supposed to pass, whatever their learning style is. The entire narrative graph consists of 77 pages and 112 WPs.

To avoid heavy maintenance and initialization of weights for all WPs (in the developed course they number 112), only weights of two main WPs were set. For example, as shown in Fig. 18, they are WPs which begin on page *P1*—for activist-pragmatic/activist-reflector, or on page *P2*—for theorist-pragmatist/theorist-reflector. For the other WPs discovered by the instructor tool weights are not defined and thus they are automatically initialized with zero values, which prevents their use by the adaptation engine.



WP ID	WP name	List of pages	activist	reflector	theorist	pragmatist
347	YES_T_R	1, 1.2, 3, 3.2, 5, 5.2, 7, 9, 9.2, 11, 11.2, 15, 17, 19, 19.2, Exam 1	41.5556	94.4444	87.3333	51.5556
443	YES_T_R	Exam 1, 21, 23, 25, 27, 29, 29.2, 31, 31.2, 33, 33.2, 35, Exam 2	30	70	70	30
410	YES_T_P	1, 1.1, 3, 3.1, 5, 5.1, 7, 9, 9.1, 11, 11.1, 15, 17, 19, 19.1, Exam 1	30	40	70	80
454	YES_T_P	Exam 1, 21, 23, 25, 27, 29, 29.1, 31, 31.1, 33, 33.1, 35, Exam 2	30	40	70	80
444	YES_A_R	Exam 1, 26, 28, 30, 32, 34, 36, 36.2, 38, 38.2, 40, 40.2, 42, Exam 2	81.1429	93	35.5714	45.5714
411	YES_A_R	2, 2.2, 4, 6, 6.2, 8, 10, 10.2, 12, 12.2, 14, 16, 18, 20, 22, 24, 24.2, Exam 1	70	60	30	40
458	YES_A_P	Exam 1, 26, 28, 30, 32, 34, 36, 36.1, 38, 38.1, 40, 40.1, 42, Exam 2	70	40	30	80
442	YES_A_P	2, 2.1, 4, 6, 6.1, 8, 10, 10.1, 12, 12.1, 14, 16, 18, 20, 22, 24, 24.1, Exam 1	94	52	42	104

Fig. 18. Initialization of WPs weights for developed course in the instructor tool

**6.3. Survey for evaluation of effectiveness.** In this section, there will be described in brief a questionnaire used for assessing the effectiveness of technical indicators of the adaptive platform such as interface, stability, scalability. On the other hand, the same questionnaire was used to determine the final effect of the adaptive learning course, i.e., the satisfaction of students with the provided educational content, test questions, exercises and tasks.

Questions and recommendations are grouped into several groups. Most of them have answers of type “*one to many*” and their answers use Likert’s five point scale [18] (1 = no, 2 = rather not, 3 = can not decide, 4 = rather, 5 = yes). As well, questions of type “*many of many*” are used. Recommendations in the survey are presented as questions of type *essay*.

The first group of questions assessed the *user interface* and *system operation*.

The second group of questions is aimed at *reliability* and *scalability*.

The next group is related to the evaluation of *learning content* and *structure of the course*.

The last group of questions assess the adaptability to the learning style, which is the main purpose of the system.

**6.4. Results and analysis.** This section examines results of the survey obtained from participants after finishing the adaptive course. These results were analyzed from the following points of view:

- adaptability—there are two issues to this aspect. The first one is whether the system delivers educational content created by the author in the sequence designed by the instructor. The second issue is related to the suitability of submitted learning objects to the learner’s learning style;
- reliability and scalability of the prototype—it should be checked if there is a delay in the execution of the application and if technical error messages appear;
- usability of the prototype—here the objective is to assess whether the prototype provides students with an intuitive and easy to use interface;
- level of efficiency of educational content—it is important first to find the level of acquiring of the material by students within the adaptive course compared to those trained with a non-adaptive one and, second, to determine the students’ level of satisfaction on the basis of visited pages with educational materials.

Each of the above aspects corresponds to a group of questions of the survey. For analysis, there were used results from the survey and, as well, assessment

results at control points of the adaptive course compared to test scores of students who have trained with a traditional, non-adaptive e-learning course in the same subject, namely “*Modern markup languages (XML)*”.

The group of questions numbered from 13 to 24 evaluated the quality of the developed learning content, course structure and student satisfaction with the training. The results of these questions are presented in Fig. 19, where the variance is strictly positive with an exception only for question number 17, which was given in negative form.

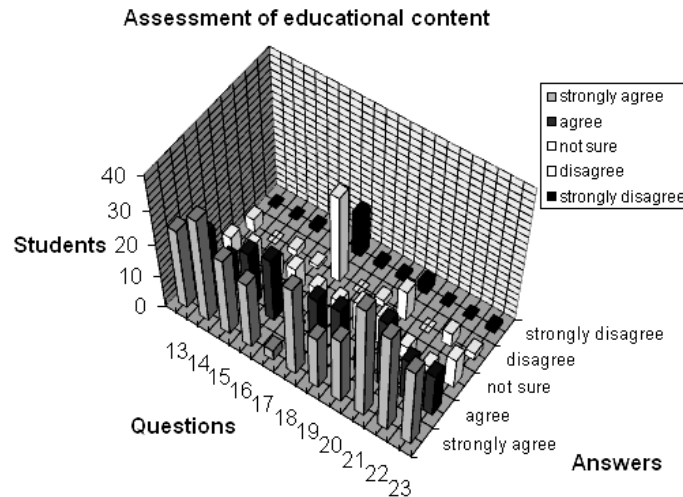


Fig. 19. Assessment of educational content

The last group of questions assessed the correlation between presented learning objects and relevant learning style (question 26 and question 27), preference to an adaptive learning system in comparison with a non-adaptive one (Question 28) and effectiveness and student satisfaction in the process of learning with an adaptive platform (question number 29, 30 and 31). The results of this group of questions are presented in Fig. 20. It is obvious that the answers are quite positive and confirm adaptability.

Another way (apart from question number 28 and 29) that was used to compare the effectiveness of the training for both the classical and the adaptive learning approach was through analysis of results of tests conducted by students in both courses. In the evaluation of the non-adaptive course 42 students have participated, and in that of the adaptive course, 49. Students from both groups had passed through the same training modules and were assessed by solving a test after each module. For the assessment, the average of their results in percents was

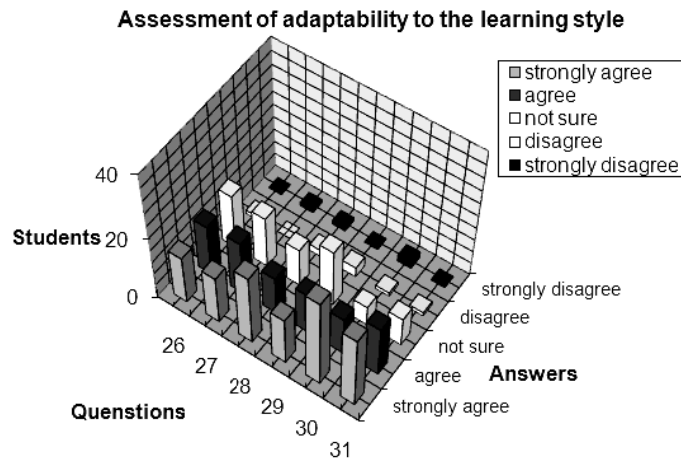


Fig. 20. Assessment adaptability to learning style

taken. The results of this evaluation are presented in Fig. 21. It may be seen that learners that are trained with the adaptive course have higher results. This fact is practical evidence that the adaptive course is characterized by higher efficiency of training compared with that of the non-adaptive one. The lowest result of the test for students taught with the adaptive course is 57% and is due to the fact that a threshold greater than or equal to 57% is defined for this course. When this threshold is not reached, the student is returned to the previous module.

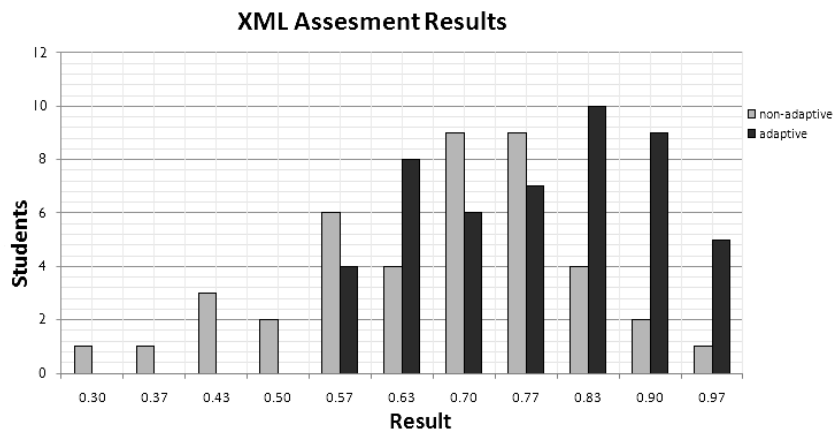


Fig. 21. Assessment results from adaptive and non-adaptive courses

Table 1 shows average test results, dispersion and standard deviation for both groups of students. According to Table 1, the group of students who take the adaptive version has average score 0.778912, but for the other group it is 0.671429. The median, dispersion and standard deviation values for the adaptive e-learning are respectively 76.77%, 1.51%, and 12.3%, while for the non-adaptive learning assessment their values are 70.00%, 2.23%, and 14.95%. These data show again that learning with an adaptive system is better than with a non-adaptive one.

Table 1. Statistical data on test results of both groups of students

	Average result	Median	Dispersion	Standard deviation
Non-adaptive system	0.671429	0.7	0.022362	0.149538
ADOPTA prototype	0.778912	0.766667	0.015125	0.122983

**7. Conclusions.** The system prototype of ADOPTA delivers educational content adaptively according to each individual learner's learning style and performance. The system follows the proposed conceptual model of AES, which foresees adaptation of teaching material not only to learning style and performance but also according to the learner's goals and preferences.

The most important contributions of the present work are as follows:

- creation of a flexible and easily extensible model of AES including support of adaptive educational content in each individual learner's style of learning and progress;
- development of a methodology for content development and design of an adaptive course in accordance with the model;
- design of a software architecture of an adaptive e-learning system following the proposed model;
- development of a software platform prototype of AES based on the proposed model and architecture;
- experimental testing and evaluation of the developed prototype.

A useful direction for future development will be the implementation of adaptability according to the learner's goals and preferences. Thus, the conceptual model of AES will be reflected fully in the developed AES. Moreover, an important area of adaptability in which the adaptive platform may evolve includes the ability to create adaptive tests. In this way, instructors will have



an opportunity for both adaptive content delivery and adaptive evaluation of learners.

Other directions in which the software prototype of the platform ADOPTA can be developed further are as follows:

- Introducing a possibility of obtaining feedback (comments) from the learner in control points and using the results for managing adaptability (i.e., adaptive navigation and adaptive content selection);
- Implicit detection of learning styles and other characteristics of the player model [3] by intelligent agents during the whole educational process;
- Including of artificial intelligence into the algorithm for adaptation;
- Construction of intelligent agents for educational games mimicking player behavior [11];
- Enabling export/import of learning objects or course in accordance with the SCORM standard;
- Developing and using tools for monitoring the interaction of students with the adaptive system.

Based on an evaluation of the results of ADOPTA presented in this paper it may be concluded that AESs improve essentially the quality and efficiency of e-learning content delivery. Thus, web-based teaching would meet the contemporary needs of modern education in an adequate way. Moreover, the ADOPTA platform allows combining traditional e-learning materials with educational games, which is a promising and fast growing research trend as well as market area. Including serious games into the e-learning process makes it more attractive and immersive for all types of students.

## REFERENCES

- [1] ADAMS E. Idealization in Applied First-Order Logic. *Synthese*, **117** (1998), No 3, Springer Netherlands, ISSN: 0039-7857, 331–354.
- [2] AGGARWAL A., N. KUMAR. Adaptive Hypermedia Systems - A Requirement's Perspective. In: Proceedings of the Internet and Multimedia Systems, and Applications, ISBN: 0-88986-508-6, August 15 – 17, 2005, Honolulu, Hawaii, USA, 140–144.
- [3] ALEKSIEVA-PETROVA A., M. PETROV, B. BONTCHEV. Game and Learner Ontology Model. In: Proceedings of Int. Scient. Conf. Computer Sci-

- ence'2011, ISBN 978-954-438-914-7, Ohrid, Macedonia, 1-2 September 2011, 392–396.
- [4] ALVES P., J. PIRES, L. AMARAL. The role of learning styles in intelligent tutoring systems. In: Proceedings of the CSEDU International Conference on Computer Supported Education, 23 - 26, March, 2009, Lisbon, Portugal, INSTICC Press, Vol. I, ISBN: 978-989-8111-82-1, 315–319.
- [5] ANIDO L. E., M. J. FERNANDEZ, M. CAEIRO, J. M. SANTOS, J. RODRIGUEZ, M. LLAMAS. Educational metadata and brokerage for learning resources. *Computers&Education*, **38** (2002), 351–374.
- [6] BONTCHEV B., D. VASSILEVA. Software architecture of a self adaptive hypermedia e-learning system. In: Proceedings of the 3rd Int. Conf. on Computer Science'2006, Istanbul, Turkey, 12 - 15 October 2006, 306–311.
- [7] BONTCHEV B., D. VASSILEVA. Adaptive courseware design based on learner character. In: Proceedings of the International Conference on Interactive Computer Aided Learning (ICL2009), 23-25 September 2009, Villach, Austria, ISBN 978-3-89958- 481-3, Kassel University Press, 724–731.
- [8] BONTCHEV B., D. VASSILEVA, B. CHAVKOVA, V. MITEV. Architectural Design of a Software Engine for Adaptation Control for the ADOPTA E-learning Platform. In: Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech' 09), 18 - 19, June, 2009, Ruse, Bulgaria, Vol. II.11, ACMBUL, ISSN: 1313-8936, 1–6.
- [9] BONTCHEV B., R. GONÇALVES. Ontology-based content development in collaborative environments with semantic services. In: Proceedings of the IADIS Int. Conf. Collaborative Technologies, ISBN 978-972-8939-21-2, 26-28 July 2010, Freiburg, Germany, 2010, 183–188.
- [10] BONTCHEV B., D. VASSILEVA. Learning Objects Types Dependability on Styles of Learning. In: Proceedings of the 8th WSEAS International Conference on ENGINEERING EDUCATION (EDUCATION '11), ISBN978-1-61804-021-3, 14-16 July 2011, Corfu Island, Greece, 227–234.
- [11] BONTCHEV B. A Framework for Educational Word Games. In: Proceedings of the Int. Conf. on Intelligent Computational Systems (ICICS'2012), ISBN 978-81-922428-4-2, Dubai, 7-8 January 2012, 134–138.
- [12] BRUSILOVSKY P. The Construction and Application of Student Models in Intelligent Tutoring Systems. *Journal of Computer and System Sciences International*, **32** (1994), No 1, 70–89.  
<http://www2.sis.pitt.edu/~peterb/papers/studentmodels.pdf>

- [13] BRUSILOVSKY P., E. SCHWARZ, G. WEBER. ELM-ART: An intelligent tutoring system on World Wide Web. In: Intelligent Tutoring Systems (Eds C. Frasson, G. Gauthier and A. Lesgold), Lecture Notes in Computer Science, Vol. **1086**, Berlin, Springer Verlag, 1996, 261–269.
- [14] BRUSILOVSKY P., J. EKLUND., E. SCHWARZ. Web-based Education for All: A Tool for Development Adaptive Courseware. In Computer Networks and ISDN Systems. In: Proceedings of the Seventh International World Wide Web Conference, 14-18 April 1998, 30 (1-7), 291–300.
- [15] BRUSILOVSKY P. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, **11** (2001), Kluwer Academic Publishers, 87–110.
- [16] CARVER C., R. HOWARD, L. LANE. Enhancing Student Learning Through Hypermedia Courseware and Incorporation of Student Learning Styles. *IEEE Transaction on Education*, **42** (1999), No 1, 33–38.
- [17] CHORFI H., M. JEMNI. PERSO : Towards an adaptative e-learning system. *Journal of Interactive Learning Research*, 15 (2004), No 4, 433–447.
- [18] DAWES J. Do Data Characteristics Change According to the number of scale points used? An experiment using 5-point, 7-point and 10-point scales. *International Journal of Market Research*, **50** (2007), No 1, 61–77.
- [19] Dublin Core Metadata Initiative (2009): Dublin Core Metadata Initiative Overview. <http://dublincore.org/documents/dces/>, 2009
- [20] DE BRA P., G.-J. HOUBEN, H. WU. AHAM: A Dexter-based Reference Model for adaptive Hypermedia. In: Proceedings of the ACM Conference on Hypertext and Hypermedia, ISBN:1-58113-064-3, Darmstadt, Germany, 1999, 147–156.
- [21] DE BRA P., A. AERTS, B. BERDEN, B. DE LANGE. Escape from the Tyranny of the Textbook: Adaptive Object Inclusion in AHA!. In: Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (Ed. G. Richards), 2003, 65–71.
- [22] DE BRA P., D. SMITS, N. STASH. Creating and Delivering Adaptive Courses with AHA! In: Proceedings of the first European Conference on Technology Enhanced Learning, EC-TEL, Crete, October 1-4 2006, Springer, LNCS, Vol. **4227**, 21–33.  
<http://wwwis.win.tue.nl/~debra/ec-tel2006.pdf>.
- [23] IEEE LTSC (2004): IEEE LTSC Working Group 12: Learning Object Metadata. <http://ltsc.ieee.org/wg12/index.html>.
- [24] IMS Global Learning Consortium (2004): IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata v1.3. [http://www.imsglobal.org/metadata/mdv1p3pd/imsmd\\_bestv1p3pd.html](http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html)

- [25] JACKSON C. J. Measurement issues concerning a personality model spanning temperament, character and experience. *Handbook of Personality and Testing*. Sage Publishers G. Boyle, G. Matthews, D. Saklofske, 73–93.
- [26] BOSS J. EJB 3.0 Tutorial.  
<http://docs.jboss.org/ejb3/app-server/tutorial/>.
- [27] GONSCHOREK M., C. HERZOG. Using hypertext for an adaptive help system in an intelligent tutoring system. In: *Proceedings of the AI-ED'95, 7th World Conference on Artificial Intelligence in Education* (Ed. J. Greer), Washington, DC: AACE, 1995, 274–281.
- [28] HALASZ F., M. SCHWARTZ. The Dexter Reference Model. In: *Proceedings of the NIST Hypertext Standardization Workshop, 1990*, 95–133.
- [29] HALASZ F., M. SCHWARTZ. The Dexter hypertext reference model. *Communications of the ACM*, **37** (1994), No 2, ISSN:0001-0782, 30–39.
- [30] HARTMANN J. et al. Ontology Metadata Vocabulary and Applications. In: *Proceedings of the Int. Conf. on Ontologies, Databases and Applications of Semantics, Workshop on Web Semantics (SWWS)*, Springer, October 2005, 906–915.
- [31] HRASTINSKI S. Asynchronous and synchronous e-learning. *Educause Quarterly*, 31 (2008). No 4, 51–55.
- [32] KARAGIANNIDIS C., D. SAMPSON. Accommodating Learning Styles in Adaptation Logics for Personalised Learning Systems. In: *Proceedings of the World Conf. on Educational Multimedia, Hypermedia and Telecommunication, 2008*, 1715–1726.
- [33] KAREAL F., J. KLEMA. Adaptivity in e-learning. *Current Developments in Technology-Assisted Education* (Eds A. Méndez-Vilas, A. Solano, J. Mesa), Formatex, 2006, 260–264.
- [34] KAWAMURA T., Y. TAHARA, T. HASEGAWA., A. OHSUGA, S. HONIDEN. Bee-gent: bonding and encapsulation enhancement agent framework for development of distributed systems. *Systems and Computers in Japan*, **31** (2000), No 13, 42–56.
- [35] KINSHUK LIN T. Application of learning styles adaptivity in mobile learning environments. *Third Pan Commonwealth Forum on Open Learning*, 4-8 July 2004, Dunedin, New Zealand.  
<http://www.col.org/pcf3/Papers/PDFs/Kinshuk.Lin.1.pdf>.
- [36] KOCH N., M. WIRSING. The Munich Reference Model for Adaptive Hypermedia Applications. In: *Proceedings of the Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*, ISBN: 3-540-43737-1, 2002, 213–222.

- [37] KOLB D. A. Learning styles and disciplinary differences. (Ed. Arthur Chickering and Associates), The Modern American College, 1981, 232–255.
- [38] MOREIRA D., M. MUSEN. OBO to OWL: a protege OWL tab to read/save OBO ontologies. *Bioinformatics*, **23** (2007), No 14, 1868–1870.
- [39] MESSICK S. The Criterion Problem in the Evaluation of Instruction: Assessing Possible, Not Just Probable, Intended Outcomes. *The Evaluation of Instruction: Issues and Problems* (Eds M. C. Wittrock, D. E. Wiley), New York: Holt Rinehart, and Winston., 1970, 183–220.
- [40] HOHL H., H. BÖCKER, R. GUNZENHÄUSER. HypadapterAn adaptive hypertext system for exploratory learning and programming. *User Modeling and user adapted Interaction*, **6** (1996), No 2-3, 131–156.
- [41] OHENE-DJAN J., M. GORLE, C. P. BAILEY, G. B. WILLS, H. C. DAVIS. Understanding Adaptive Hypermedia: An Architecture for Personalisation and Adaptivity. In: *Proceedings of the Workshop on Adaptive Hypermedia AH2003*, August 26-30, 2003, Nottingham, UK.
- [42] OPPERMAN R., R. RASHEV, A. KINSHUK. Adaptability and adaptivity in learning systems. *Knowledge Transfer* (Ed. A. Behrooz), Vol. **II**, 1997, 173–179.
- [43] PARAMYTHIS A., S. LOIDL-REISINGER. Adaptive learning environments and e-learning standards. *Electronic Journal of eLearning*, **2** (2004), No 1. <http://www.ejel.org/volume-2/vol2-issue1/issue1-art11.htm>.
- [44] PETROV M., B. BONTCHEV. Question and test interoperability – organization, reuse and management of assessments’ contents. In: *Proceedings of IADAT Int. Conf. on Education – IADATe-2004*, ISBN 84-933971-0-5, Bilbao, Spain, 7-9 July 2004, 435–439.
- [45] PETROV M., A. ALEKSIEVA-PETROVA, B. BONTCHEV. Learning Content and Interoperability. In: *Proceedings of the 2nd Int. Conf. Computer Science*, ISBN 954 438 526 6, Chalkidiki, Greece, 30.09-2.10, 2005, Part II, 236–241.
- [46] POPESCU E. Diagnosing Students’ Learning Style in an Educational Hypermedia System. *Cognitive and Emotional Processes in Web-based Education: Integrating Human Factors and Personalization* (Eds C. Mourlas, N. Tsianos, P. Germanakos), *Advances in Web-Based Learning Book Series*, IGI Global, ISBN: 978-1-60566-392-0, 187–208.
- [47] REY-LÓPEZ M., A. FERNÁNDEZ-VILAS, R. DÍAZ-REDONDO, J. PAZOS-ARIAS. Providing SCORM with adaptivity. In: *Proceedings of the 15th international conference on World Wide Web*, 2002, ISBN:1-59593-323-9, 981–982.

- [48] RICHTERS M., M. GOGOLLA. On Formalizing the UML Object Constraint Language OCL, LNCS Vol. **1507**, 1998, ISBN: 3-540-65189-6, 449–464.
- [49] ROMERO C., J. J. RÍDER, S. VENTURA, C. HERVÁS. AHA! meets SCORM. IADIS ELearn.Virtual Multiconference on Computer Science and Information Systems.  
[http://www.iadis.net/dl/final\\_uploads/200504A006.pdf](http://www.iadis.net/dl/final_uploads/200504A006.pdf).
- [50] SADLER-SMITH E. Learning Style: Frameworks and Instruments. *Educational Psychology*, **17** (1997), No 1/2, 51–63.
- [51] SCORM (2006): SCORM Overview. NOVEMBER 16, 2006.  
<http://www.adlnet.org/Technologies/scorm/SCORMSDocuments/Previous%20Versions/SCORM%202004%203rd%20Ed/SCORM.2004.3ED.DocSuite.zip>
- [52] STOYANOV S., P. KIRSCHNER. Expert Concept Mapping Method for Defining the Characteristics of Adaptive E-Learning: ALFANET Project Case. *Educational Technology, Research & Development*, **52** (2004), No. 2, 41–56.
- [53] Sun (2007) The Java EE 5 Tutorial. <http://download.oracle.com/javase/5/tutorial/doc/javaeetutorial5.pdf>.
- [54] VASSILEVA D., B. BONTCHEV. Self adaptive hypermedia navigation based on learner model characters. In: Proceedings of the IADAT-e2006 Conference, Barcelona, Spain, July 12-14, 2006, ISBN: 84-933971-9-9, 48–52.
- [55] VASSILEVA D., B. BONTCHEV. Management of storyboard graphs for adaptive courseware construction, in WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, ISSN: 1790-0832, **7** (2010), No 3, 321–330.
- [56] VIET A. N., H. N. VIET, D. H. SI. Developing adaptive hypermedia system based on learning design level B with rules for adaptive learning activities. VNU. *Journal of Science, (Natural Sciences and Technology)*, **25** (2009), No 1, 1–12.
- [57] WENGER E. Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Cognitive Approaches to the Communication of Knowledge, Morgan Kaufmann Publishers, Inc., Los Altos, Vol. **42**, 1987, 99–157.

*Faculty of Mathematics and Informatics*  
*St. Kl. Ohridski University of Sofia*  
*5, J. Bourchier Blvd*  
*1164 Sofia, Bulgaria*  
*e-mail: ddessy@fmi.uni-sofia.bg*

*Received February 22, 2012*  
*Final Accepted March 29, 2012*