Chapter 5

Personalization and User Modeling in Adaptive E-Learning Systems for Schools

Todorka Glushkova

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/61084

Abstract

The manuscript presents a model for the personalization of e-Learning systems in secondary schools. Approaches are discussed about the implementation of this model by the application of the SCORM-standard, ITL (ITL-Interval and temporal logic), policies, etc. Comments on the possibilities for increasing the relevance of e-Learning systems in the real classroom environment schools are also included.

Keywords: E-Learning, User Modelling, Personalization, Adaptation, Interaction, SCORM, ITL

1. Introduction

Dynamically changing realities in modern society require more dynamic and adequate changes in education, which is inherently conservative. Every innovation and change in the traditional school system requires a long time for synchronizing the legal basis, approbation, and implementation in school practice. This generates a continuous delay in the fast increasing requirements to the educational system, which makes it difficult to meet public expectations. The world community sees a way to solve this problem through the application of ICT and e-Learning in the educational process. Environments that offer a variety of teaching materials and services to different user groups, such as students, parents, learners, employees, etc., are created. As a rule, all these systems are developed faster and cost much cheaper than their traditional equivalents. They enable the implementation of new and different approaches from the traditional and provide solutions as well as access to educational materials within the process of learning. The standardization of individual modules and processes bring order to the variety of systems for computer training and make it possible to use them independently because of the software and hardware platform features. All this seemed to solve the problems of didactic theory and practice, but the reality is different. There is a delay in the process of implementation and actual use of these systems despite the rapid development of information and communication technologies. Research groups from different universities and educational communities define the main reason as learning by means of ICT is an innovative process that requires in-depth research by pedagogical science and practice [1]. Psychological attitudes, motivation, and cognitive characteristics of students of different age groups are different in terms of learning processes. The didactic theory and practice for many years examine these processes and provide solutions for the improvement of the traditional forms of training, however, the mechanisms in e-Learning environments of educational portals are still not well investigated and explained. There is a gap between the expectations of developers and real results in the educational practice. The rapid development of computer science and technology for the creation of learning environments requires a high level of qualification and experience of the developers of such systems. These developers are highly qualified and highly specialized IT professionals who create systems in accordance to their abstract vision of the learning process. However, they rarely or never are pedagogical specialists and therefore, do not know the actual in-depth psychological processes of learning. This leads to the fact that institutions have learning environments that have a good software perspective quality but poor quality as a pedagogical tool. To these reasons, we can add the difference in terminology, the desire to maximize profits of software vendors, unclear criteria for evaluation of the learning process, etc. As a result, we see many factors that negatively affect the whole process and can significantly hinder it.

Despite these problems, there are many areas in which the results are very good. Summarizing the results of higher education in the analysis of the Sloan Consortium in [2], successfully use of e-Learning environments in the USA and Canada grew by 20% over the last few years. The indicators are particularly good for students who are trained in a distance form of teaching, as well as with electronic training courses that are similar in nature to the traditional teaching process. Secondary schools also have sectors that experience very good results. Examples are as follows: the using of educational environments in blended learning (classroom training and independent work) and the creation of a portfolio of students who successfully combined with project training [3].

The problem analysis allows us to conclude that the creation of training systems must comply with specifics of the particular educational institution and be developed in direct communication with educational experts as were probed directly in the real learning environment. This publication will present a model for personalizing learning systems for electronic and distance learning in secondary schools by application of didactic methodology, setting of educational goals and objectives, the motivation of the student, and his or her personal goals, plans, and ambitions.

The structure of the manuscript corresponds to the described methodology. In section 2, “Interaction and Adaptation” discusses various aspects of interactivity and adaptability connected with the personalization of the learning process and provides access to educational resources. Here, the adaptive levels of the system in horizontal and vertical plans are reviewed.

Section 3, "Adaptive levels and interaction Student-Learning system", describes the three adaptive levels and some mechanisms for their implementation of the SCORM-based e-learning system.

Section 4, "Personalization and User modelling", are connected with the opportunities for development of the personalization and UM on different adaptive levels according to the described methodology. An algorithm is proposed for the implementation of the model in the e-Learning environment.

The results of the partial implementation of the proposed model of e-Learning in secondary schools are encouraging. Work on the realization of the full adaptive model continues.

2. Interaction and adaptation

According to the definitions in [4], [5], and [6] e-Learning is a computer and an internet-based learning, in which the delivery of electronic learning resources is carried out on the principles of dynamic interaction with the educational system and the other participants in the learning process, according to didactic set goals and objectives and according to the characteristics of the course and the personal characteristics of the student. Based on this definition, the team of University of Plovdiv, together with partners from the Institute of Information Technologies (BG), University of Limerick (Ireland), De Montfort University in Leicester (UK), Humboldt University (Germany), Secondary school in Brezovo (BG), etc., developed a system for electronic and distance training (DeLC[1]). As part of this project, an environment is developed for e-Learning and distance training for secondary schools. In order to minimize the problems mentioned above, we chose a methodology by which, together with pedagogical specialists, creates step by step the different prototypes of the system and tested them directly in the real learning environment (Figure 1).

To create an interactive and adaptive system that meets these requirements, it is necessary to model the different interactive levels and adaptive aspects. We accept the definition in [7] and define it as a dialog between users and the learning system and will view it at the following three levels:

- Standard Experience – the physical structure and hierarchy of the learning content remains unchanged;
- Personal Experience – the hierarchy of content changes and adapts to the user's behaviour and selections;
- Open Experience – an open and live system with continuous engagement between the producer, user, and message.

1 DeLC- Distributed eLearning Centre

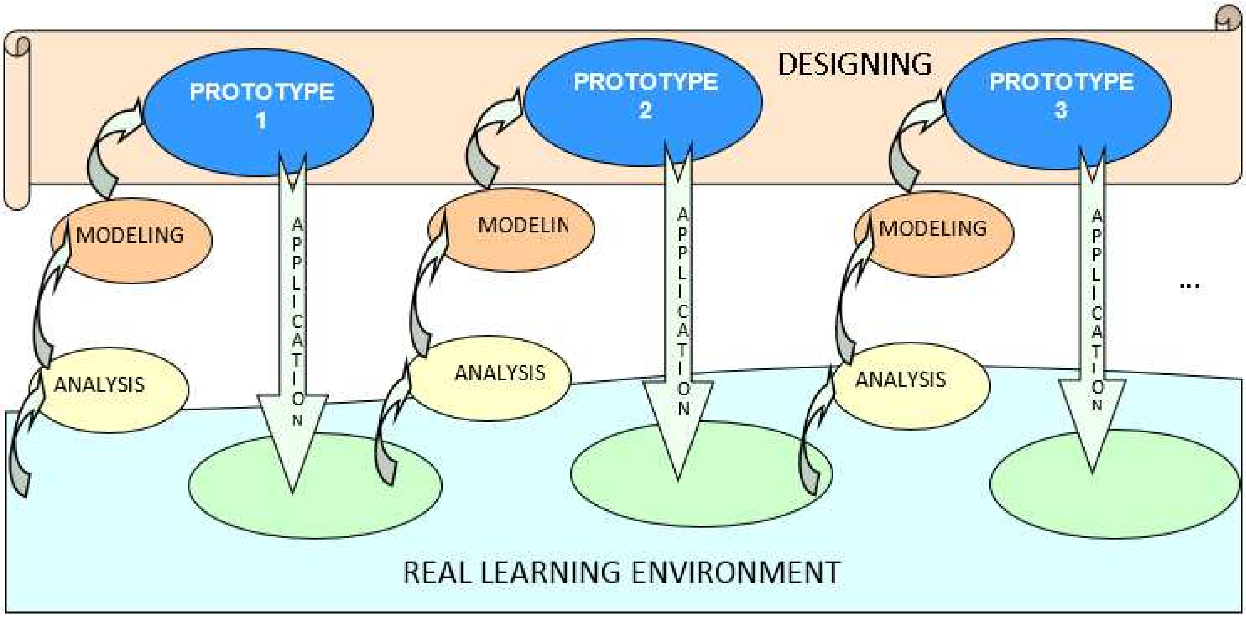

Figure 1. Iterative model of research methodology

Within these levels, there are six categories of interactivity: Feedback; Control; Productivity; Creativity; Communication and Adaptation. Ensuring interactivity for each level and related categories requires the development of a comprehensive adaptive model to provide personalization of the training through: the basic knowledge of the student; his plans and purposes; his cognitive characteristics; his preferences and habits; emotional profile, and so on (see [8]).

According to various aspects of the application and the use of e-Learning systems, the adaptability can be defined differently. We will define adaptability as a feature of the training system to be adapting itself and changed according to the requirements and the characteristics of the users before and during use of it. The main elements of the adaptive model are "condition-action" rules that change the parameters of the environment and realize the adaptation to a user‘s knowledge, goals, abilities, preferences, etc. The different methods and frameworks for creation of adaptive models are as follows:

- Rule-based – we look at these systems from some aspects: as a declarative interpretation of rules; as a hybrid representation based on logical deduction; as a users’ stereotypes; as an overlay model of connecting and co-interacting with the model of the relevant applied area. Presenting the cognitions is connected with the accommodation of the system conventions, the attitude and the convictions of the users, and the stereotypes and the user groups that can be activated dynamically with the particular conditions, etc.

- Frame and Network-based – these models are associated with figuring the sciences as interrelations between separate facts of semantic net and frame structures. It can be used successfully on a small applied domain that can be easily identified and structured.

- Supposition-based – Such systems work with a multitude assumptions of the consumer that forms on the student’s knowledge base and domain independent rules. The suppositions are facts on the user that the system takes with a certain level in security and cogency. The degree of cogency in the system is being raised if the user gives a good feedback and falls if he gives a bad one. The suppositions, established on the base of the direct communication with the user, are better defined that the system is adopted on the base of logical deductions. Formally, we can differentiate the system assumptions on three categories: what the student knows; what he doesn't know; and the student's aims, tasks, and plans. The first two can be realized by stereotype and overlay models as the knowledge of students are being adapted to relevant domain ontology. The third group is part of the Goal and Task Model of the e-Learning system. The most simple is the method of linearly parameterization. It is more complex but more reliable as the model uses formulas, predicates, ITL, policies, and grid models.

- Based on statistical rules and theoretical conclusions – This model permits the adapting of rules according to the state of the entrance data. The opportunities for setting-up the adaptation are based on the information from past learning sessions.

For the development of the adaptive model, we use separate elements from each of these methods we: use the stereotypes and the overlay model at the initial determination of common behaviours rules; define the concrete dependencies from the described school subject domain, by using of domain ontologies; make a system from assumptions for the learner, based on his stereotype, cognitions, and goals; store the information from the last learning sessions and processed it statistically; and deduced abstract conclusions for the user groups and the separate learners. The implementation of the model requires the consideration of the various adaptability aspects of horizontal and vertical principles. The first one is connected with the adaptation to some personal characteristics of the student. This model of realization in this aspect is discussed in [9]. There are two types of adaptation – adaptivity and adaptability. The first allows users to use different facilities for presentation and navigation in predetermined learning content. The second level includes mechanisms for adaptation to knowledge and preferences of students dynamically in the learning process. On this basis, we will distinguish the following three adaptive levels: Elementary Adaptive Level, Static Adaptive Level, and Dynamic Adaptive Level. The first two are connected with adaptivity, and the last one – with adaptability.

In the next section, we will look at these adaptive levels. We will focus our attention on the Static Adaptive Level and will comment on some ideas for the realization of the Dynamic Adaptive Level.

3. Adaptive levels and student-learning system interaction

Elementary adaptive level (EAL) – the adaptation in this level is connected by the use of static user information such as type of training, grade, class, name, access to learning material (by mobile or fixed device), etc. Here, we can use a stereotype approach. The authors of educational resources develop packages of lessons, tests, etc., in accordance with government educational requirements and standards for the typical student, school subject, the class and form of training. The created educational resources are common for all students in the described groups.

We can realize an adaptation on this level by the preparation of training materials, common for all students in the phase before the start of the learning process. These characteristics prove the relationship with the first interactive level – Standard Experience, because the physical structure and hierarchy of the learning content remains unchanged. However physical and cognitive interaction occurs for the users. The student receives the entire information independently if he knows it. At this level, users are an abstract group of people with common characteristics – background knowledge, preferences, cognitive performance, and more. This personalization is the lowest formal level. Although users have their accounts in the school e-Learning environment, they can work in it only on the predetermined way for all other users in the same stereotyped group.

Static adaptive level (SAL) – this level is based on the elementary level and is directly related with mechanisms to provide adequate learning materials for individual students according to their knowledge base, personal goals, and plans.

Before we present the adaptive mechanisms of this level, we will comment on the concept of persona as an aggregated user type. The persona is a description of a fictitious learner. This description is based on different methods, including the personal experience of the teachers, hypotheses, statistical methods, and heuristic analysis [10].

Adaptability of this level is realized through the collection preparation of educational materials and services, foreseeing the actions and behaviour of the persona. The realization is based on the log-information about past interactions between the student and LMS according to the set of rules defined by the teachers. It is necessary to define the background knowledge of this student. We can determine the knowledge in different ways – by initial testing, by results from completed to this moment training sessions, etc. Based on these values, the system joins this student to the persona, who is closest to these characteristics. The system compares the individual characteristics, plans, and objectives of the student with the typical didactic aims, defined in BES. As a result, this lesson that most closely matches with the basic knowledge, goals, plans, and personal characteristics of the student that is associated with this persona starts from the Lesson repository.

Moreover, special attention will be paid to the creation of courses by pedagogical specialists in the article. In accordance with pedagogical theory, this process is cyclical and begins by placing the main didactic aims, passes through specifying learning tasks, develops the profiles of aggregated user types (personas), the establishes learning scenarios with different personas, develops prototypes of the training process, shares this prototype among specialized pedagogical community of educators, experts and heuristic evaluation of this prototype testing in real learning environment, and corrects existing errors and inaccuracies. Each stage of this process requires a qualitative evaluation and heuristic analysis of the pedagogical community and the implementation of appropriate tools in a common environment – ex. Integrated Learning Design Environment (ILDE) [11] (Figure 2).

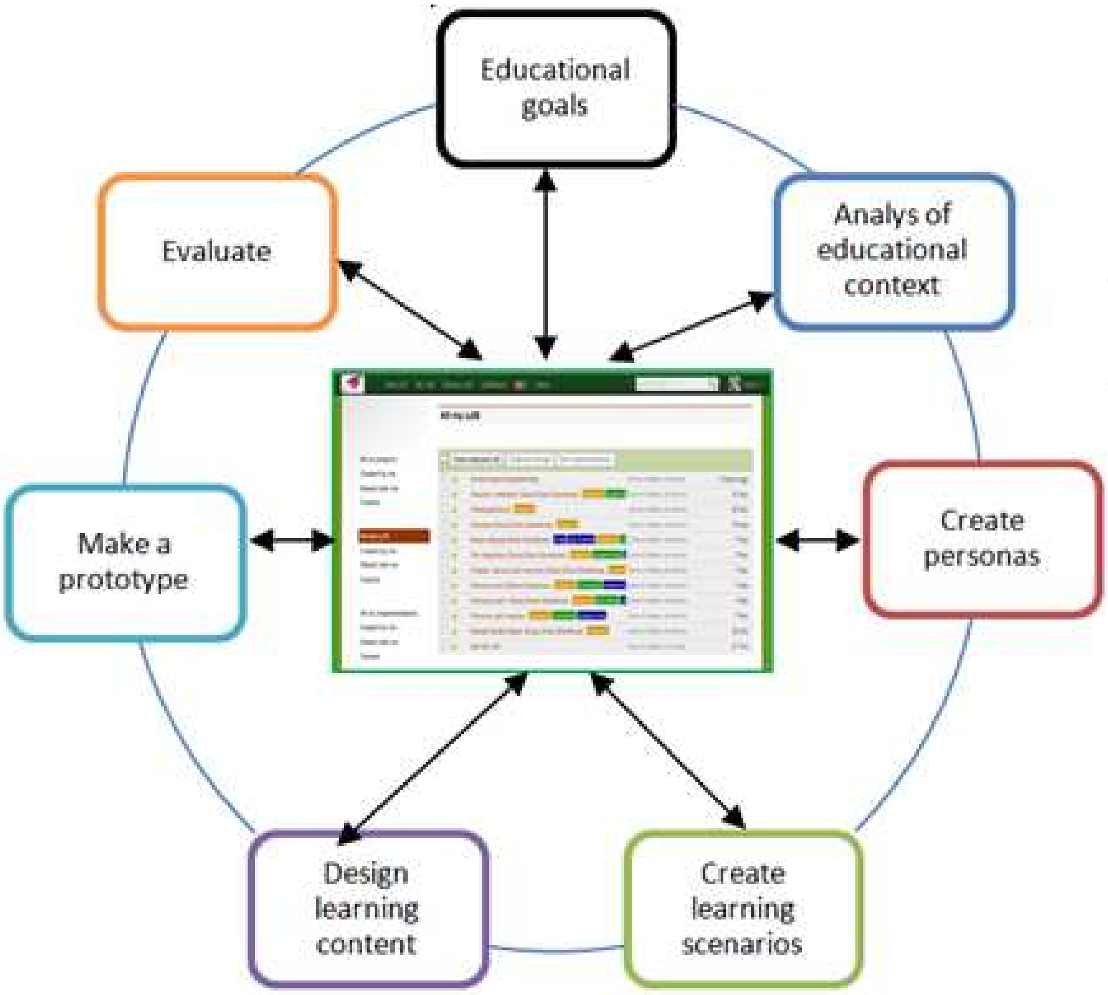

Figure 2. Creation of learning course in ILDE

Creation of the training course, according to the developed scenario is realized based on the selected standard for e-Learning. In developing the DeLC-system, we use the standard SCORM[2] [12]. The team developed a special SCORM-editor for the teachers and authors of the educational content – SELBO [13].

We will concentrate our attention on the two basic characteristics of the lesson – the content and the structure. The content of lessons is related with specific topics that are connected with some school subject domains. The e-lesson presents a semantic structure of the knowledge that is connected with some school subjects. The formalization can be realized through the creation of ontologies in which each concept from the respective area are associated with real information resources that represent them in the lesson. According to the main characteristics of the school subject domain, the authors of e-content in accordance with didactic aims define the structure of the lesson. The didactic aims are related with type of the lesson (for new knowledge, for exercises, summary, and testing). We use Bloom's taxonomy to formalize these

2 SCORM- Sharable Content Object Reference Model

aims with the cognitive levels – knowledge, comprehension, application, analysis, synthesis, and evaluation [14]. The teacher could structure the lesson in different ways depending on the predefined didactic aims. We came to the conclusion that there is a correspondence between the different types of e-lessons and the cognitive levels of Bloom's taxonomy. Therefore, we can formalize the different types of e-lessons according to didactic aims by creating standard scenarios for training and templates that describe them. The template is a combination of structure and learning scenarios.

To create electronic lessons by using algorithm requires a thorough knowledge of the standard SCORM, which creates objective problems for teachers who are not IT specialists. To partly solve this problem, we can use the SCORM Best Practices Guide for Content Developers (BPG) [15], which offers a number of basic templates and models that correspond to different educational scenarios.

In order to increase the formalization level of templates and models, we created a system for its parameterization. Thus, we received a number of different groups of templates that more fully meets the requirements and objectives of the learning process. We can use the following for the parameterization of templates:

- Number_of_SCOs – type Integer, to describe the number of SCOs[3] in the template. If the parameter value Has_test is "yes", the number of questions Num_Quest =Number_of_SCOs – 1;
- Has_test – type Boolean to determine whether there is a final test or not in the template. Defaults to "yes" and is realized with the last SCO;
- Num_Quest – number of questions in the final test
- The ordered pair (objective_n, min_value_n) connects each target variable (objective) and the minimum value for which LMS will mark it as successfully passed (n<Number_of_SCOs);
- The ordered pair (SCOn, template_num), for each n <Number_of_SCOs and template_num <= 10, which connects each SCO with instance of the main BPG-template;

Let's define two operatios:

- Set (SCO_Number_of_SCOs (Asset k); Objective_k) – for setting values of k-th target variable for the k-th issue of the last SCO, where k <Number_of_SCOs, and
- Read (SCO_k, Objective_k) – start-up of the information SCO_k if Objective_k has a value less than the predetermined.

In dialogue with the SCORM-based authoring tool for generating of electronic lessons, the teacher will determine the values of these parameters and the system will generate the structure of the desired template. If specific values of the parameters are not mentioned, the system will get the default values. After the parameterization, the teacher will receive the

3 SCO- Sharable Content Object

parameterized template with the SCORM rules that served as a guide to the sequence of educational activities in the educational scenario depending on the behaviour of the individual student.

For example, if we get parameter values: number_of_SCOs = 10; has_test = "yes"; (objective_n; 0.75) for each n <10; (SCO_n, template_2), for each n <10; Set (SCO10 (asset_n), objective_n) for each n <10; Read (SCO_n, objective_n) for each n <10, we get the following chart describing the scenario of the lesson (Figure 3):

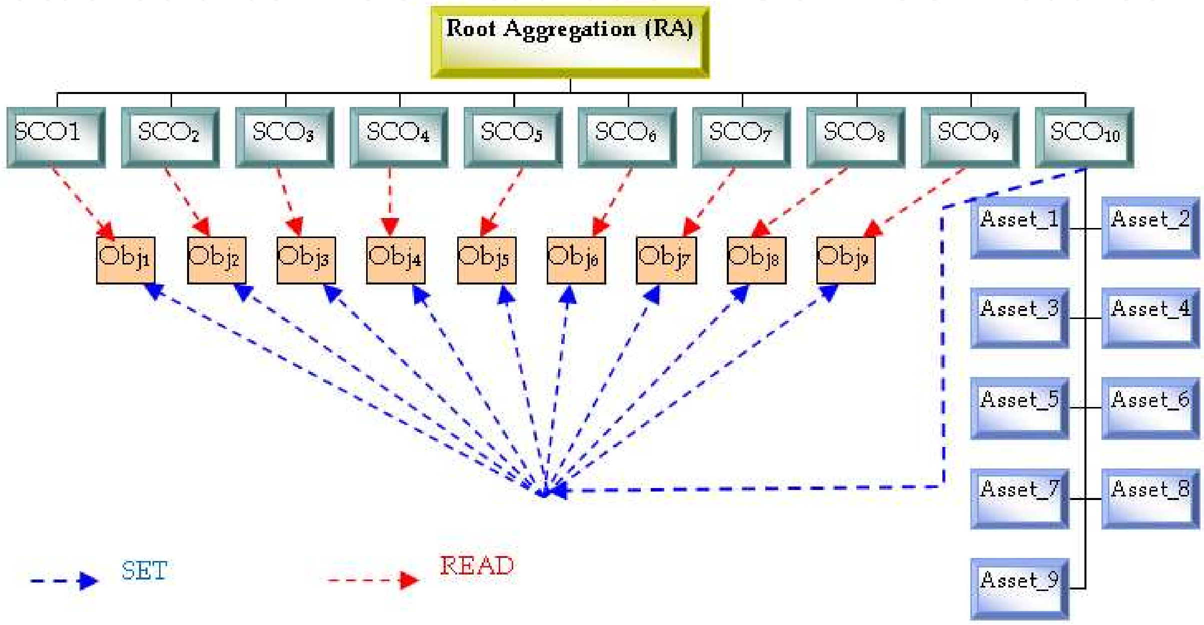

Figure 3. Parametrized Content Structure Diagram

Therefore, based on the Bloom taxonomy of didactic purposes, lesson types formalization of their structure and navigation rules can be created by the step by step algorithm for creating of e-lessons.

The teacher in some school subjects create e-lessons in a specialized development environment, which is in dynamic interaction with the respective ontology. This author's tool has to maintain information about the compulsory concepts in the relevant discipline according to Bulgarian Educational Standards. Concepts that are mandatory taught according to BES and those that are determined from the teacher for this lesson are marked with AND and those that contain additional information are marked with OR. The algorithm includes the following steps:

1. Through a dialogue with the system (e.g. personal assistant), the teacher defines the class, form of training, and didactic aims of establishing a lesson. The system filters relevant domain ontology and retrieves concepts – mandatory and complementary.

2. Depending on the subject area and selected didactic aims, the system offers the most appropriate templates that can be selected from.

3. In a step by step process, the teacher determines the values of parameters such as the number of data objects (SCOs), presence of preliminary and final test, minimum values of the target variables for their passing, number of attempts to solve the tests, etc. As a result, the system generates a parameterized template, which includes the structure of the lesson (SCORM CAM[4]) and the rules that will manage the learning process (SCORM Sequence & Navigation Model[5]).

4. The author connects the SCOs with nodes of the structural graph in the template of the lesson.

5. The author ccreated lesson (the system generates zip-package and imsmanifest.xml)

6. He upload created lesson in SCORM-environment of the education portal.

E-Learning resources (SCOs) are associated with concepts of relevant ontology. They are stored in some online SCOs Repository. In ontologies and related items, SCOs are presented into the development environment for creating electronic lessons [16]. The authors determine the structure of the e-lesson by using the parameterization of some basic templates. In this way, they create an instance of the template in which there are no free parameters. LMS manages educational processes and determines the training scenario in accordance with the structure of the lesson and learning scenario, which are related to the didactic aims, behaviour, and basic knowledge of the students. The e-lesson will be presented in the system as a specific instance of some basic template, which, by setting the values of parameters, is associated with specific learning resources.

One example is the Lesson "Summary on complex verb tenses» for 7th grade students for independent distance training". The didactic aim is to reach higher levels of Bloom's taxonomy – application, analysis, synthesis, and evaluation. The system offers BPG-templates №7, 8, and 10, and the teacher chooses Template 7. In this template, SCOs containing learning information are grouped in a separate Aggregation B. The student must answer questions from the preliminary test and if wrong (i.e. target variables that monitor test results are less than the minimum values), he has to become familiar with the educational content of the information SCOs. Then, he will make the final test in the last SCO. The template can be used in the creation of educational resources, which is necessary to verify and ensure a certain volume of background knowledge. It is essential to fill the gaps and to allow the student to successfully pass the final test. LMS manages the values of the variables (objectives) and only if they are larger than the specified minimum, the training is considered to be successfully completed. The teacher gives the following values of the parameters through a dialogue in a step-by-step process: Number_of_SCOs=9; Has_pre_test="yes"; Num_Quest=9; Has_post_test="yes"; (Obj_n,1) – i.e., answered correctly for all questions from 1 to 9; (Obj_n; 0,75) – gave a very good answer to the questions for $\forall\, n \in [10, 18]$; (SCOn, pattern_2) for $\forall\, ^\circ n \in [3, 11]$; Set(SCO1 (Asset_k); Obj_k) for $\forall\, k \in [1, 9]$; Set (SCO2 (Asset_k); Obj_k) for $\forall\, k \in [10, 18]$; Read (SCO (k +2), Obj_k) for $\forall\, k \in [1, 9]$. The system generates a CAM- model and S & N rules. The teacher

4 CAM-SCORM Content Aggregation Model

5 S&N Model- SCORM Sequence and Navigation Model

writes the test questions and puts the SCOs in Aggregation B. These data objects (SCOs) include both basic information on various complex verb tenses as well as tasks and exercises for students who will have to pass successively through the levels «application» – «analysis» – «synthesis» – «evaluation». The teacher makes a SCORM-package of the lesson and uploads it in the SCORM-based school education portal (http://sou-brezovo.org). These characteristics prove the relationship with the second interactive level – Personal Experience – because the hierarchy of content changes and adapts to the user's behaviours and selections.

Dynamic adaptive level (DAL) – This level is related to the dynamic interaction between students and the system during the training process (in run-time). After selecting the most appropriate e-lesson from the Lesson DB, the LMS starts the learning process according to the training scenario. The learning scenario is realized by a sequence of actions that is previously defined by the author of the lesson. The system observes the intermediate results during the training and information from the already completed training sessions. Based on this information, the LMS adapts itself dynamically to the changing characteristics of the learning environment as it generates new "condition-action" rules and either continues the training process or stops it. If the parameters are not appropriate, the system has to choose and to start a new and more appropriate e-lesson.

We are convinced that in the process of dynamic interaction between the learners and the training system, it is essential to use intelligent agents who interact with the system and with each other to provide a flexible change of training scenarios depending on the behaviours and actions of the individual student. For the managing of the dynamic adaptation of LMS, we can use Interval temporal logic (ITL) and policies.

Morris Sloman in [17] defines the policies as a set of rules for activating different states and actions, depending on the behaviour of the consumers or the current state of the system. There are different techniques to formalize the policies – graphical modelling, using the object-oriented methods for defining of policies, etc. We will use the opportunities provided from ITL [18] as it builds on a classical logic tier and allows to describe dynamic processes in the course of their implementation. It is a flexible notation for handling events that varied in time intervals, allows series, and parallel compositing using a well-defined mathematical proof system. ITL includes four components – logic tier, temporal structures, conditions, and intervals. Classical logic manages variables, constants, functions, and predicates. If we want to describe the dynamic processes, it is necessary to add temporal structures as skip, chop, and chopstar. The states are specific transmission of values to the observed variables and the intervals are sequences of states.

We will describe the next three sets: S-set of students, O-set of available objects or resources, and A-actions that can be performed with these resources. Then, we can introduce the user authentication as one of the Boolean variables:

- $Autho^{+}(S, O, A)$ – Positive identification of the user S, who has right to use the resource O by performing action A. For example $Autho^{+}(Ivan, Lesson1, Read)$ or $Autho^{+}(Ivan, Test1, Write)$;

- $Autho^{-}(S, O, A)$ – Negative identification – the user S refusal to use the resource O by performing action A. For example $Autho^{-}$ (Ivan, Lesson1, Write).

Upon the initial start-up of the system, these variables have a default value of "false". The mathematical model of **Autho** is a matrix with 3 columns – users, objects, actions, and n-number of lines for all users in the system. The access to resources will be allowed if they satisfied certain "condition-action" rules of the type: $F \rightarrow W$ i.e., F always followed by W in the final state of the observed subinterval. According to this definition, the Access Rules take the following form: $F \rightarrow autho^{+}(S, O, A)$– rule for positive identification and $F \rightarrow Autho^{-}(S, O, A)$ – rule for negative identification. For example: If in the initial step the access was denied, but in the next moment, it is authorized in the duration of 10 time units then: $((Autho^{-}(S, O, A) \wedge skip) \vee (Autho^{+}(S, O, A) \wedge len <= 10) \rightarrow Autho^{+}(S, O, A))$. If two users M and N are grouped and one of them has access, then the second one also receives access: $In(M, N) \wedge Autho^{+}(M, O, A) \rightarrow Autho^{+}(N, O, A)$.

The Access Rules determine whether the particular user is entitled to access this learning resource or service. To realize the access itself, the management passes the Implementation Rules, which has the following more general form: $F \rightarrow Autho(S, O, A)$. There are two alternatives in access: Open Access and Restricted Access. Open Access has low security – i.e., if access is not prohibited, it is allowed: $\neg Autho^{-}(S, O, A) \rightarrow Autho(S, O, A)$. Restricted Access means the system checks whether access is allowed and it has meanwhile been prohibited i.e., $(Autho^{+}(S, O, A) \wedge \neg Autho^{-}(S, O, A)) \rightarrow Autho(S, O, A)$.

Another way to access learning resources is the delegation of rights to the unauthorized user. For example, the teacher gives access rights to other teachers for reading a lesson: $Teacher(S, Lesson) \rightarrow Candeleg^{+}(S, \_, Lesson, Read)$. The rules for delegating access, which author A1 gives teacher T2 to make corrections in Lesson1 is: $(Autho(A1, Lesson1, Write) \wedge Candeleg(A1, T1, Lesson1, Write)) \rightarrow Autho(T1, Lesson1, Write)$.

The policy P is a collection of rules: $P \cong (w \wedge (\Lambda\, ri) \wedge fin)$, where w is the initial state, w' is the final state, and Λr_i is a conjunction of intermediate states. For example, the policy for Author of Lesson1 (Author), teacher, who use this Lesson1 (Teacher), and student (Student) is:

$$
\begin{gathered}
P1 \cong ((Author(S, Lesson1) \rightarrow Autho^{+}(S, Lesson1, Read)) \\
(Author(S, Lesson1) \rightarrow Autho^{+}(S, Lesson1, Write)) \\
(Teacher(S, Lesson1) \rightarrow Autho^{+}(S, Lesson1, Read)) \\
(Teacher(S, Lesson1) \rightarrow Autho^{-}(S, Lesson1, Write)) \\
(Student(S, Lesson1) \rightarrow Autho^{+}(S, Lesson1, Read)) \\
(Autho^{+}(S, Lesson1, A) Autho^{-}(S, Lesson1, A)) \rightarrow Autho(S, Lesson1, A))
\end{gathered}
$$

The first step towards the creation of our school e-Learning system is the standardization of key processes associated with the personalization of access to e-lessons.

The teachers create e-lessons in specialized SCORM-compliant and ontology-based development environment, then publish them in the education portal in a special Lesson-DB. Further to SCORM-metadata, we will use some additional specifications such as:

- **Info** – title of the lesson, school subject, author, etc. features that are supported by the SCORM-metadata;
- **Subdomain** – matrix with concepts that will be included in the lesson and the extent of their studying Subdomain(concept, m), where m=1,2,3 as: 1 – low level of studying (mandatory minimum, according to BES); 2 – good level и 3 – high level;
- **Num_Grade** (the grade, for which is intended the lesson) – an integer from 1 to 12;
- **Form_of_training** (form of training): 1 – regular training; 2 – self training;
- **Lesson_Status** (status of the lesson) – an integer between 1 and 4: 4-free for use by all users in this and other portals in DeLC-education network; 3-free for use only by students and teachers in the portal; 2-authorized use only for certain users; 1-unavailable for other users, except for the author; **Didactic_aims** (didactic aims, according to Bloom's taxonomy) – an integer between 1 and 5: 1 acquisition of new knowledge (level "knowledge" and "comprehension" in the Bloom's taxonomy); 2 actualization of old knowledge (level "comprehension", "application", and "analysis"); 3 exercise and improvement of knowledge (level "application", "analysis", and "synthesis"); 4-generalization (levels "analysis", "synthesis", and "evaluation"); and 5-exam (level "evaluation").

Therefore, any electronic lesson in the education portal is a vector with the above dimensions:

$$\mathbf{Lesson}\big(\text{Info}\big(\text{ID, title, domain, author,...}\big), \text{Subdomain}\big(\text{concept}, \text{m}\big), \\ \text{Num\_Grade, Form\_of\_training, Lesson\_status, Didactic\_aims}\big)$$

For example, the lesson "Past imperfect tense of the verb", school subject "Bulgarian language", for 5th grade; author Sarafov, with concepts from matrix Subdomain, designed for regular students, free for use for all users in the education system, and is a lesson for new knowledge we get:

$$\text{Lesson1 }\big(\text{Info}\big(\text{ID, Past imperfect tense of the verb,Bulgarian language, Sarafov,...}\big), \\ \text{Subdomain}^* \, 5,\ 2,\ 4,\ 1\big),$$

where Subdomain* is present with Table 1.

Conceptions	Level of Studying
Verb	3
Person of the verb	2
Tense of the verb	2
Communication moment	3
Moment of action	3
Main orientation moment	2
Additional orientation moment	2

Table 1. Subdomain

When the student requires launching of a lesson around a chosen theme, the system checks the availability of the appropriate e-lesson from the Lesson DB. Lessons that meet the initial user requirements are usually more than one, so the system should provide an appropriate mechanism for selecting the most appropriate among them. After a dialogue with the student, the personal agent defines his personal aims, preferences, etc., and transmits this vector to the system for choices. After the comparison with the vectors of the uploaded e-lessons in the Lesson DB, the e-Lessons with the highest level of similarity are extracted. The result will be a number of e-lessons andthe system should choose the most appropriate. This selection can be realized by the use of some intelligent algorithm (ex. CBR-approach).

The preferences and personal goals of each student can also formalize the policy which defines the sequence of actions in this training scenario. After the identification of the student in the training environment, based on the profile and persona-stereotypical information and a dialogue with his personal agent, the system receives the necessary initial values of the observed variables. After determining the initial state, the policy management can be transferred to a special Policy-Engine, which is part of the infrastructure of the run-time environment of the educational e-Learning portal. Initially, based on the dialogue with the student, the Policy of Preferences registers in the Policy-Engine and then starts the Mechanism for Selecting of Lesson that makes a request to the Lesson DB. After the selection a particular lesson, this e-lesson is filed to SCORM-Learning Management System for implementation. The scenario, which will run activities in the learning process, are described and formalized in the SCORM Sequence & Navigation-model and the corresponding parameterized template by which is created this lesson. Policy-Engine can continually modify policies according to the information coming from the behaviour of the learner.

The learning scenario may include mandatory implementation actions (e.g. solving tests). If a student fails to successfully complete these actions, the learning process falls in a critical condition and the Policy-Engine has to choose more appropriate lessons. In this case, the learning process is temporarily interrupted and the LMS restarts the training process with the new lesson.

http://dx.doi.org/10.5772/61084

The Policy of Preferences is expressed by the rules of condition-action types. Conditions present a number of behaviors that trigger certain actions. The formal semantics of the model is based on ITL as the rules are the following:

when B [increase | decrease] preference in Lesson [low | medium | high], where B is behaviour and Lesson is the e-lesson.

The degree of preference can be expressed as an integer. The larger number represents a higher degree of preference. It is initially assumed that the student doesn't have any preferences and all values are 0. We define the meaning of low, medium, and high level of preferences as 1, 2, and 3. We will look at an example of training with two lessons on the same learning material. The first lesson is more difficult and presents the studying concepts in a higher level than the second one. The student initially has not decided what his preferences are. In the Policy-Engine, there are defined policies, which specify that the lessons that guarantee more than 70% results in the final test are preferable than those that only guarantee between 50% to 70% and the lessons that ensure less than 50% are not preferred. We can express the policy with the following rules:

Score (Lesson1, Lesson2):

When (1: test_result >= 70%) increase preference in Lesson to high

When (2: 50%<=test_result <70%) decrease preference in Lesson to medium

When (3:test_result<50) decrease preference in Lesson to low

The Policy-Engine determines the information needed for the implementation according to these rules. LMS through the SCORM RTE[6] and the mechanism of the target variables (objectives) determines the outcome of the student in solving the test. After each experience, the Policy-Engine checks the assumption as defined by the rules and determines whether they are appropriate. Let’s assume that the student has an aim to study the learning material at a high level (3). The system starts Lesson1 and the results of the three consecutive attempts to resolve the final test are 55, 49%, and 60%. After the first attempt to solve the final test, the Policy-Engine activates the second rule because the result is between 50% and 70%. This determines the preference in 2. The next value is <50%. According to rule three, the Policy-Engine reduces the preference from 2 to 1. The last attempt to solve the test starts again with rule two and increases the preference from 1 to 2. This result is unsatisfactory for the personal aims of the student and as a result, the Policy-Engine defines the lesson as inappropriate. The learning process suspends the former lesson and continues with Lesson 2. The student's results from solving the final test for this lesson are 64%, 68%, and 72%. At the first attempt, the preferences rise to 2, the second is retained the same level, while the third attempt increases it to 3, which is quite satisfactory for the student’s personal aims that the student has set.

The dynamic adaptive levels most directly correspond to the third type of interaction – Open Experience – because the communication is dynamic with continuous engagement between the system and student.

6 RTE- Run-Time Environment

4. Personalization and user modeling

User modelling is an important feature of any e-learning system, to personalize and tailor the e-Learning to individual characteristics, knowledge, didactic aims, and the preferences of the students [19], [20], [21]. On the basis of the previous section, we can describe the adaptability of the system for e-Learning to knowledge and the preferences of students in elementary, static, and dynamic levels [22].

The Elementary Adaptive Level is guaranteed by the profile information about the student before starting his training process in the system. Based only on this adaptive level, the e-learning system offers only learning resources that are common to all students of the same grade and form of training.

The Static Adaptive Level is based on the model for selecting the most appropriate lesson from the Lessons DB as the student is joined to a particular persona in the stereotypical hierarchy. By personas, students with similar characteristics are presented in the e-learning system together. The lessons are prebuilt by the parameterization of the basic BPG-templates and models from the authors of e-learning content in a special authoring environment. These lessons are placed in a special repository – Lesson DB – and are described in metadata as described above.

The Dynamic Adaptive Level is implemented through the Policies of Preferences and Policy-Engine, which dynamically monitors the behaviours of student and his preferences with the relevant lesson in the actual learning process and can replace the current lesson with another that is more appropriate for the individual student.

In the adaptation process in terms of the user modelling, we will look at: the information athering about the learner, processing the information and its update, and finding and presenting the appropriate training resources for the considered student. The model describes the notion of the e-learning system for user knowledge, for his preferences, and aims. This model must be continuously updated according to the dynamic changes in the process of accumulation of knowledge about the particular student (Figure 4). The algorithm includes the following steps:

- Step 1. Filling the static profile information. According to the grade and form of training, the student is associated to any persona in a stereotypical hierarchy. The initial parameters are filled in interactive mode or the system gets the default values from the general stereotype model. Stereotyping and personas are used to transfer more general information about the group in the assumption of the individual user.

- Step 2. According to the persona, which is associated with the student, the system determines the common characteristics of the group and includes default values. Then, in the dialogue mode,the school subject, topic, and personal didactic aims of the student are determined. The rules are updated on the basis of collected information. The Policy-Engine launches the Searching Mechanism for the more appropriate lesson from the Lesson DB and submits it to the LMS for implementation.

- Step 3. The system manages individualized learning process. If there are any discrepancy found between personal aims,the knowledge of the student, and the rules, the Policy-Engine interrupts the current training scenario and restarts the Searching Mechanism for a new choice.
- Step 4. The system stores the new values of the parameters and change the rules by which Policy-Engine manages personal learning process of the individual student.

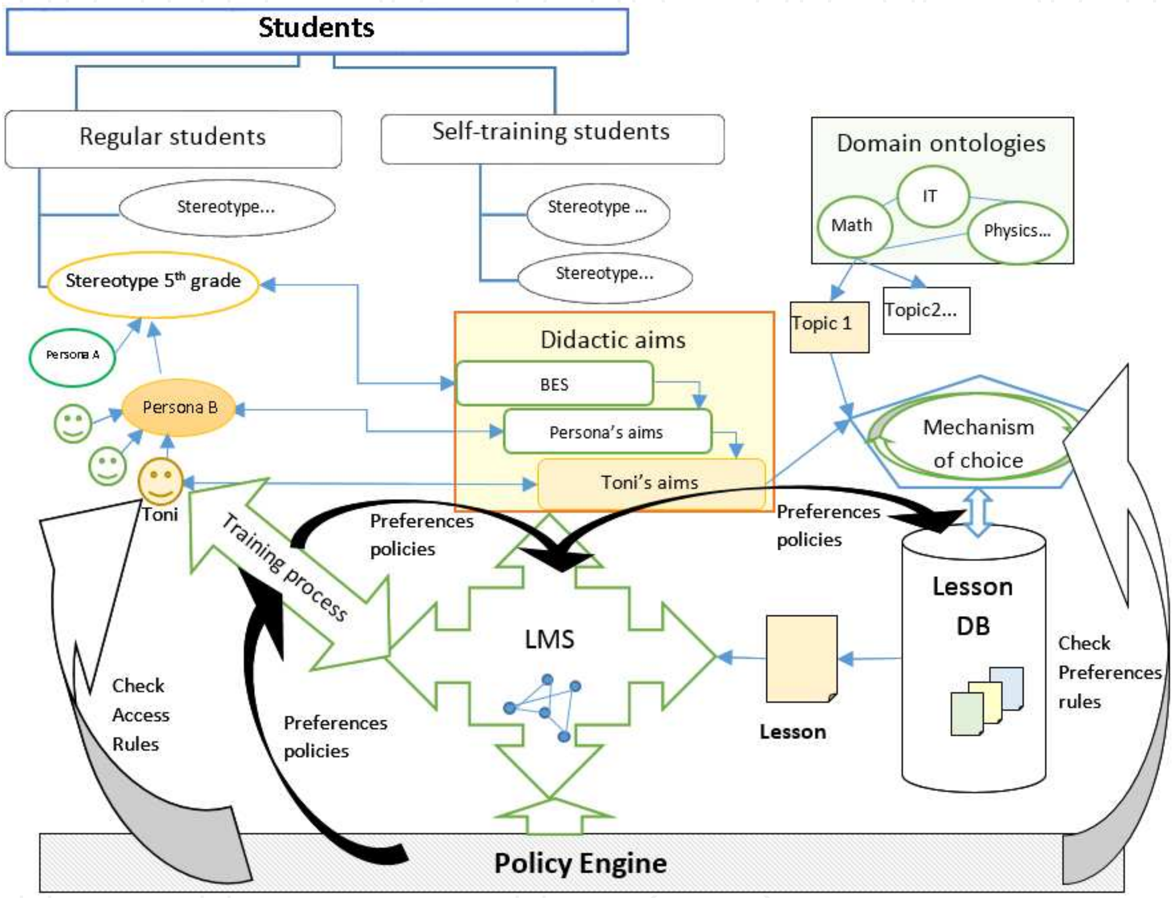

Figure 4. User modelling and personalization

The information in this user model can be considered as information specific to the school subject domain and information that is independent of it. The first type includes the data connected with the Dynamic Adaptive Level as an evaluation of the student; his background knowledge and records of his behavior (number of passed lessons, number of errors during solving test, number of inappropriate lessons, etc.). The information that is not dependent on the some subject domain is related to the personal goals of the learner, with his motivation, experience, preferences, interests, and personal data such as name, years, type of training, etc.

The presented algorithm provides a continuous actualization of information. Such one is independent from the specific school subject domain and one that is domain-dependent. The model is continuously updated to correctly present the student in the e-Learning environment.

We created several versions of SCORM-based e-Learning portal of the secondary school "Hristo Smirnenski"-Brezovo, which is based on the conceptual framework of the system DeLC and supports SCORM RTE [23]. The latest version of the environment ensures the personalization in the elementary and static levels. We developed the mechanism of parameterization of the basic BPG-templates and models, and created an authoring tool for the designing and packaging of SCORM-based e-lessons. Ontologies provide developers with predefined resources covering a specific school subject domain that can be used directly in the content. The establishment of educational environment is based on the adaptation of the corporate portal of the Delphi group. For the realization of the educational portal, we used the portal framework Liferay (http://liferay.com), which has implemented LMS of SCORM RTE [24]. There are many services implemented into the portal that supports the training process in different subjects and raises the level of interactivity in learning.

5. Conclusions and future work

The proposed user model allows to increase the level of personalization in the e-Learning system. This is essential for learners from all forms of training, but is particularly important for students using the distance form of training, pupils with special educational needs, and disabled children. The implementation at the elementary level of the model is provided by the Autho - rules, which depend on stereotyped groups and personas with their access rights to portal resources. Users could be students, teachers, parents, authors of learning content, and so on. If they are students, access must be allowed to educational materials for the appropriate grade, form of training, and so on. If they are teachers, according to their stereotypical information, the mechanism provides them an access to learning resources and services related to their school subjects. If they are authors of educational content, they are allowed access to the Lesson DB for editing and adding of e-Lessons. If they are parents, they are allowed an access to information about their children. Different scenarios for access formalize a sequence of different rules for each group of users managed from the Policy-Engine. The second level of user-modelling is realized through the model for selecting the most appropriate lesson from the repository of lessons – lesson DB and their meta-description by the lesson-vector. The Dynamic Adaptive Level is implemented through the set of Policies of Preferences. The Policy-Engine monitors the behaviour of students and their preference at the relevant lesson and can dynamically replace the current lesson with a more appropriate one.

Based on the MOOC Integrated Learning Environment (http://ilde.upf.edu/handson3), the authors of e-Learning resources successively pass through several steps – the definition of didactic aims, analysis of educational content, creation of personas, designing of learning scenario, creation of e-Lesson in specialized SCORM authoring environment, share a draft version of the created e-lesson for evaluation, and heuristic analysis from other pedagogical specialists. At each step, the authors directly share with their counterparts in the integrated environment. After the completion of the first cycle and depending on the evaluation, authors can then correct their lessons and again pass through the step-by-step cycle or publish the e-lesson in the Lesson DB of education portal. This process is cyclical and leads to the continuous improvement and refinement of the developed learning resources. It meets specified in figure 1 workflow. The published lessons are created in dialogue with educational specialists and tested directly in the real learning environment. This largely ensures the sustainability of the model and overcomes some of the problems in the development of e-Learning.The report is part of the work on the project IT 15- FMIIT-004 "Research in the field of innovative ICT business orientation and training" to fund "Scientific Researches" of Plovdiv University "Paisii Hilendarski".

Author details

Todorka Glushkova*

Address all correspondence to: todorka.glushkova@gmail.com

Plovdiv University “Paisii Hilendarski”, Bulgaria

References

[1] Carliner, S., Patti Shank, editors, The e-Learning handbook: past promises, present challenges. 1st ed. USA: Pfeiffer; 2008. 560 p., ISBN: 978-0-7879-7831-0

[2] Allen, I. Elaine; Seaman, Jeff. Entering the Mainstream: The Quality and Extent of Online Education in the United States, 2003 and 2004. 1st ed. USA: Sloan Consortium (Sloan-C); 2004. 27 p., ISBN:0-9677741-8-7

[3] Abrami, R., Barrett, H., Directions for research and development on electronic portfolios. Canadian Journal of Learning and Technology. 2005; 31(3):1-15, ISSN 1499-6685

[4] Drucker, P., Need to Know: Integrating e-Learning with High Velocity Value Chains. 1st ed. USA: Delphi Group; 2000. 12 p., Delphi Group White Paper

[5] Stoyanov, S., I. Ganchev, I. Popchev, M. O’Droma, From CBT to e-Learning. Information Technologies and Control. 2005;3(4):2-10, ISSN 1312-2622.

[6] Stoyanovich, L.,Staab S., Studer R.,. e- Learning, based on the Semantic Web. In: WebNet2001-World Conference on the WWW and Internet; 23-27.10.2001; Orlando, Florida. USA:2001. p. 23-27, ISBN 1-880094-46-0.

[7] Shedroff, N., A unified field theory of Design. In: Jacobsen R., editor. Information Interaction Design. Cambridge: MIT Press; 1999. p. 267-292.

[8] Glushkova, T., Adaptive Model for E-Learning in Secondary School. In: Elvis Pontes, editor. E-Learning – Long-Distance and Lifelong Perspectives. 1st ed. Croatia: InTech; 2012. p. 3-22, ISBN 978-953-51-0250-2. DOI: 10.5772/29342

[9] Glushkova, T., Adaptive environment for e-Learning in secondary schools [PhD thesis]. Plovdiv University: Plovdiv University; 2011. 220 p. Available from: http://procedures.uni-plovdiv.bg/docs/procedure/163/1820394915693780406.pdf

[10] Lene, N., Personas. In: Rikke Friis, editor. The Encyclopedia of Human-Computer Interaction. 2nd ed. Denmark: The Interaction Design Foundation; 2014. https://www.interaction-design.org/encyclopedia/personas.html.

[11] Mor, Y., Mogilevsky, O., Learning Design Studio: Educational Practice as Design Inquiry of Learning. In: EC-TEL 20013; 17-21 September 2013; Paphos. Berlin: Springer; 2013. p. 233-245.

[12] Glushkova, T., Trendafilova, M.,Uzunova, N., Application of SCORM standard for e-Learning in secondary school. In: Informatics in the Scientific Knowlege- ISK'2006; 18-22 June 2006; Varna. Varna: VFU; 2006. p. 205-216, ISSN 1313-4345, ISBN – 10:954-715-303-X, ISBN-13:978-954-715-303-5.

[13] Stoyanov, S., D. Mitev, I. Minov, T. Glushkova. E-Learning Development Environment for Software Engineering Selbo 2. In: 19th Database and Expert Systems Application (DEXA 2008); 1-5 September 2008; Turin. Springer: 2008. p. 100-104, ISBN: 978-3-540-85653-5. DOI: 10.1109/DEXA.2008.89

[14] Bloom, B., Taxonomy of Educational Objectives Book 1: Cognitive Domain. 2nd ed. Addison Wesley Publishing Company; 1984 (first published June 1956). 204 p., ISBN13: 9780582280106

[15] Learning System Architecture Lab. SCORM Best Practices Guide for Content Developers. 1st ed. Pittsburgh, USA: Carnegie Mellon University; 2003. 80 p.

[16] Mitev, D., Popchev I. Intelligent agents and sevices in eLearning development environment Selbo 2. In: ISK'2008; 26-28 June 2008; Varna. Bulgaria: ISK; 2008. p. 275-284. DOI: ISBN-13:978-954-715-303-5

[17] Sloman, M.. Policy driven management for distributed systems. Journal of network and Systems Management. 1994; 2(3):333-360, ISSN 1064-75-70 (Print), 1573-7705(online).

[18] Moskowski, B. Reasoning about Digital Circuits [Dissertation]. Department of computer science: Stanford University Stanford, CA, USA; 1983. 148 p. Available from: http://dl.acm.org/citation.cfm?id=911281 DOI: AAI8329756

[19] Kobsa, A. User modelling and user-Adapted Interaction. Springer Netherlands. 2004; 14(5):469-475, ISSN: 0924-1868(paper) 1573-1391(online). DOI: 10.1007/s11257-005-2618-3

[20] Glushkova,T., User modelling of distributed e-Learning system for secondary school. In: DIDMATHTECH; 2006; Komarno. Slovakia: DIDMATHTECH; 2006. p. 117-123. ISBN: 978-80-89234-23-3

[21] Brusilovsky, P. Adaptive hypermedia. User Modelling and User Adapted Interaction, Ten Year Anniversary Issue. 2001; 11(1-2):87-110, ISSN: 0924-1868.

[22] Glushkova, T., Stoyanova, A., Interaction and adaptation to the specificity of the subject domains in the system for e-Learning and Distance training DeLC. In: Informatics in the Scientific Knowlege; 17-20 June 2008; Varna. Bulgaria: ISK; 2008. p. 295-307. ISSN: 1313-4345, ISBN: 13:978-954-715-303-526-28

[23] Glushkova, T, Stoyanov, S., Trendafilova, M, Cholakov, G., Adaptation of DeLC system for e_learning in Secondary school. In: CompSysTech'2005; 16-18 June 2005; Varna. Bulgaria: CompSysTech; 2005. p. IV.15.1-15.6. ISBN: 954-9641-38-4

[24] Glushkova, T., E-Learning environment for supporting of secodary school education. Cybernetics and Information Technologies. 2007; 7(3):89-106, ISSN: 1311-9702.

IntechOpen