Stereotype Student Model for an Adaptive e-Learning System

Ani Grubišić, Slavomir Stankov, and Branko Žitko

***Abstract*—**This paper describes a concept of stereotype student model in adaptive knowledge acquisition e-learning system. Defined knowledge stereotypes are based on student's proficiency level and on Bloom's knowledge taxonomy. The teacher module is responsible for the whole adaptivity process: the automatic generation of courseware elements, their dynamic selection and sorting, as well as their adaptive presentation using templates for statements and questions. The adaptation of courseware is realized according to student's knowledge stereotype.

***Keywords*—**Adaptive e-learning systems, adaptive courseware, stereotypes, Bloom's knowledge taxonomy.

I. Introduction

THROUGHOUT the history of the computers, a lot of attention was paid to the idea of using them as intelligent collaborators or personal teachers who will reveal and explain the complicated domain knowledge. An *e-learning* as a term occurred in late 90-ies of last century and it represents the intersection between world of information and communication technology (ICT) and world of education [1].

Teaching approaches and techniques that are directed towards the needs of individual students are called *adaptive instruction* [2]. *Adaptive systems* (AS) are systems that can change their structure, functionality or interface in order to adapt to different needs of individuals or groups, as well as to changes of their needs over time [3]. Adaptive systems and adaptive instruction define a new class of systems called *adaptive educational systems* (AES) or *adaptive e-learning systems* that adapt the process of learning, teaching and testing students' knowledge to different students' characteristics. The two best-known representatives of adaptive e-learning systems are the *intelligent tutoring systems* (ITS) and *adaptive educational hypermedia systems* (AEHS). Adaptive educational hypermedia systems adapt mainly to learning styles. They adapt the presentation to the student's learning style providing the possibility of sequencing the courseware and adapting the content itself [4], [5], [6].

Brusilovsky, although a great supporter and originator of the adaptive hypermedia idea, states that the information about the student, obtained by testing student's knowledge, is more reliable than information gained from making conclusions on the basis of student's navigation [7].

A. Grubišić is with the Faculty of Science, University of Split, Teslina 12, 21000 Split, Croatia (corresponding author phone: ++385 21 385 133; fax: ++385 21 384 086; e-mail: ani.grubisic@pmfst.hr).

S. Stankov is with the Faculty of Science, University of Split, Teslina 12, 21000 Split, Croatia (e-mail: slavomir.stankov@pmfst.hr).

B. Žitko is with the Faculty of Science, University of Split, Teslina 12, 21000 Split, Croatia (e-mail: branko.zitko@pmfst.hr).

The intelligent tutoring systems were supposed to take the lead among the adaptive e-learning systems. However, this did not happen because of their inflexibility and cost of development [8]. Therefore, we have decided to realize adaptation in intelligent tutoring system according to student's knowledge, because it provides an unambiguous and stable foundation sustained by well known taxonomy – a Bloom's knowledge taxonomy [9].

The main assumption underlying the intelligent tutoring systems is that each student is unique. This paradigm is based on creating a student model that remembers the student's preferences and progress in learning and teaching process [10], [11], [12], [13], [14]. Even though all components of the ITSs are important, it is clear that the student model is of the utmost importance. If a student model is "bad" to the extent that it does not even closely describe the student's characteristics, then all the decisions of other ITS components that are based on this model are of poor quality. Therefore, considerable research is carried out in the field of student modeling.

This paper presents an approach to realization of adaptivity in the e-learning systems according to student's knowledge using stereotypes defined on the grounds of the Bloom's knowledge taxonomy. This approach, implemented in a system called Adaptive Courseware Tutor (AC-ware Tutor), has automatic generation, dynamic selection and sequencing of courseware elements, as well as automatic generation of tests and questions [15]. In the second chapter we explain the motivation for this research that presents a continuation of age-long research in the area of the intelligent tutoring systems, as well as some related work in this area. In the third chapter, adaptive knowledge acquisition model is described along with its modules and their functionalities, as well as, the results from the conducted prototype evaluation. In the fourth chapter we conclude with stating the main features of our approach.

II. Background and Related Work

The research described in this paper presents a continuation of lifelong research, development and application of a Tutor-Expert System (TEx-Sys) model for the designing intelligent tutoring systems in any domain knowledge [16], [17].

A methodology for evaluating e-learning systems effectiveness was developed [18], [19]. In a meta-analysis undertaken over the 11 conducted experiments, effect size 0.71 of the xTEx Sys was calculated [20]. The resulting effect size was large (according to [21]), but it is far less than the 2-sigma, the effectiveness of tutoring [33].

The results obtained from these experiments and meta-analysis have shown that it is necessary to make improvements in the TEx-Sys model that would eventually lead to an increase in the effect size. This fact about the necessity of introducing improvements, has directed this research towards finding ways to achieve greater effect sizes. We had to determine where (in which module) and how improvements should be introduced in order to reach the fabulous 2-sigma effectiveness.

Since it is not always possible to realize tutoring in the learning and teaching process, it is impetus to find ways to achieve approximately equal efficiency using different methods and techniques of adaptation in other forms of instruction.

Regardless of which adaptation approach is considered, the following should be taken into account: every student should learn at their pace; adaptation should happen often; each student must successfully complete the learning and teaching process; when something is learned successfully, the student should continue; no one should be taught what they already know [22].

In order to realize a quality adaptation, a student model should be designed with the great care. Student modeling is useless if it has no application. The student model is used for adapting learning and teaching in the selected domain knowledge: a courseware, feedback, help generation and size of the learning element are adapted. In order to realize maximum adaptivity, ITS has to provide the optimal use of information contained in the student model to guide their actions to the specific needs of each student.

One way to model students is using stereotypes. A *stereotype student model* enables distinguishing some typical characteristics of students [23], [24]. Stereotypes represent a collection of characteristics or facet. The student can be described with the set of characteristics that have value. To enable the computer system to effectively use stereotypes, it must know the stereotypes and their triggers. The trigger is tied to a particular situation. It contains the name of the stereotype that it activates and rating (a number between 0 and 1000) which is assigned to the stereotype. Rating expresses the likelihood that the stereotype is suitable for the particular situation.

There is very little information in the student model that can be said with certainty to be true, because stereotype model is based on probability (MYCIN system [25]). Therefore, the student model must have, for each student, truth rating about each piece of information. As a result, stereotypes usually consist of a set of triples (attribute, value, rating). The rating represents the fact that every person belonging to a certain stereotype has a certain characteristics.

The origin of stereotype modeling can be found in a system Grundy [26]. The Grundy is a system that plays the librarian and suggests books to users based on his/her characteristics. He uses stereotypes to cluster standard features of the user's characteristics. A more complex stereotype model can be found in a Unix Consultant (UC) [27], [28]. The UC is a computer system that uses natural language to advice student how to use the UNIX operating system.

Behaviors that should be measured in the adaptive e-learning systems, especially in the intelligent tutoring systems, are those which can be predicted based on the results of specific teaching techniques. Knowledge provides more valid and reliable basis for determining the adaptation results than other preferences or abilities [29]. Therefore, we have decided to realize adaptation in our approach according to student's knowledge, because it provides an unambiguous and stable foundation sustained by well known Bloom's knowledge taxonomy. We use this taxonomy for designing a stereotype student model.

III. Stereotype Student Model and Adaptive Courseware

The new model of adaptive knowledge acquisition takes into account the current level of student's knowledge and their cognitive characteristics that determine the complexity and level of presented courseware elements.

Most ITSs provide learning environments that are based on "free" approach to learning, that is, students choose their own learning path in the courseware. This way they can skip learning certain courseware elements, or learn something in the wrong order, or learn something that is too hard or too easy for them. This is the reason why the ITSs have to guide students in the learning and teaching process and present them just the courseware that has suitable difficulty. Therefore, the ITSs, while generating, selecting, sequencing and presenting learning content to student, should take into account current student knowledge and complexity of the courseware elements. This approach will reduce the cognitive overloading and will allow individualized guidance of learning and teaching process.

The proposed model of adaptive knowledge acquisition is based on automatic and dynamic generation and adaptive selection, sequencing and presentation of the courseware. Automatic courseware generation in the new model indicates that the courseware elements for learning and testing knowledge are created by system itself, based on the domain knowledge ontology. Dynamic courseware generation means that courseware is created in the moment of execution. Adaptive selection, sequencing and presentation of courseware are done automatically and dynamically in accordance with a stereotype student model using templates for statements and question.

In the following subchapters we will describe domain knowledge and adaptive courseware model in a stereotype based intelligent tutoring system and their relation with Bloom's knowledge taxonomy.

A. Ontology Based Domain Knowledge

Nowadays, ontology is commonly used to formalize knowledge in the ITSs [30]. Domain knowledge is presented with concepts and relations between them.

In order to clearly indicate for each relation r in the ontology that it connects concepts K_1 and K_2 and what the nature of that relation is, we define ***domain knowledge (DK)*** as a set of triplets (K_1, r, K_2). In this way we define that the concept K_1 is ***superconcept*** of concept K_2 and that concept K_2 is ***subconcept*** of concept K_1. If triplet (K_1, r, K_2) is in DK, then (K_1, r, K_2) is not in DK.

Since the basic elements of the domain knowledge triples are concepts and relations between them, we use a graph theory as a mathematical foundation for managing subsets and elements of domain knowledge, as well as for domain knowledge visualization. Therefore, we define a directed domain knowledge graph on which all the rules from the graph theory apply.

A directed ***domain knowledge graph DKG*** is equal to a set of ordered pairs of those concepts from the domain knowledge that are related. The vertex from DKG is called a ***root*** if it has no superconcepts and has subconcepts. The vertex from DKG is called a ***leaf*** if it has superconcepts and has no subconcepts. A ***graph unit*** C_i as the largest associated subgraph of DKG whose vertex set contains only one root.

The central root has no superconcepts and it is a starting vertex for every path in the graph. Any vertex in the unit, except the central root, has one or more superconcepts. The graph unit is, therefore, an associated subgraph where at least one path exists between every vertex (except the central root) and the central root. In graph unit there are no isolated vertices. Thus, each root of the domain knowledge defines one graph unit.

B. Student Model and Bloom's Knowledge Taxonomy

People in the same stages of learning have similar knowledge. Once the system has determined the student knowledge level, it can assume that the student knows as much as other students who are at that level. In a good student modeling, assumptions about the student level are grouped in a simple and effective way.

The way of classifying students into categories according to some criterion(s) is called a stereotyping. This method is derived from psychology because it is considered that people use stereotypes (or clusters or features) in order to simplify and classify the complexity of the world they live in [23].

In the new model of adaptive knowledge acquisition we utilize the accustomed pedagogical terminology and renowned the Bloom's knowledge taxonomy in order to define students' stereotype. The common assessment criteria are consistent with the Bloom's knowledge taxonomy in a way that the grade sufficient (or 2 or D) describes the knowledge recall, good (or 3 or C) the knowledge comprehension, very good (or 4 or B) the knowledge application, excellent (or 5 or A) the knowledge analysis, synthesis and evaluation, and insufficient (or 1 or E, F) describes the lack of knowledge.

Students can be grouped according to their grades, which are described in the above manner. We have five grades, which allow us to classify students into five categories. All students in a certain category have in common the fact that they posses the same level of knowledge, that is, the knowledge level is a common feature of all students in the same category. This means, for example, that all students with grade very good (or 4 or B) are characterized by possessing a knowledge on the application level, all students with grade sufficient (or 2 or D) are characterized by a knowledge on the recall level, etc.

In our approach, we use *knowledge stereotypes.* If we know the student stereotype, then we know his/her level of knowledge. Goes the other way around, if we know the student's level of knowledge, then we know the student's stereotype. In this way, we have stereotyped students according to the Bloom's knowledge taxonomy. We have defined five knowledge stereotypes: novice, beginner, intermediate, advanced and expert (Table I). The number of stereotypes corresponds exactly to custom pedagogical practices that assess students in a range from 1 to 5 (or F, E to A).

In this way, all students who do not have any knowledge (besides some recall knowledge) are considered *novices*, those who have recall knowledge (and maybe some comprehension knowledge) are considered *beginners*, those who have recall and comprehension knowledge (and maybe some application knowledge) are considered *average*, those who have recall, comprehension and application knowledge (and maybe some analysis, synthesis and evaluation knowledge) are considered *advanced*, while those who can recall, comprehend, apply and analyze, synthesize and evaluate knowledge are considered *experts*.

C. Adaptive Courseware Model

The goal of student modeling is to determine the student's overall competencies in specific domain knowledge. The student modeling is also used for adapting the behavior of the intelligent tutoring systems to optimize learning and teaching process in the selected domain knowledge. Since, one of the adaptation techniques used in ITSs is courseware sequencing, we have decided to use it in our approach.

The courseware elements are units (U), modules (M) and lessons (L), as well as tests (T). The units, modules and lessons are the courseware elements intended for learning, and tests are used to verify the student's knowledge. A ***courseware CW*** is an array of courseware elements.

TABLE I
THE KNOWLEDGE STEREOTYPES

Grade	*Stereotype*	*Recall*	*Comprehension*	*Application*	*Analysis, synthesis and evaluation*
1/E,F	Novice	maybe	none	none	none
2/D	Beginner	mostly	maybe	none	none
3/C	Intermediate	mostly	mostly	maybe	none
4/B	Advanced	mostly	mostly	mostly	maybe
5/A	Expert	mostly	mostly	mostly	mostly

A ***learning courseware element LCE*** is an associated subgraph (V'_{LCE}, A'_{LCE}) of the domain knowledge graph. For each learning courseware element it is only required to define a concept K_x that will be the root of that subgraph. A tag $\mathbf{LCE_{Kx}}$ denotes learning courseware element whose root is concept K_x (LCE can be replaced with U for unit, M for module and L for lesson).

A ***learning courseware element rank*** R_{LCEKx} is ordered triplet **(u, p, l)**, where **u** denotes graph unit C_i whose subgraph is LCE_{Kx}, **p** is length of path from the central root to K_x and **l** denotes the level of courseware element (1-unit, 2-module, 3-lesson).

A ***testing courseware element T*** is a sequence of questions which test students' knowledge about the relations between concepts. It is uniquely determined by learning courseware elements that precede it in the courseware CW, as well as, by the current student stereotype.

1) Automatic Generation of the Learning Courseware Elements

Automatic courseware generation in the new model indicates that the courseware elements for learning and testing knowledge are created by system itself, based on the domain knowledge ontology. First all units have to be generated, then for every unit its modules have to be generated, and finally, for each module its lessons have to be generated.

For each level of courseware element there is a specific generation algorithm (see Fig. 1).

2) Automatic and Adaptive Knowledge Assessment

The biggest problem that exists for the Computer Adaptive Tests (CAT) is requirement for the existence of a database with an enormous number of questions of different difficulty defined by teacher [31]. A much better solution is to automatically and dynamically generate questions and tests based on the domain knowledge concepts where questions are defined by adaptive e-learning system itself. Such tests are also adapting themselves to the student's knowledge. This kind of knowledge testing is used in the AC-ware Tutor model.

Testing courseware elements contain questions generated over one subset of domain knowledge. Since knowledge is tested iteratively in the learning and teaching process, for generating testing courseware element it is only necessary to know over which subset of domain knowledge its questions are generated, and the current stereotype of student.

In our approach, questions are generated using templates. Those templates have four difficulty levels closely related to stereotypes and the Bloom's taxonomy. Each difficulty level examines a certain knowledge level: the first level contains templates that test knowledge recall, the second level contains templates that test knowledge comprehension, the third level contains templates that test knowledge application and the fourth level contains templates that test knowledge analysis, synthesis and evaluation. Templates enable generation of objective test items [32]: completion, short-answer, single choice, matching, ranking, multiple choice (Table II).

Each question examines knowledge about relations between concepts, as well as, knowledge about the concepts themselves. Shown knowledge is scored on a scale from 0 to 4 points. Regardless of the question, if the answer is "I don't know", then the score is 0 points. Incorrect answer scores also 0 points, while the correct answer scores as many points as the question difficulty level is.

1. Select the root K_x of **unit** U_{Kx}.
2. Set of vertices V' of **module** M_{Kx} is made of root K_x and its subconcepts that belong to the same unit.
3. If a subconcept K_y of K_x has only one subconcept K_z, then both K_y and K_z, as well as subconcepts of K_z, belong to set V' (only if they belong to the same unit).
4. If a subconcept K_y of K_x has more than one subconcept, then K_y and all its subconcepts belong to set V' (only if they belong to the same unit)
5. If we cannot find any concepts in steps 3 and 4, then module cannot be generated. If K_x has subconcepts, then a **lesson** whose set of vertices is made of K_x, its subconcepts and their subconcepts (if they exist) can be generated.
6. For each leaf K_w in **module** M_{Kx}, check if a **module** M_{Kw} with a root K_w can be generated (such as described in steps 2, 3 and 4).
7. Repeat step 6 for each generated module.

Fig. 1 An algorithm for generating modules

TABLE II
KNOWLEDGE LEVELS, STEREOTYPES AND QUESTION TEMPLATES

Stereotype	***Question template***
Novice	-
Beginner	Are $\boldsymbol{K_x}$ and $\boldsymbol{K_y}$ directly connected? Are $\boldsymbol{K_x}$ and $\boldsymbol{K_y}$direcly connected with $\boldsymbol{r}$? Is $\boldsymbol{K_x}$ subconcept of $\boldsymbol{K_y}$? Is $\boldsymbol{K_x}$ superconcept of $\boldsymbol{K_y}$? What directly connects $\boldsymbol{K_x}$ and $\boldsymbol{K_y}$?
Intermediate	Are $\boldsymbol{K_x}$ and $\boldsymbol{K_y}$ indirectly connected? What is $\boldsymbol{K_x}$? Which subconcept is directly connected by $\boldsymbol{r}$ with $\boldsymbol{K_x}$? Which superconcept is directly connected by $\boldsymbol{r}$ with $\boldsymbol{K_x}$?
Advanced	$\boldsymbol{K_x}$is ____________ of $\boldsymbol{K_y}$ How many concepts is $\boldsymbol{K_x}$ connected with? How many subconcepts does $\boldsymbol{K_x}$ have? How many superconcepts does $\boldsymbol{K_x}$ have? Connect given concepts
Expert	$\boldsymbol{K_x}$ and $\boldsymbol{K_y}$ are directly connected by ____ Sort concepts: Is $\boldsymbol{K_{slot}}$ of $\boldsymbol{K_x}$ $\boldsymbol{K_{filler}}$? What is $\boldsymbol{K_{slot}}$ of $\boldsymbol{K_x}$? Whose $\boldsymbol{K_{slot}}$ is $\boldsymbol{K_{filler}}$?

We define two weight functions on the domain knowledge graph whose values are determined after each knowledge test. The **first** edges-function $\mathbf{X_A}$*:A'*$\rightarrow${-1,0,1,2,3,4} defined by, *$X_A(K_xK_y)$=score obtained by answering a question that applies to edge K_xK_y*, $\forall K_xK_y \in A'$, is a weight function on a set of edges of the domain knowledge graph.

When the student model initializes, all edges in the domain knowledge graph have the weight -1, which means that the knowledge about the relationship between those two concepts has not been tested yet. The function X_A allows assigning weights to those edges that connect the concepts mentioned in certain question. Each edge from *A'* has the weight between 0 and 4. Thus, domain knowledge graph with weight function X_A becomes edge-weighted graph where the weighting function values change after each knowledge test.

Based on the values of edge-weighted function X_A, the values of the **second** vertex-weighted function X_V are determined. The function $\mathbf{X_V}$*:$V_{DKG}\rightarrow[0,1]$* defined by $\forall K_x \in V_{DKG}$, $\forall K_xK_{yi} \in A_{DKG}$, $X_A(K_xK_{yi}) \neq -1$, $\forall K_{yi}K_x \in A_{DKG}$, $X_A(K_{yi}K_x) \neq -1$

$$X_V(K_x) = \left(\frac{\sum_{i=1}^{k} X_A(K_x K_{y_i})}{4 * subK_x} + \frac{\sum_{i=1}^{k} X_A(K_{y_i} K_x)}{4 * \sup K_x} \right) * \frac{1}{2},$$
$$subK_x \neq 0, \sup K_x \neq 0 \quad (1)$$
$$X_V(K_x) = \frac{\sum_{i=1}^{k} X_A(K_x K_{y_i})}{4 * subK_x}, subK_x \neq 0, \sup K_x = 0$$
$$X_V(K_x) = \frac{\sum_{i=1}^{k} X_A(K_{y_i} K_x)}{4 * \sup K_x}, subK_x = 0, \sup K_x \neq 0$$
$$X_V(K_x) = 0, subK_x = 0, \sup K_x = 0$$

is a weight function on a set of vertices of the domain knowledge graph.

The value $X_V(K_x)$ represents the weighted sum of values of the function X_A on the edges towards superconcepts of the concept K_x and towards subconcepts of concept K_x. We believe that the student knows the concept K_x completely if and only if $X_V(K_x)$=1, what is true only if all the values of the function X_A on the edges towards superconcepts and subconcepts are 4. Then the probability of knowing the concept is the highest, that is, equal 1.

3) Knowledge Stereotypes Determination

The results of testing knowledge, that is, the values of weight functions, allow defining the following subsets of domain knowledge concepts:

1. $Z_{Level0}=\{K_x \in E_{CON}, X_V(K_x)<0,2\}$ is a subset of domain knowledge concepts that the student does not know ***(L=0)***.
2. $Z_{Level1}=\{K_x \in E_{CON}, 0,2 \le X_V(K_x)<0,4\}$ is a subset of domain knowledge concepts that the student knows at ***level 1 (L=1)*** – the knowledge on the recall level.
3. $Z_{Level2}=\{K_x \in E_{CON}, 0,4 \le X_V(K_x)<0,6\}$ is a subset of domain knowledge concepts that the student knows at ***level 2 (L=2)*** - the knowledge on the comprehension level.
4. $Z_{Level3}=\{K_x \in E_{CON}, 0,6 \le X_V(K_x)<0,8\}$ is a subset of domain knowledge concepts that the student knows the ***level 3 (L=3)*** - the knowledge on the application level.
5. $Z_{Level4}=\{K_x \in E_{CON}, 0,8 \le X_V(K_x) \le 1\}$ is a subset of domain knowledge concepts that the student knows at ***level 4 (L=4)*** - the knowledge on the analysis, synthesis and evaluation level.

The definition of the function X_V and previously mentioned subsets, have allowed us to define *stereotypes according to knowledge*. Therefore, at the lowest level of formalization we assign stereotypes to students as follows:

1. A student is a **novice** if after the knowledge test, the majority of concepts from his/her student model are in the set Z_{Level0}. He/she has knowledge at level 0 ***(L=0)***,
2. a student is a **beginner** if after the knowledge test, the majority of concepts from his/her student model are in the set Z_{Level1}. He/she has knowledge at level 1 ***(L=1)***,
3. a student is an **intermediate** if after the knowledge test, the majority of concepts from his/her student model are in the set Z_{Level2}. He/she has knowledge at level 2 ***(L=2)***,
4. a student is an **advanced** if after the knowledge test, the majority of concepts from his/her student model are in the set Z_{Level3}. He/she has knowledge at level 3 ***(L=3)***,
5. a student is an **expert** if after the knowledge test, the majority of concepts from his/her student model are in the set Z_{Level4}. He/she has knowledge at level 4 ***(L=4)***.

4) Adaptive Selection, Sequencing and Presentation of the Courseware Elements

Once we have defined stereotypes and the way we assign them to students, it is necessary to define how to select, sequence and present the courseware elements to students according to their stereotype.

A ***selection*** of the courseware elements is done according to the level of courseware elements. Courseware element level determines the maximum amount of knowledge that the student will learn in a single learning and teaching cycle. Since we have defined five stereotypes of students (novice, beginner, moderate, advanced and expert) and we have only three levels of courseware elements (units, modules, lessons), some of the stereotypes will learn the courseware elements with the same level (Table III).

TABLE III
STEREOTYPE, COURSEWARE ELEMENT LEVEL AND KNOWLEDGE LEVEL

Stereotype	***Maximum courseware element level***	***Knowledge level***
Novice	Lesson	Recall
Beginner	Lesson	Comprehension
Intermediate	Module	Comprehension
Advanced	Module	Application
Expert	Unit	Analysis, synthesis and evaluation

TABLE IV
RULES FOR ADDING TESTING COURSEWARE ELEMENTS FOR PARTICULAR STEREOTYPE

Stereotype	***Rules for adding testing courseware elements***
Novice	- when the label **u** of the graph unit changes, - when the length **p** of path from the central root to the courseware element root changes (but only if there would be at least 2 lessons before and at least 2 lessons after test), - after three lessons(but only if there would be at least 2 lessons after test) - the last courseware element has to be test

Once all learning courseware elements with a certain level are selected for a particular stereotype, ***sequencing*** of those courseware elements is, in fact, adding sorted elements in courseware. Courseware elements are ***sorted*** according to their rank **(u, p, l)** and name: *firstly* by label of the graph unit **u**, *secondly* ascending by length of path from central root to root of the courseware element **p**, *thirdly* by level of courseware element **l**, and *finally* alphabetically by courseware element root name.

Testing courseware elements are added in the courseware according to rules that depend on particular stereotype. All stereotypes have in common that the last element of courseware has to be a test. Considering the courseware element level that has been assigned to particular stereotype, we suggest the rules for adding testing courseware elements. A set of rules for stereotype novice can be seen in Table IV.

After selecting and sequencing the courseware elements that is adapted to student's stereotype, we need to define a way of ***presenting*** the courseware elements which is, also, adapted to stereotypes. The courseware elements presentation is done according to the Bloom's taxonomy. Namely, each stereotype is presented appropriate knowledge level (Table III). We see that the stereotypes intermediate, advanced and expert knowledge are presented the knowledge of the same level, while the stereotypes novice and beginner are presented the knowledge at a higher level, in order to enable much faster progress to intermediate stereotype.

The way in which the knowledge will be presented to student in the learning and teaching process is defined by templates for generating statements. Since we have defined four questions difficulty categories, which correspond to the Bloom's knowledge taxonomy, we define the corresponding four statements difficulty categories for learning and teaching. Each statement category presents knowledge at a certain level: the first statement category contains templates that present knowledge on the recall level, the second on the comprehension level, the third on application level and the fourth one on the analysis, synthesis and evaluation level (Table V).

TABLE V
KNOWLEDGE LEVELS, STEREOTYPES AND QUESTION TEMPLATES

Stereotype	***Statement template***
Novice	K_x **r** K_y. K_x is subconcept of K_y. K_x **r** K_y. K_x is superconcept of K_y.
Beginner Intermediate	K_x and K_y are indirectly connected because there is **Path**K_xK_y. K_x **r:** $K_{y1},\ldots, K_{yn}$. These are the subconcepts of K_x. $K_{y1},\ldots,K_{yn}$ **r** K_x . These are the superconcepts of K_x. K_y **r** K_x. K_x has superconcept K_y. K_x **r** K_y. K_x has subconcept K_y.
Advanced	K_x is directly connected with ***sub****K_x* + ***super****K_x* concepts. These concepts are: $K_1,\ldots,K_n$. K_x has ***sub****K_x* subconcepts. $K_x\mathbf{r}_1$ $K_{y11},\ldots,K_{yn1},\ldots, K_x\mathbf{r}_m$ $K_{y1m},\ldots,K_{ynm}$. K_x has ***super****K_x* superconcepts. $K_{y11},\ldots,K_{yn1}\mathbf{r}_1K_x ,\ldots,$ $K_{y1m},\ldots,K_{ynm}\mathbf{r}_mK_x$.
Expert	The concepts are directly connected in the following way: $K_x\mathbf{r}_1,\ldots, \mathbf{r}_n, K_{n+1}$. K_{slot} of K_x is K_{filler}.

D. Student Model Initialization

The testing courseware elements are used for determining student's stereotype. The problem is how to determine the initial stereotype according to which student will be presented his/her first set of courseware elements. This problem is solved by generating the initial test over a specific subset of domain knowledge.

The first set of questions in initial test is generated based on the question templates from the third difficulty category (***L=3***). The concepts, that are included in the questions that the student has answered incorrectly, become an input for generation of questions based on templates from the second difficulty category (***L=2***) and the concepts, that are included in the questions that the student has answered correctly, become an input for generation of questions based on templates from the fourth difficulty category (***L=4***). The concepts, that are included in the questions based on templates from the second difficulty category that were answered incorrectly, become an input for generation of questions based on templates from the first difficulty category (***L=1***). Therefore, the initial test has minimum two and maximum three iterations.

E. Adaptive Learning Cycles

The whole process of learning and teaching in the presented model that supports automatic and dynamic generation and adaptive selection, sequencing and presentation of the courseware based on test results and adapted to student stereotype assigned according to test results, consists of three-phase cycles: *Learning and teaching adapted to stereotype* → *Knowledge testing* → *Determining stereotype.*

Entrance to the first cycle of learning and teaching process is the initial student stereotype defined on the basis of the initial test. Each cycle ends with determining new student stereotype (or keeping the previously assigned one). In each cycle, the student is adaptively presented only one part of the domain knowledge according to his/her stereotype assigned in the previous cycle. Then a knowledge test of the concepts previously presented is conducted, which leads to changing the student model (the values of weight functions that determine it change based on the test results). After that knowledge test, the new stereotype is determined (it is desirable that it changes towards the expert, but it can also stagnate). The learning and teaching process ends when the student shows knowledge of concepts, at least at level 1. The output from the learning and teaching process is student's final stereotype (Fig. 2).

Note that the student cannot complete the learning and teaching process as a novice. If a student wishes to complete the learning and teaching process with better stereotype, then he/she re-enters the learning and teaching process with initial stereotype equal to the his/her final stereotype. It is desirable that student, during learning and teaching process, reaches the expert stereotype and finishes the learning and teaching process as expert.

F. Evaluation

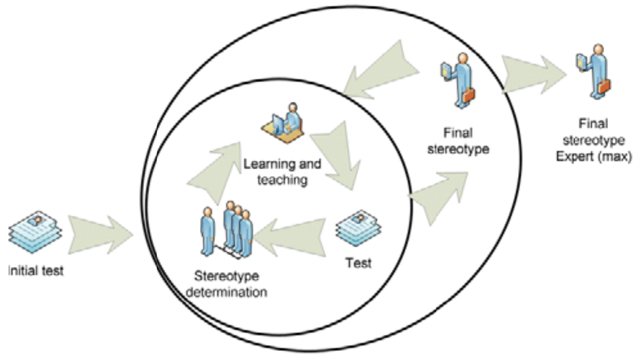

Fig. 2 Learning cycles in the Adaptive Courseware Tutor

The model AC-ware Tutor that uses stereotype student model was implemented as a prototype version. The prototype uses domain knowledge "Computer as a System".

This prototype was tested in a research that was conducted with 30 students. The main objective of the pilot testing was to establish a sense of "satisfaction" while using the AC-Ware Tutor and to see if the student were assigned a appropriate stereotype.

The learning and teaching in the AC-ware Tutor lasted for two hours after which the students filled a questionnaire about "satisfaction" while working with the system.

The *first* group consisted of 14 students from the first year of undergraduate study at the Faculty of Science, University of Split. Students in this group had no prior knowledge about the e-learning systems, but they have had a very good knowledge related to the domain knowledge "Computer as a System" that was presented in the AC-ware Tutor.

The *second* group consisted of 16 students from the first year of graduate studies at the Faculty of Science, University of Split and the Faculty of Science, Mathematics and Science Education, University of Mostar. Students in this group have just been learning about the different e-learning systems.

Interpretation of the results of the questionnaire in pilot testing shows that students have understood the course content and that they believe that the usage of the AC-ware Tutor would have positive impact on the quality of traditional teaching.

IV. Conclusion

The idea of a new model of an intelligent tutoring system is based on traditional architecture, but with a substantial improvements that are associated with adapting the whole process of learning, teaching and testing to current level of student's knowledge. In this regard, adaptation to student's knowledge is achieved by applying stereotypes and the Bloom's knowledge taxonomy.

The students' stereotypes are associated with the current level of knowledge, based on the Bloom's knowledge taxonomy and structured according to the knowledge scale: novice, beginner, intermediate, advanced and expert.

The selection of courseware elements is performed according to the courseware element level. The courseware element level determines the amount of knowledge that the student will learn in a single cycle learning and teaching. Given that we have defined five stereotypes of students (novice, beginner, moderate, advanced and expert), and we have only three elements levels (units, modules, lessons), some of the stereotypes will be presented the courseware elements with the same level.

Once all learning courseware elements of a certain level are selected for a particular stereotype, sequencing of those courseware elements is, in fact, adding sorted elements in courseware, as well as, adding testing courseware elements according to rules that depend on particular stereotype.

Courseware elements presentation is done according to the Bloom's taxonomy. Namely, each stereotype is presented appropriate knowledge level. The way in which knowledge will be presented to student is defined by statement templates. There are four statements difficulty categories that correspond to a certain knowledge level.

Testing courseware elements contain certain number of questions generated using templates. Those templates have four difficulty levels closely related to stereotypes. Each difficulty level examines a certain knowledge level, according to the Bloom's taxonomy. Templates enable generation of objective test items. In our approach the tests and questions are automatically and dynamically generated for each student separately, and therefore are not repeated.

Prototype tests that were conducted with two groups of students, besides the positive comments, gave an interesting critique of the prototype implementation of the AC-ware Tutor, what has a stimulatory effect on further research. In this sense, the system needs a graphical component that would facilitate the adaptation of courseware. Furthermore, since the ontology allows multilingual naming of concepts and relations, translation of templates for generating statements and questions would enable creating multi-language version of the AC-ware Tutor.

Acknowledgment

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

References

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

[2] O. Park and J. Lee, "Adaptive instructional systems," Handbook of research for educational communications and technology, pp. 634–664, 1996.

[3] D. Benyon and D. Murray, "Adaptive systems: From intelligent tutoring to autonomous agents," Knowledge-Based Systems, vol. 6, no. 4, pp. 197–219, 1993.

[4] C. Ullrich, "Descriptive and Prescriptive Learning Theories," Pedagogically Founded Courseware Generation for Web-Based Learning, pp. 37–42, 2008.

[5] B. Bontchev and D. Vassileva, "Adaptive courseware design based on learner character", pp. 23–25. 2009

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

[7] P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia," User Modeling and User Adapted Interaction, vol. 6, no. 2–3, pp. 87–129, 1996.

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

[9] B. S. Bloom, Taxonomy of educational objectives. The classification of educational goals, Handbook I Cognitive Domain. Green, New York, NY: Committee of College and University Examiners, Longmans, 1956.

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

[11] H. L. Burns and C. G. Capps, "Foundations of intelligent tutoring systems: An introduction," in Foundations of intelligent tutoring systems, M. C. Poison, J. J.Richardson (Ed.), Lawrence Eribaum, London, pp. 1–19., 1988.

[12] J. W. Rickel, "Intelligent computer-aided instruction: A survey organized around system components," Systems, Man and Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, no. 1, pp. 40–57, 1989.
[13] B. Woolf, "AI in Education. Encyclopedia of Artificial Intelligence," New York, Wiley, pp. 434–444, 1992.
[14] J. Self, "Formal approaches to student modelling," Student modelling: The key to individualized knowledge-based instruction, vol. 25, 1994.
[15] A. Grubišić, "Adaptive student's knowledge acquisition model in e-learning systems," PhD Thesis, University of Zagreb, Croatia, Faculty of Electrical Engineering and Computing, 2012.
[16] S. Stankov, "Isomorphic Model of the System as the Basis of Teaching Control Principles in an Intelligent Tutoring System," PhD Thesis, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Croatia (in Croatian), 1997.
[17] S. Stankov, M. Rosic, B. Zitko, and A. Grubisic, "TEx-Sys model for building intelligent tutoring systems," Computers & Education, vol. 51(3), pp. 1017–1036, 2008.
[18] A. Grubišic, S. Stankov, and B. Žitko, "An approach to automatic evaluation of educational influence," in Proceedings of the 6th WSEAS International Conference on Distance Learning and Web Engineering, Stevens Point, Wisconsin, USA, pp. 20–25, 2006.
[19] A. Grubišić, "Evaluating effectiveness of intelligent e-learning systems," Master's thesis, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia (in Croatian), 2007.
[20] A. Grubišić, "A Meta-Analytic Estimation of a Common Effect Size from a Series of Experiments Related to an E-Learning System Effectiveness Evaluation," Intelligent Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation, p. 327, 2010.
[21] J. Cohen, Statistical power analysis for the behavioral sciences. Lawrence Erlbaum Associates, 1988.
[22] A. M. Riad, H. K. El-Minir, and H. A. El-Ghareeb, "Review of e-Learning Systems Convergence from Traditional Systems to Services based Adaptive and Intelligent Systems," Journal of Convergence Information Technology, vol. 4, no. 2, 2009.
[23] E. Rich, "Users are individuals: individualizing user models," International journal of man-machine studies, vol. 18, no. 3, pp. 199–214, 1983.
[24] I. Beumont and P. Brusilousky, "Adaptive Educational Hypermedia", pp. 93–98., 1995.
[25] E. H. Shortliffe, R. Davis, S. G. Axline, B. G. Buchanan, C. C. Green, and S. N. Cohen, "Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system," Computers and Biomedical Research, vol. 8(4), pp. 303–320, 1975.
[26] E. Rich, "User modeling via stereotypes," Cognitive Science: A Multidisciplinary Journal, vol. 3(4), pp. 329–354, 1979.
[27] R. Wilensky, Y. Arens, and D. Chin, "Talking to UNIX in English: an overview of UC," Communications of the ACM, vol. 27, no. 6, pp. 574–593, 1984.
[28] R. Wilensky, D. N. Chin, M. Luria, J. Martin, J. Mayfield, and D. Wu, "The Berkeley UNIX consultant project," Computational Linguistics, vol. 14(4), pp. 35–84, 1988.
[29] R. Glaser and A. J. Nitko, "Measurement in Learning and Instruction.," 1970.
[30] T. B. Lee, J. Hendler, and O. Lassila, "The semantic web," Scientific American, vol. 284, no. 5, pp. 34–43, 2001.
[31] R. Conejo, E. Guzmán, E. Millán, M. Trella, J. L. Pérez-De-La-Cruz, and A. Ríos, "Siette: a web-based tool for adaptive testing," International Journal of Artificial Intelligence in Education, vol. 14, no. 1, pp. 29–61, 2004.
[32] N. E. Gronlund, Measurement and Evaluation in Teaching, 5th Revised ed. New York: Macmillan Publishing Company, 1985.
[33] B.S., Bloom (1984) The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. Educational Researcher, 13, pp. 4-16.

Ani Grubišić was born in Split, Croatia on 10th November 1978. She gained a BA for mathematics and computer science teacher at the Faculty of Science, University of Split, Croatia in 2001. The theme of her graduate thesis was "Model tracing – a diagnostic technique of intelligent tutoring systems". She gained a MS degree in computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia in 2007. The topic of her MS thesis was "Evaluating educational influence of intelligent e-learning systems". She gained a PhD degree in computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia in 2012. The topic of her PhD thesis was "Adaptive student's knowledge aquisition model in e-learning systems". Since 2002 she has been employed as an Assistant Professor at the Department of Computer Science, Faculty of Science, University of Split, Croatia. Her research interest includes intelligent tutoring systems, adaptive courseware, and effectiveness evaluation.

Slavomir Stankov was born in Risan, Montenegro on 29th August 1947. Today he is a full professor at the Department of Computer Science, Faculty of Science, University of Split, Croatia. In 1974 he enrolled post graduate study of electronics at Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture (FESB), University of Split, where he got his MS degree in 1979 with theses "An approach to projecting a system for processor control". In 1997 at the FESB, University of Split he defended his PhD theses titled "Isomorphic Model of the System as the Basis of Teaching Control Principles in an Intelligent Tutoring System". His main research interests are in the areas related to the information and communication technology applications in education, in particular the intelligent tutoring systems and authoring shells for their design, implementation and deployment. He has authored or co-authored more than seventy scientific and expert papers.

Branko Žitko was born in Split, Croatia on 8th February 1975. He gained a BA for mathematics and computer science teacher at the Faculty of Science, University of Split, Croatia in 2001. The topic of his graduate thesis was "Project Approach to Computer Based Intelligent Tutoring Systems". He gained a MS degree in computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia in 2005. The topic of his MS thesis was "Knowledge Evaluation Model in Intelligent e-Learning Systems". He gained a PhD degree in computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia in 2010. The topic of his PhD thesis was "Model of Intelligent Tutoring System Based on Processing of Controlled Language over Ontology". Since 2002 he has been employed as an Assistant Professor at the Department of Computer Science, Faculty of Science, University of Split, Croatia. His research interests are intelligent tutoring systems, natural language processing, knowledge representation and software engineering.