

A METHOD FOR LEARNING SCENARIO DETERMINATION AND MODIFICATION IN INTELLIGENT TUTORING SYSTEMS

ADRIANNA KOZIERKIEWICZ-HETMAŃSKA, NGOC THANH NGUYEN

Institute of Informatics
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: adrianna.kozierkiewicz@pwr.wroc.pl

Computers have been employed in education for years. They help to provide educational aids using multimedia forms such as films, pictures, interactive tasks in the learning process, automated testing, etc. In this paper, a concept of an intelligent e-learning system will be proposed. The main purpose of this system is to teach effectively by providing an optimal learning path in each step of the educational process. The determination of a suitable learning path depends on the student's preferences, learning styles, personal features, interests and knowledge state. Therefore, the system has to collect information about the student, which is done during the registration process. A user is classified into a group of students who are similar to him/her. Using information about final successful scenarios of students who belong to the same class as the new student, the system determines an opening learning scenario. The opening learning scenario is the first learning scenario proposed to a student after registering in an intelligent e-learning system. After each lesson, the system tries to evaluate the student's knowledge. If the student has a problem with achieving an assumed score in a test, this means that the opening learning scenario is not adequate for this user. In our concept, for this case an intelligent e-learning system offers a modification of the opening learning scenario using data gathered during the functioning of the system and based on a Bayesian network. In this paper, an algorithm of scenario determination (named ADOLS) and a procedure for modifying the learning scenario AMLS with auxiliary definitions are presented. Preliminary results of an experiment conducted in a prototype of the described system are also described.

Keywords: e-learning system, intelligent tutoring system, learning scenario, personalization.

1. Introduction

Intelligent tutoring systems (in this work also called intelligent e-learning systems or just e-learning systems) provide directed, customized and individualized instructions or feedback to students. They are able to offer educational material suitable for a user's learning style, knowledge, interests, abilities, etc., adapt the learning environment to the student's preferences, and offer adaptive tests appropriate to the learner's current knowledge level. It has been shown that students are interested and more motivated if they learn using intelligent tutoring systems where the learning process is individualized.

Such formulated tasks are required to apply algorithms which allow proposing an individual learning process for each user. The procedure for providing a personalized learning process consists of several steps. First a student's profile which contains two types of data, user data and usage data (Kobsa *et al.*, 2001), is built. The system has to collect information concerning demographic data

(such as login, name, age, sex, educational level, IQ), the learning style (related to perception, receiving, processing and understanding of information by the student), abilities (verbal comprehension, word fluency, computational ability, spatial visualization, associative memory, perceptual speed, reasoning), personal character traits (such as concentration, motivation, ambition, self-esteem, level of anxiety, locus of control, open mind, impetuosity, perfectionism) and interests (humanities, formal science, natural science, economics and law, technical science, business and administration, sports and tourism, artistic science, management and organization, education).

During the registration process the student fills in a questionnaire to provide demographic data. The learning style, abilities, character traits and interests are assessed using psychological tests and questionnaires. Some of the collected information about students, i.e., name, login, telephone, allow only identifying and distinguishing among students in a e-learning system, whereas others reflect stu-

dents' needs and preferences. Very significant for the recommendation process is data referring to the learning style, i.e., visual students remember better what they see so they need more pictures, flows, charts, films in their learning scenario. If a learner is identified as active, the system should propose various experiments, tests, exercises which permit him/her to discover new facts and relationships because this type of student prefers doing something with acquired information. Also, abilities and character traits influence the learning process and should be considered during the planning of a learning scenario, i.e., a student with low level of motivation needs more complicated tasks and exercises. A learner characterized by word fluency processes educational material given as a text better (Koziarkiewicz, 2008a).

In the next step a user is classified to a group of similar students according to some attributes chosen by experts because it is assumed that similar students will learn in the same or a very similar way (Kukla *et al.*, 2004). Each group gathers students who exhibit similar learning styles, abilities, character traits or interests. Before a student starts to learn, the system has to choose an opening learning scenario adequate for him/her. By the opening learning scenario we mean the order and presentation form of educational material proposed to a student after registration in the system. An opening learning scenario is determined on the basis of successfully finished scenarios of students who belong to the same class as the new learner. Such a solution allows offering the opening learning scenario which is suitable for the student's preferences and needs. We can assume that, if similar students described by similar features finish a similar learning scenario with positive notes, the new learner is expected to successfully complete the offered learning scenario. The method of opening learning scenario determination is based on consensus theory and will be described in detail in Section 4.

After determining the opening learning scenario, lessons are presented to the user on its basis. After each lesson, his/her current knowledge level is evaluated by using a test. Questions selected for the test fit the student's knowledge level. The advantage of adaptive tests is more accurate evaluation in a shorter time period.

A student learns according to the opening learning scenario until the first test failure. If the test result is not sufficient, then it is a signal for the system that the offered learning scenario is not suitable for this user. The system tries to improve the process and modifies the learning scenario by suggesting re-learning, changing the lessons' order or presentation methods. The procedure for modifying of a learning scenario during a learning process uses a Bayesian network and is described in Section 5. The learning process is finished when all lessons from the learning scenario are taught. The idea of a personalized learning process is presented in Fig. 1 (Koziarkiewicz-Hetmańska, 2009; Kukla *et al.*, 2004).

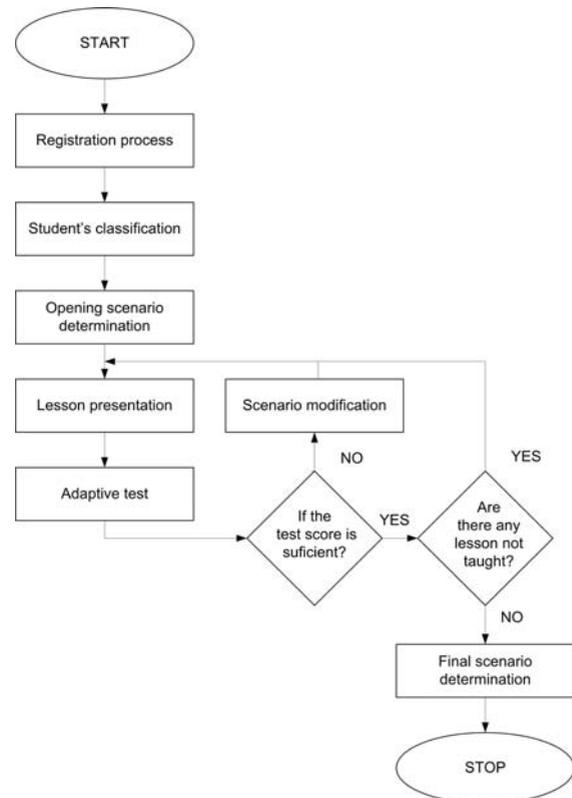


Fig. 1. Idea of a personalized learning process.

This paper is devoted to methods of learning scenario determination and modification in an intelligent tutoring system. The proposed procedures allow offering an optimal learning path in each step of an educational process. Consensus theory is applied in the algorithm to generate the opening learning scenario (Nguyen, 2002; 2008). The method by Koziarkiewicz (2008b) is refined and matched to a graph-based knowledge structure. The procedure of modification of a learning scenario uses the Bayesian network. It is original and innovative and presented for the first time. So far, there has been no solution to the problem of modification of a learning scenario during a learning process and based on a current student's characteristic. Both algorithms apply a graph-based knowledge structure defined in this work.

In the next section, an overview of different forms of knowledge representation and methods used in e-learning systems is presented. In Section 3, our model is included, in which a knowledge structure and a definition of a learning scenario are presented. Section 4 contains a description of the method of determination of an opening learning scenario with auxiliary definitions. The procedure of modifying a learning scenario during a learning process is presented in Section 5. Next, the concept of an experiment using a prototype of an e-learning system is proposed. Section 7 shows preliminary results of an experiment. Finally, conclusions and future work are described.

2. Related work

A typical intelligent e-learning system consists of three modules: a student module, a domain module and a tutor module. The first one contains descriptions of student knowledge, behavior, demographic data, learning style, interests, etc. The domain module is responsible for knowledge representations. The last module, serving as a teacher in traditional learning, controls the learning process. Methods implemented in the tutor module allow determining the opening learning scenario, modifying a learning scenario during a learning process, offering a suitable method of evaluation, etc.

2.1. Domain module: Knowledge representation. Intelligent tutoring systems have been more popular in recent years. The first intelligent tutoring web-based systems appeared in 1995–1996. In intelligent e-learning systems, knowledge is represented in different ways. In one of the first intelligent tutoring systems, ELM-ART, domain knowledge is represented as a multi-layered overlay model (Weber and Brusilovsky, 2001). The first layer announces that a learner visited a page corresponding to a unit. The second layer describes completed exercises. In the next layer, information about units which could be known by learning other units is stored. In the last layer, it is possible to mark some units as known.

A Bayesian network is another methodology of representing knowledge in an intelligent tutoring system. In the work of Gamboa and Fred (2001), a knowledge base consists of two types of nodes: learned and shown. Each node represents one topic—atomic content part. The topic is composed by a hypertext page and a set of questions. Some nodes are dependent on others.

Knowledge can be represented as a semantic network with frames and production rules (Stankov *et al.*, 2000). Basic components are nodes representing domain knowledge objects, links illustrating relations between nodes, properties and frames.

Hewahi (2007) presents a hierarchical rule structure. Each rule is structured as a rule name, condition(s), an action or a decision if the condition is satisfied, the parent rule, the rule to be tried if the current rule is matched and fired and the rule to be followed in case of a failure of the current rule.

In the WELSA system (Popescu *et al.*, 2008), domain knowledge is represented as a learning object. The most complex learning object is the course, while the smallest is the elementary resource. Each course consists of several chapters, each chapter can contain several sections and subsections. The lowest level subsections contain the actual educational resources. The learning object is described by a set of metadata regarding the instructional role (fundamental or auxiliary), the media type (text, image, animation, sound, video), level of abstractness and formality,

the type of competence, etc. All descriptions are independent of any learning style.

An ontology and a concept map are very popular forms for knowledge representation. In the work of Bozueghoub *et al.* (2004), the ontology contains concepts and relationships among them. These authors use two types of relationships: narrower/broader to support hierarchical links between concepts and a rhetorical relationship. The RDF schema to describe the proposed model is used.

In the research by Klaus-Dieter *et al.* (2005), knowledge is modeled as an outlined graph structured in vertices called learning units and edges representing links between the units. Each link from unit u_1 to u_2 corresponds to a possible transition from unit u_1 to target unit u_2 . The graph is represented by a communication matrix (the form of an extended adjacency matrix).

In the EDUCE system (Kelly and Tangney, 2002), knowledge is structured into two hierarchical models consisting of concepts and learning units. Concepts represent the section into which the knowledge base is divided, learning units present the content of a concept in different ways, for example, by using text, image, video, audio, animation. The INSPIRE system (Grigoriadou *et al.*, 2002) is structured in three levels of abstraction: learning goals, concepts and educational materials. Learning goals consist of a subset of concepts. Authors distinguished different types of concepts: outcome, prerequisite and related. The educational material is associated with each outcome concept and consists of knowledge modules developed according to three different levels of performance: *Remember*, *Use* and *Find*. Outcome concepts of a learning goal are organized in a layered structure.

Kukla *et al.* (2004) assume that the knowledge structure is considered on two levels: conceptual (knowledge pieces and relations between them) and presentation (sequences of hypermedia pages called presentations). The learning scenario is a sequence of presentations and tests after each presentation. In the book by Nguyen (2008), the knowledge structure is modified and represented on two levels: concepts and a presentation of concepts. A learning scenario does not contain tests. In the work of Kozierkiewicz (2008b), the described knowledge structure was extended to the third level: versions of presentations.

2.2. Tutor module: Methods of generating educational material. The method of determination of a learning scenario is based on the knowledge structure and its presentation. In ELM-ART (Brusilovsky *et al.*, 1996), the problem of providing a student with suitable learning material is solved by using visual adaptive annotation of links. A student is offered a special link in the form of colored bullets: green, red, white and orange which inform the student about the educational status of the next pages. Such a solution allows the student to navigate indepen-

dently through pages of the course. ELM-ART offers also the "next topic" button. The next units are determined based on the student's current characteristic. The next step is chosen by computing all prerequisites that are necessary to fulfil the goal. Subsequent concepts belonging to the set of prerequisites which have not been learned yet are offered to the student. This method, called curriculum sequencing, is also applied in KBS Hyperbook, InterBook, PAT InterBook, CALAT, VC Prolog Tutor, ELM-ART-II, AST, ADI, ART-Web, ACE and ILESA (Brusilovsky, 1999).

The Bayesian network is a very popular methodology applied in intelligent e-learning systems to determine suitable learning materials. Gamboa and Fred (2001) proposed a model of an intelligent tutoring system using Bayesian networks. For offering material the pedagogical module is responsible learning. The role of this module is to choose between the following actions: show a new topic, deepen a current topic, review a previous topic and present the next page in the index. The Bayesian network for the pedagogical action was built. Important variables are two parameters: time spent on the corresponding topic and the question answered. The highest probability node is selected.

A Bayesian belief network was used to generate the learning path also by Nguyen *et al.* (2008). For each user of the system, the best learning path based on the learner's profile and knowledge map is determined. The generation process consists of several steps: learner evaluation, creation of a knowledge map, candidate path selection based on the shortest path searching and learning path generation using the Bayesian belief network.

In the EDUCE system (Kelly and Tangney, 2004), there are four different types of learning materials offered (verbal/linguistic, visual/spatial, logical/mathematical and musical/rhythmic). EDUCE tries to predict which type of resource a student would prefer. Using information about the time of learning, how many times a user looked at the type of material and attempted a question and after which resource he/she gets the question right, the system customizes the learning path. In EDUCE, a naive Bayes algorithm is applied.

Rius *et al.* (2008) discuss advantages of applied ontology in automated generation of a personalized learning scenario. Essalmi *et al.* (2007) proposed ontology describing learning scenarios and the process of personalization of a learning scenario based on this ontology. In the proposed ontology, the operations of the Guilford model are associated to the levels of Bloom's taxonomy. A student achieves goals defined according to Bloom's taxonomy.

In WELSA (Popescu *et al.*, 2008), the user model is created dynamically by monitoring and analyzing the student's behavior. When the student's learning preferences are detected, system uses proper adaptation techniques to propose a learning material suitable for the user's needs. The adaptation techniques were decomposed into elemen-

tary adaptation actions such as sorting, inserting or removing learning objects, i.e., if the student's preference is denoted as a visual, a specific multimedia object, based on video and images, it should be inserted into the course.

In INSPIRE (Grigoriadou *et al.*, 2002), the lesson generation module is responsible for planning the learning path. In the generation process, the student's current knowledge level is taken into account. The generated lesson includes a set of presentations of the outcome concepts, links to brief presentations of the prerequisite concepts focusing on their relation with the outcome, and links to definitions of related concepts.

In the paper by Kukla *et al.* (2004), the idea of an algorithm for determination of an opening learning scenario consists of two steps: the choice of presentations and the knowledge pieces order and choice of corresponding tests to presentations. In the paper by Nguyen (2008), the algorithm of determination of an opening learning scenario is based on the choice of concept orders and presentations and presentation orders. This method was improved by adding the third step: the choice of suitable versions of lessons (Koziarkiewicz, 2008b).

In the works of Koziarkiewicz-Hetmańska (2009) as well as Koziarkiewicz-Hetmańska and Nguyen (2010), the concept of modification of a learning scenario during a learning process is proposed. The procedure of modification is conducted in three steps. If a student failed the test for the first time, he/she is offered the same lesson but in a different version. After the second test failure, the system offers changing the lesson order based on data of students who belong to the same class. After another failure, the user is proposed a modification of the lessons order based on all collected data. The student has three chances of passing the test for each lesson. To the best of our knowledge, the problem of modification of learning scenarios during learning processes has not been solved in an effective way yet.

3. Knowledge structure

In the field of education, several postulates specify how educational material should be taught. It is natural that for designing and creating e-learning systems, results of education researchers should be used. It has been reported that students prefer learning materials divided into smaller pieces. Therefore, in this paper the division of the educational material into lessons is assumed. Each lesson exists in one of the following forms: textual, graphical, interactive. This solution allows offering interesting, multimedia courses and creating a learning environment suitable for a student's preferences. Between lessons linear orders occur. This means that all lessons from our repository need to be learned, but some of them should be learned before others. The relation between lessons defines the order in which lessons should be presented to a user. After each

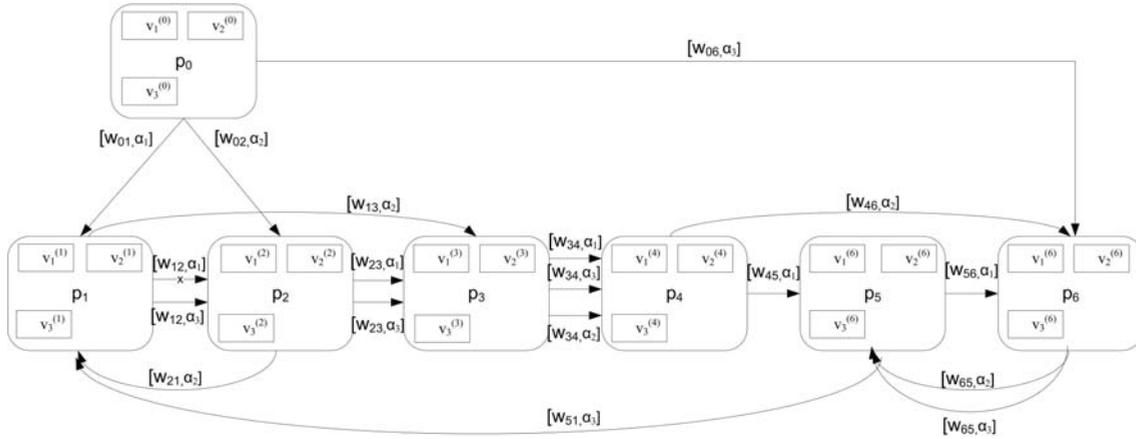


Fig. 2. Graph-based knowledge structure.

lesson, the student has to pass a test. The exception is the first lesson which contains information about the goals of the coursework and its requirements.

The knowledge structure consists of lessons, relations between them and their versions. Some data stored in an e-learning system are used to define the knowledge structure for improving and making it more flexible. During the functioning of an intelligent tutoring system, it collects and stores information such as the average score for each lesson, the average time of learning of each lesson and the difficulty degree of each lesson, which is measured by the number of failed tests. These data are stored separately in two-dimensional arrays for different student classes and different lesson orders:

$$AS_g = [as_{iz}]_{\substack{i=0, \dots, q \\ z=1, \dots, q}}, \quad (1)$$

where as_{iz} stands for 100% minus the average score for lesson p_z which was learnt after lesson p_i , $i \in \{0, \dots, q\}, z \in \{1, \dots, q\}, g \in \{1, \dots, G\}$, G is the number of classes, that is, the number of different groups of similar students, each class created based on some users' attributes chosen by an expert.

$$AS = [asc_{iz}]_{\substack{i=0, \dots, q \\ z=1, \dots, q}}, \quad (2)$$

where

$$asc_{iz} = \frac{1}{G} \sum_{y=1}^G as_{iz}$$

for $i \in \{0, \dots, q\}, z \in \{1, \dots, q\}, g \in \{1, \dots, G\}$.

$$AD_g = [ad_{iz}]_{\substack{i=0, \dots, q \\ z=1, \dots, q}}, \quad (3)$$

where ad_{iz} is the difficulty degree of lesson p_z , represented by the number of test failures referred to lesson p_z (learnt after lesson p_i) divided by the number of all tests taken by students who learnt lesson p_z after lesson p_i ,

$i \in \{0, \dots, q\}, z \in \{1, \dots, q\}, g \in \{1, \dots, G\}$, G is the number of classes.

$$AD = [adc_{iz}]_{\substack{i=0, \dots, q \\ z=1, \dots, q}}, \quad (4)$$

where

$$adc_{iz} = \frac{1}{G} \sum_{y=1}^G ad_{iz}$$

for $i \in \{0, \dots, q\}, z \in \{1, \dots, q\}, g \in \{1, \dots, G\}$,

$$AT_g = [at_{iz}]_{\substack{i=0, \dots, q \\ z=1, \dots, q}}, \quad (5)$$

where at_{iz} is the average time of learning lesson p_z (which was learnt after lesson p_i), $i \in \{0, \dots, q\}, z \in \{1, \dots, q\}, g \in \{1, \dots, G\}$, G stands for the number of classes.

$$AT = [atc_{iz}]_{\substack{i=0, \dots, q \\ z=1, \dots, q}}, \quad (6)$$

where

$$atc_{iz} = \frac{1}{G} \sum_{y=1}^G at_{iz}$$

for $i \in \{0, \dots, q\}, z \in \{1, \dots, q\}, g \in \{1, \dots, G\}$.

Let us assume that P is a finite set of lessons. Each lesson $p_i \in P, i \in \{0, \dots, q\}$ is a set of different versions $v_k^{(i)} \in p_i, k \in \{1, \dots, m\}$, m is the number of different versions, $i \in \{0, \dots, q\}$. R_C are called linear orders on P . A binary relation $\alpha \in R_C$ is called linear if the relation is reflexive, transitive, antisymmetric and total. By

$$W = [w_{iz}]_{\substack{i=0, \dots, q \\ z=1, \dots, q}}$$

we denote the weight matrix, where w_{iz} could equal $as_{iz}, asc_{iz}, ad_{iz}, adc_{iz}, at_{iz}, atc_{iz}$ for $i \in \{0, \dots, q\}, z \in \{1, \dots, q\}$

The graph-based structure of knowledge is defined in the following way (Kozierkiewicz-Hetmańska, 2009; Kozierkiewicz-Hetmańska and Nguyen, 2010).

Definition 1. The *graph-based knowledge structure* is a labeled and directed graph:

$$Gr = (P, E, \mu), \quad (7)$$

where P is the set of nodes, E is the set of edges, $\mu : E \rightarrow L$ is the function assigning labels to edges, $L = \bigcup_{f=1}^{\text{card}(R_C)} L(\alpha_f)$ is the set of labels where $L(\alpha_f) = (W, \alpha_f)$, $f \in \{1, \dots, \text{card}(R_C)\}$, $\alpha_f \in R_C$.

Figure 2 presents an example of a defined knowledge structure. For the defined knowledge structure, the learning scenario is as follows.

Definition 2. By the *Hamiltonian path* based on order $\alpha_f \in R_C$ in graph Gr we mean a sequence of nodes:

$$hp = \langle p_0, \dots, p_q \rangle, \quad (8)$$

where (a) for each $i \in \{0, \dots, q\}$ $p_i \neq p_{i+1}$ and (b) for each $e \in E$ $\mu(e) \in L(\alpha_f)$, where $f \in \{1, \dots, \text{card}(R_C)\}$.

Definition 3. By the *learning scenario* s we mean a Hamiltonian path hp based on an order $\alpha_f \in R_C$ in which exactly one element from each node p_i , $i \in \{0, \dots, q\}$ occurs:

$$hp = \langle v_k^{(0)}, \dots, v_n^{(q)} \rangle, \quad (9)$$

where $v_k^{(0)} \in p_0, \dots, v_n^{(q)} \in p_q$ for $k, n \in \{1, \dots, m\}$, m being the number of different versions of lessons.

We let S_C be the set of all possible scenarios following the defined knowledge structure.

Example 1. For knowledge structure presented in Fig. 2 some different scenarios are defined as follows:

$$\begin{aligned} s_1 &= \langle v_1^{(0)}, v_1^{(1)}, v_2^{(2)}, v_1^{(3)}, v_3^{(4)}, v_3^{(5)}, v_3^{(6)} \rangle, \\ s_2 &= \langle v_1^{(0)}, v_2^{(1)}, v_2^{(2)}, v_2^{(3)}, v_2^{(4)}, v_1^{(5)}, v_1^{(6)} \rangle, \\ s_3 &= \langle v_1^{(0)}, v_1^{(6)}, v_2^{(5)}, v_1^{(1)}, v_1^{(2)}, v_1^{(3)}, v_3^{(4)} \rangle. \end{aligned}$$



4. Determination of an opening learning scenario

In traditional learning, after the inauguration lesson all students start the learning process according to a syllabus. They learn the same educational material at the same time, i.e., during one class only one particular subject is being covered and taught in the same way by a teacher. In an e-learning system, the learning process is assumed to be personalized and adapted to student preferences. After the registration process, a new student is classified to a class of students similar to him/her. Then the opening learning scenario for the new student is chosen from the set

S_C of all possible scenarios, which generates the minimal sum of distances to the final successful scenarios of students who belong to the same class as the new student. The proposed method is based on consensus theory (Kozierekiewicz, 2008b). The problem of determination of an opening learning scenario is defined in the following way: *For given learning scenario $s^{(1)}, s^{(2)}, \dots, s^{(n)}$, one should determine a scenario $s^* \in S_C$ such that the condition*

$$\sum_{i=1}^n d(s^*, s^{(i)}) = \min_s \sum_{i=1}^n d(s, s^{(i)}) \quad (10)$$

is satisfied.

For this problem the first task is to define the distance function between two different scenarios $s^{(1)}$ and $s^{(2)}$.

Definition 4. By $d : S_C \times S_C \rightarrow [0, 1]$ we mean a distance function between scenarios $s^{(1)}$ and $s^{(2)}$. The distance function is computed as

$$d(s^{(1)}, s^{(2)}) = \lambda_1 \sigma(\alpha^{(1)}, \alpha^{(2)}) + \lambda_2 \delta(\alpha^{(1)}, \alpha^{(2)}), \quad (11)$$

where S_C is the set of learning scenarios, $\lambda_1 + \lambda_2 = 1$, the values of λ_1 and λ_2 are chosen by an expert from the set $(0, 1)$, $s^{(1)}$ and $s^{(2)}$ are based on orders $\alpha^{(1)}$ and $\alpha^{(2)}$, respectively.

The value of the distance function $d(s^{(1)}, s^{(2)})$ is estimated in two steps:

Step 1:

$$\sigma(\alpha^{(1)}, \alpha^{(2)}) = \frac{1}{q+1} \sum_{i=0}^q S(\alpha^{(1)}, \alpha^{(2)}),$$

where $S(\alpha^{(1)}, \alpha^{(2)}) = |k^{(1)} - k^{(2)}| / (q+1)$ if lesson p occurs in $\alpha^{(1)}$ in position $k^{(1)}$ and in $\alpha^{(2)}$ in position $k^{(2)}$.

Step 2:

$$\delta(\alpha^{(1)}, \alpha^{(2)}) = \frac{1}{q+1} \sum_{i=0}^q \Theta(v_k^{(1)(i)}, v_y^{(2)(i)}),$$

where

$$\Theta(v_k^{(1)(i)}, v_y^{(2)(i)}) = \begin{cases} 1 & \text{if } v_k^{(1)(i)} \neq v_y^{(2)(i)}, \\ 0 & \text{otherwise.} \end{cases}$$

The proposed distance function is a metric because it is the sum of two elements being metrics. A proof that the function σ is a metric is given by Nguyen (2008). The function δ is a Hamming metric.

Example 2. Let $\lambda_1 = \lambda_2 = 1/2$ and

$$s_1 = \langle v_1^{(0)}, v_1^{(1)}, v_2^{(2)}, v_1^{(3)}, v_3^{(4)}, v_3^{(5)}, v_3^{(6)} \rangle,$$

$$s_2 = \langle v_1^{(0)}, v_1^{(6)}, v_2^{(5)}, v_1^{(1)}, v_1^{(2)}, v_1^{(3)}, v_3^{(4)} \rangle.$$

The distance function $d(s^{(1)}, s^{(2)})$ between scenarios $s^{(1)}$ and $s^{(2)}$ equals

$$\sigma(\alpha^{(1)}, \alpha^{(2)}) = \frac{1}{q} \sum_{i=1}^q S(\alpha^{(1)}, \alpha^{(2)})$$

$$= \frac{\frac{2}{7} + \frac{2}{7} + \frac{2}{7} + \frac{2}{7} + \frac{3}{7} + \frac{5}{7}}{7} = \frac{16}{49},$$

$$\delta(\alpha^{(1)}, \alpha^{(2)}) = \frac{1}{q} \sum_{i=1}^q \Theta(v_k^{(1)(i)}, v_y^{(2)(i)})$$

$$= \frac{0 + 0 + 1 + 0 + 0 + 1 + 1}{7} = \frac{3}{7}.$$

Therefore

$$d(s^{(1)}, s^{(2)})$$

$$= \frac{1}{2} \sigma(\alpha^{(1)}, \alpha^{(2)}) + \frac{1}{2} \delta(\alpha^{(1)}, \alpha^{(2)}) = \frac{1}{2} \cdot \frac{16}{49} + \frac{1}{2} \cdot \frac{3}{7}$$

$$= \frac{16 + 21}{98} = \frac{37}{98} \approx 0,378.$$

The opening learning scenario is conducted in two steps: the first one depends on a proper order of lessons, and in the second one the system chooses suitable versions of lessons. Such a solution allows adapting a learning scenario to the student's preferences. The procedure of determining the opening learning scenario which we named ADOLS (an algorithm for determination of an opening learning scenario) is presented as follows:

Algorithm ADOLS

Given: $s^{(1)}, s^{(2)}, \dots, s^{(n)}$

Result: s^*

BEGIN

1. $i=0$;
2. For lesson p_i determine a multiset: $I(p_i) = \{jm : \text{there exists a scenario that } p_i \text{ occurs on its } jm\text{-th position}\}$;
3. Compute $J(p_i) = \sum_{jm \in I(p_i)} jm$;
4. $i++$;
5. Repeat Steps 2-4 until $i \leq q$;
6. Set lessons in relation α^* according to the increasing values of $J(p_i)$;
7. $i=0$;
8. For each $k \in \{1, \dots, m\}$ compute $f(v_k^{(i)})$, that is the number of appearances of

$v_k^{(i)}$ in scenarios $s^{(1)}, s^{(2)}, \dots, s^{(n)}$

9. For lesson p_i choose a version of lesson $v_t^{(i)}$ such that:

$$f(v_t^{(i)}) = \max_{k \in \{1, \dots, m\}} f(v_k^{(i)})$$

10. $i++$;
11. Repeat Steps 8-10 until $i \leq q$;
12. If the determined opening learning scenario s^* is not based on any order $\alpha_f \in RC$, $f \in \{1, \dots, \text{card}(RC)\}$ then {
 $i=1$;
 while ($i \leq n$) {
 Compute $\sum_{j=1}^n d(s^{(i)}, s^{(j)})$;
 $i++$;
 }
 Choose a scenario s^* such that:
 $\sum_{i=1}^n d(s^*, s^{(j)}) = \min_i \sum_{i=1}^n d(s^{(i)}, s^{(j)})$
 } else
 END

The proof of the correctness of Algorithm ADOLS should be given. It will be presented in future works.

Example 3. For the learning scenario given below,

$$s_1 = \langle v_1^{(0)}, v_1^{(1)}, v_2^{(2)}, v_1^{(3)}, v_3^{(4)}, v_3^{(5)}, v_3^{(6)} \rangle,$$

$$s_2 = \langle v_1^{(0)}, v_2^{(1)}, v_2^{(2)}, v_2^{(3)}, v_2^{(4)}, v_1^{(5)}, v_1^{(6)} \rangle,$$

$$s_3 = \langle v_1^{(0)}, v_1^{(6)}, v_2^{(5)}, v_1^{(1)}, v_1^{(2)}, v_1^{(3)}, v_3^{(4)} \rangle,$$

s^* is determined. The first lesson order is determined:

$$I(p_0) = \{1, 1, 1\}, \quad I(p_1) = \{2, 2, 4\},$$

$$I(p_2) = \{3, 3, 4\}, \quad I(p_3) = \{4, 4, 6\},$$

$$I(p_4) = \{5, 5, 7\}, \quad I(p_5) = \{6, 6, 3\},$$

$$I(p_6) = \{7, 7, 2\}$$

and

$$J(p_0) = 3, \quad J(p_1) = 8, \quad J(p_2) = 10,$$

$$J(p_3) = 14, \quad J(p_4) = 17, \quad J(p_5) = 15,$$

$$J(p_6) = 16.$$

Thus $\alpha^* = \langle p_0, p_1, p_2, p_3, p_4, p_5, p_6 \rangle$.

Next, a version of lessons is selected.

Table 1. Number of appearances $v_k^{(i)}$ in scenarios $s^{(1)}, s^{(2)}, s^{(3)}$.

Lesson	$f(v_1^{(i)})$	$f(v_2^{(i)})$	$f(v_3^{(i)})$	Chosen
p_0	3	0	0	$v_1^{(0)}$
p_1	2	1	0	$v_1^{(1)}$
p_2	1	2	0	$v_2^{(2)}$
p_3	2	1	0	$v_1^{(3)}$
p_4	0	1	2	$v_3^{(4)}$
p_5	1	1	1	$v_1^{(5)}$
p_6	2	0	1	$v_1^{(6)}$

Thus, the determined opening learning scenario s^* equals

$$s^* = \langle v_1^{(0)}, v_1^{(1)}, v_2^{(2)}, v_1^{(3)}, v_3^{(4)}, v_1^{(5)}, v_1^{(6)} \rangle.$$

If we assumed a learning scenario like in Fig. 2, then s^* does not belong to S_C and we should proceed to Step 12. Then we obtain

$$s^* = \langle v_1^{(0)}, v_2^{(1)}, v_2^{(2)}, v_2^{(3)}, v_2^{(4)}, v_1^{(5)}, v_1^{(6)} \rangle.$$

◆

5. Modification of the learning scenario

In traditional learning, if a student has a problem with passing a test, the teacher tries to analyze the reason for mistakes. Sometimes the learner is not concentrated or well prepared. It is also possible that lessons are too hard or not well explained. The student is proposed repetitions of learning material and the retaking of the test. Sometimes learning the same educational material is enough to master this part of knowledge. Sometimes the student needs to read additional books, notes or receive credit for different lessons.

In this paper an original method based on a Bayesian network is proposed. The Bayesian network is an acyclic and directed graph. Let X_i for $i \in \{1, \dots, n\}$ a random variable stored in a node of a graph. A Bayesian network represents joint probability distribution $P(x_i | x_{\text{par}(i)})$ over the set of random variables $X = \cup_i^n X_i$, where $x_{\text{par}(i)}$ is a set of parent variables of variable X_i for each $i \in \{1, \dots, n\}$. The joint probability is computed in the following way:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | x_{\text{par}(i)}). \quad (12)$$

The conditional probability equals

$$P(x_i | x_j) = \frac{P(x_i, x_j)}{P(x_j)}. \quad (13)$$

The above equation could be transformed using Bayes' rule:

$$P(x_i | x_j) = \frac{P(x_j | x_i) \cdot P(x_i)}{P(x_j)}. \quad (14)$$

An e-learning system stores opening and final scenarios, the student's demographic data, learning styles, the time of learning each lesson, learning results, etc. The above-mentioned data are used to modify the learning scenario and build the Bayesian network.

Let us assume that a student is offered an opening learning scenario $s = \langle v_{k_0}^{(0)}, \dots, v_{k_z}^{(z)}, v_{k_i}^{(i)}, \dots, v_{k_q}^{(q)} \rangle$ based on order $\alpha_f \in R_C$, where $k_1, k_z, k_i, k_q \in \{1, \dots, m\}$, $f \in \{1, \dots, \text{card}(R_C)\}$. If the student fails the test after lesson p_i which was learned after lesson p_z ,

$i, z \in \{0, \dots, q\}$, the Bayesian network is built. The procedure of creation of a Bayesian model consists of two steps: construction of a graph representing the qualitative influences of the modeled situation, and assignment of probability tables to each node in the graph.

In our model, the following variables are considered:

- *time*: the time of learning lesson p_i which was learnt after lesson p_z , the set of values: *average* if $time \in [atc_{iz} - \sigma, atc_{iz} + \sigma]$, where σ is the standard deviation, *less* if $time < atc_{iz} - \sigma$, *more* if $time > atc_{iz} + \sigma$;
- *number*: the number of tests taken for lesson p_i , the set of values equals $\{1, 2, 3, > 3\}$;
- *score*: the last of test scores in percent for lesson p_i which was learnt after lesson p_z , the set of values: 1 if $score \in (0\%, 10\%]$, 2 if $score \in [10\%, 20\%]$, 3 if $score \in (20\%, 30\%]$, 4 if $score \in (30\%, 40\%]$, 5 if $score \in (40\%, 50\%]$;
- *version*: the number of versions of the lesson which occurs in the opening learning scenario s most often before learning lesson p_i ; the set of values equals $\{1, 2, \dots, m\}$;
- *lesson*: p_i , the number of failed lesson and their version;

After each lesson, the system prepares an adaptive test which should be taken by the user. If the student fails the test for lesson p_i which was learned after lesson p_z , $i \in \{0, \dots, q\}$, the system dynamically creates a Bayesian network based on collected data. A graphical representation of the created Bayesian network is presented in Fig. 3.

The second step in creating the Bayesian network is the assignment of probability tables to each node in the graph. The probabilities are estimated based on observing the student's interaction with the system. It is necessary to assess the local distributions:

$p(\text{time} = wt)$ for each $wt \in \{\text{less}, \text{average}, \text{more}\}$,

$p(\text{number} = wn)$ for each $wn \in \{1, 2, 3, > 3\}$,

$p(\text{score} = ws)$ for each $ws \in \{1, 2, 3, 4, 5\}$,

$p(\text{lesson} = v_k^{(i)})$ for each $k \in \{1, \dots, m\}$
and $i \in \{0, \dots, q\}$,

$p(\text{version} = k)$ for each $k \in \{1, \dots, m\}$,

and the conditional distributions:

$p(v_l^{(z+1)} \in \alpha_f(p_z) | \text{time} = wt, \text{number} = wn, \text{score} = ws, \text{lesson} = v_k^{(i)}, \text{version} = k)$,

$p(v_k^{(z+2)} \in \alpha_f(\alpha_f(p_z)) | v_l^{(z+1)} \in \alpha_f(p_z))$,

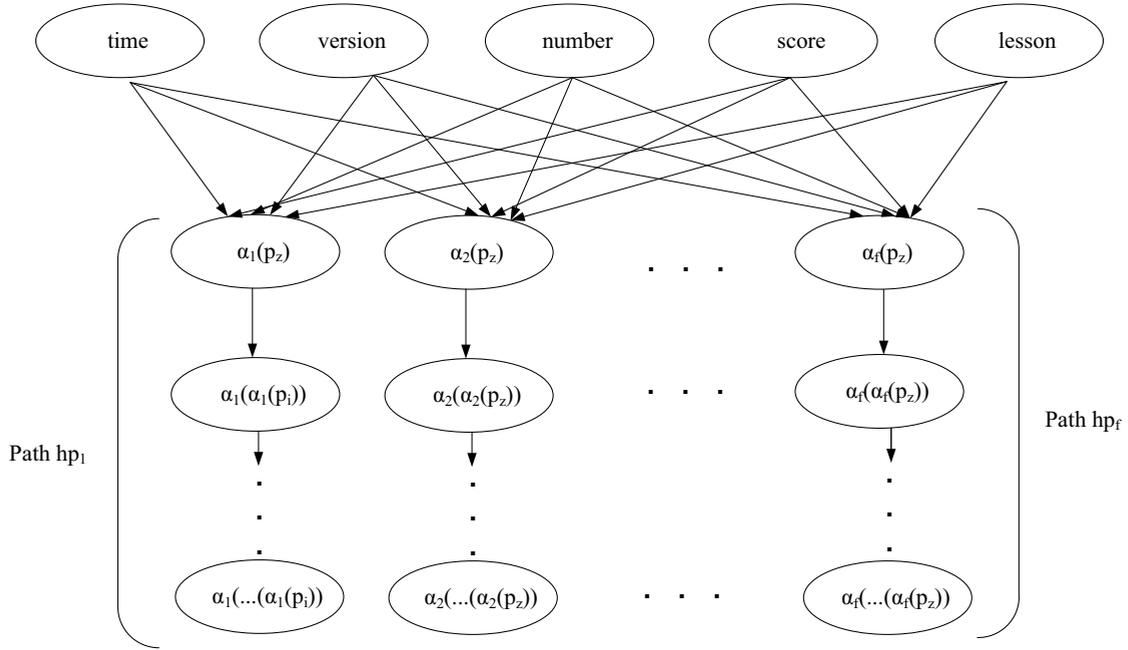


Fig. 3. Model of the Bayesian network.

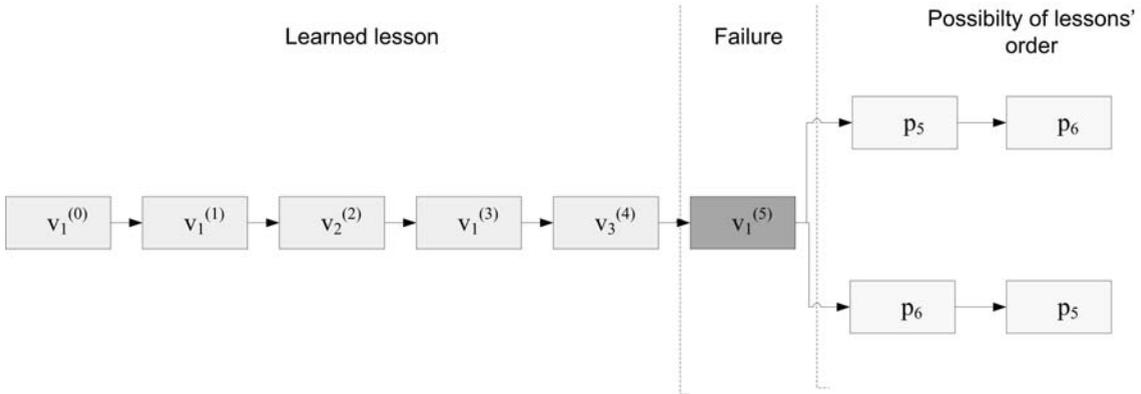


Fig. 4. Situation described in Example 4.

⋮

$$p(v_n^{(z+d+1)} \in \alpha_f(\dots(\alpha_f(\alpha_f(p_z)))) | v_k^{(z+d)} \in \alpha_f(\dots(\alpha_f(p_z))))$$

for each $wt \in \{less, average, more\}$, $wn \in \{1, 2, 3, > 3\}$, $ws \in \{1, 2, 3, 4, 5\}$, $k, n, l \in \{1, \dots, m\}$, $i, z \in \{0, \dots, q\}$, $f \in \{1, \dots, card(R_C)\}$, $d \in \{2, \dots, (q-z)\}$.

For example, if $p(version = 2) = 0.45$, this means that 45% of students complete the course according to learning scenarios where the lesson version 2 dominated. The probability $p(v_2^{(5)} | time = less, number = 1, score = 2, lesson = v_3^{(5)}, version = 3) = 0.3$ means that 30% of student who failed the test for lesson $v_3^{(5)}$ for

the first time scored between 10% and 20% and with additional conditions that the time of learning was less than the average and the lesson version 2 dominated finish the course if the system offers them in the next step the version of lesson $v_2^{(5)}$.

By using the Bayesian network it is possible to determine a new learning scenario. A student passes the test for lesson p_z but fails the test for lesson p_i . The choice of the next lesson is dependent on the time of learning lesson p_i , the most popular version of lessons, the number of test failures, the test score for lesson p_i and the version of lesson $v_k^{(i)}$. In the Bayesian network there exists a Hamiltonian path hp attached based on order α_f from a graph-based knowledge structure. Hence, $(p_z, p_{z+1}) \in \alpha_f$ and $p_{z+1} = \alpha_f(p_z)$ for fixed $f \in \{1, \dots, card(R_C)\}$. The algorithm of modification of a learning scenario is based on

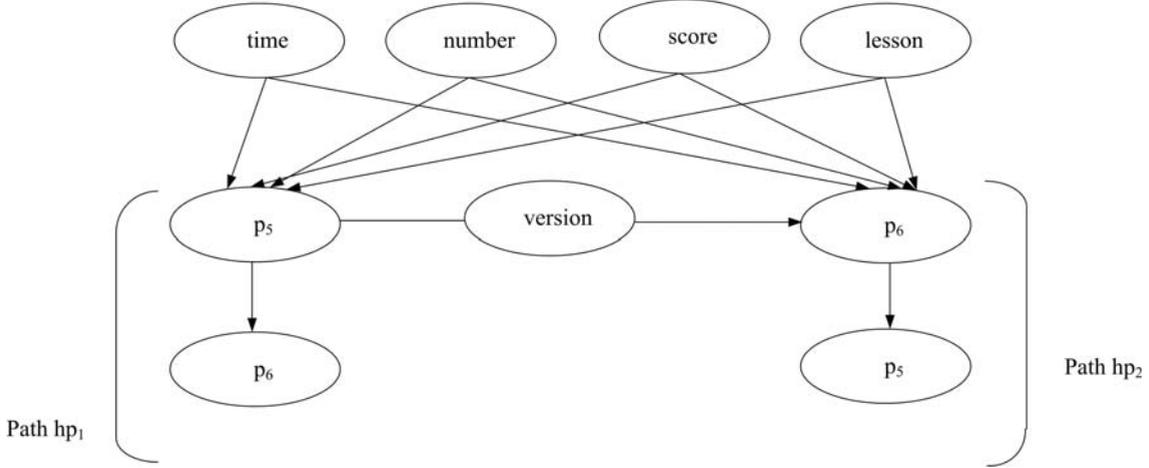


Fig. 5. Model of the Bayesian network for the situation described in Example 4.

the highest posterior probability, and therefore we choose a lesson and a version of the lesson for which the following condition is satisfied:

$$\arg \max_{k,f} p(v_k^{z+1} \in \alpha_f(p_z), time, number, score, lesson, version). \quad (15)$$

The choice of subsequent lessons depends only on the previous lesson. Let us assume that in the previous step the order α_{mi} was chosen, where $mi \in \{1, \dots, \text{card}(R_C)\}$. Therefore, in subsequent steps versions of lessons should be chosen which satisfy the following condition:

$$\arg \max_k p(v_n^{z+d+1}, v_k^{z+d}), \quad (16)$$

where $d \in \{2, \dots, (q - z)\}$, $k, n \in \{1, \dots, m\}$, $z \in \{0, \dots, q\}$.

The procedure of modification of the learning scenario which is named AMLS (the algorithm for modification of learning scenario) is given as follows.

Algorithm AMLS

Given: $p_i, p_z, wt, wn, ws, v_k^{(i)}, n$ knowledge structure $Gr, s^{(1)}, s^{(2)}, \dots, s^{(n)}$, data collected during functioning of the system

Result: s' - modified scenario

BEGIN

1. Create graph Gr' by removing nodes: p_0, \dots, p_z and edges between them;
2. Estimate all local and conditional probabilities;
3. $f = 2; mi = 1; mk = 1; mp = p(v_1^{(z+1)} \in \alpha_1(p_z), time = wt, number = wn, score = ws, lesson = v_k^{(i)}, version = n)$

4. $l = 1;$
 5. If $p(v_l^{(z+1)} \in \alpha_f(p_z), time = wt, number = wn, score = ws, lesson = v_k^{(i)}, version = k) > mp$ then $\{p(v_l^{(z+1)} \in \alpha_f(p_z), time = wt, number = wn, score = ws, lesson = v_k^{(i)}, version = k) = mp; mi = f; mk = l; \}$
 6. $l++;$
 7. Repeat Steps 5 and 6 until $l \leq m;$
 8. $f++;$
 9. Repeat Steps 4-8 until $f \leq \text{card}(R_C);$
 10. Add $v_{mk}^{(z+1)} \in \alpha_{mi}(p_z)$ to scenario s' ;
 11. $d = 1;$
 12. $n = mk;$
 13. $mk = 1; mp = p(v_1^{z+d+1}, v_n^{z+d});$
 14. $l = 2;$
 15. If $p(v_l^{z+d+1}, v_n^{z+d}) > mp$ then $\{mp = p(v_l^{z+d+1}, v_n^{z+d}); mk = l; \}$
 16. $l++;$
 17. Repeat Steps 15 and 16 until $l \leq m;$
 18. Add $v_{mk}^{(z+d+1)}$ to scenario s' ;
 19. $d++;$
 20. Repeat Steps 12-19 until $d \leq q;$
- END

Example 4. Let us assume a knowledge structure presented in Fig. 2 and $m = 3$. Suppose that the system determined for a new user the opening learning scenario $\langle v_1^{(0)}, v_1^{(1)}, v_2^{(2)}, v_1^{(3)}, v_3^{(4)}, v_1^{(5)}, v_1^{(6)} \rangle$ and the student passes tests for lessons p_1, p_2, p_3, p_4 but fails the test for lesson p_5 . The student needs to learn lessons p_5, p_6 to complete the course. The described situation is presented in Fig. 4, the model of the Bayesian network is presented in Fig. 5.

We assumed that for our learner $time$ for lesson p_5 is $less$, he tried to pass a test for lesson p_5 only once, so

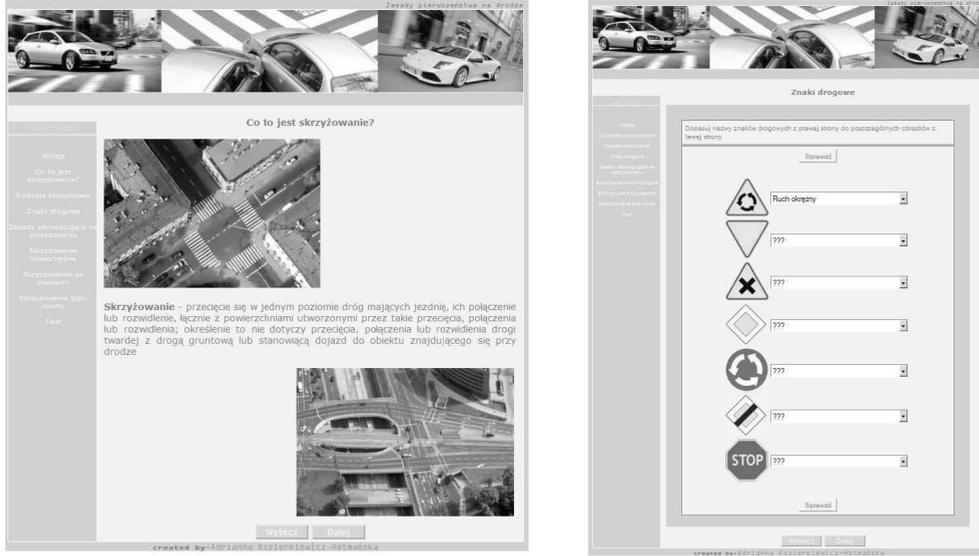


Fig. 6. Graphical and the interactive version of lessons: definition of an intersection and road signs.

$number = 1$; he scored 15%, so $score = 2$, $lesson = v_1^{(5)}$ and $version = 1$. Let us assume that, based on the final scenarios and all collected data, the probabilities are determined as follows:

$$p(time = less) = 0.3,$$

$$p(number = 1) = 0.3,$$

$$p(score = 2) = 0.1,$$

$$p(lesson = v_1^{(5)}) = 0.1,$$

$$p(version = 1) = 0.15,$$

$$p(v_1^{(5)} | time = less, number = 1, score = 2, lesson = v_1^{(5)}, version = 1) = 0.8,$$

$$p(v_2^{(5)} | time = less, number = 1, score = 2, lesson = v_1^{(5)}, version = 1) = 0.45,$$

$$p(v_3^{(5)} | time = less, number = 1, score = 2, lesson = v_1^{(5)}, version = 1) = 0.13,$$

$$p(v_1^{(6)} | time = less, number = 1, score = 2, lesson = v_1^{(5)}, version = 1) = 1,$$

$$p(v_2^{(6)} | time = less, number = 1, score = 2, lesson = v_1^{(5)}, version = 1) = 0.9,$$

$$p(v_3^{(6)} | time = less, number = 1, score = 2, lesson = v_1^{(5)}, version = 1) = 0.1,$$

$$p(v_1^{(5)} | v_1^{(6)}) = 0.9,$$

$$p(v_2^{(5)} | v_1^{(6)}) = 0.8,$$

$$p(v_3^{(5)} | v_1^{(6)}) = 0.3.$$

We present probabilities essential in the situation considered. First, we have to determine

$$p(v_k^{(i)}, time = less, number = 1,$$

$$score = 2, lesson = v_1^{(5)}, version = 1),$$

where $i \in \{5, 6\}$, $k \in \{1, 2, 3\}$.

Using Eqn. (12), we compute

$$p(v_1^{(5)}, time = less, number = 1, score = 2,$$

$$lesson = v_1^{(5)}, version = 1)$$

$$= p(v_1^{(5)} | time = less, number = 1, score = 2,$$

$$lesson = v_1^{(5)}, version = 1)$$

$$\cdot p(time = less) \cdot p(number = 1)$$

$$\cdot p(score = 2) \cdot p(lesson = v_1^{(5)})$$

$$\cdot p(version = 1)$$

$$= 0.8 \cdot 0.3 \cdot 0.1 \cdot 0.1 \cdot 0.15 = 0.00036$$

$$p(v_2^{(5)}, time = less, number = 1, score = 2,$$

$$lesson = v_1^{(5)}, version = 1) = 0.0002025$$

$$p(v_3^{(5)}, time = less, number = 1, score = 2,$$

$$lesson = v_1^{(5)}, version = 1) = 0.0000585,$$

$$p(v_1^{(6)}, time = less, number = 1, score = 2,$$

$$lesson = v_1^{(5)}, version = 1) = 0.00045,$$

$$p(v_2^{(6)}, time = less, number = 1, score = 2,$$

$$lesson = v_1^{(5)}, version = 1) = 0.000405,$$

$$p(v_3^{(6)}, time = less, number = 1, score = 2,$$

$$lesson = v_1^{(5)}, version = 1) = 0.000045.$$

We choose $v_1^{(6)}$ because for this version of the lesson the following condition is satisfied:

$$\arg \max_{v_k^{(i)}} p(v_k^{(i)}, time, number, score, lesson, version).$$

(17)

In the next step, we compute

$$p(v_1^{(5)}, v_1^{(6)}) = 0.3 \cdot 0.00045 = 0.000405,$$

$$p(v_2^{(5)}, v_1^{(6)}) = 0.00036,$$

$$p(v_3^{(5)}, v_1^{(6)}) = 0.000135.$$

For the version of the lesson $v_1^{(5)}$, the condition

$$\arg \max_k p(v_k^{(5)}, v_1^{(6)}) \quad (18)$$

is satisfied.

The opening learning scenario was modified because the student failed the test for lesson p_5 . The modified learning scenario equals

$$s = \langle v_3^{(0)}, v_3^{(1)}, v_3^{(2)}, v_1^{(3)}, v_2^{(4)}, v_1^{(6)}, v_1^{(5)} \rangle.$$

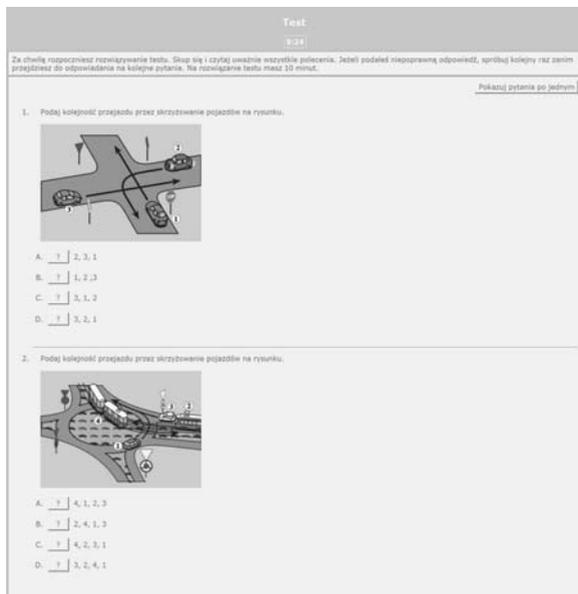


Fig. 7. Example questions from the test.

6. Concept of experiments

A designer of e-learning systems has to combine knowledge from different scientific areas: computer science as well as education. Such an approach requires evaluation of proposed solutions through an experiment. In our research we want to show that a student achieves better test results in a shorter time if the learning scenario is adapted to the student's preferences in each step of the learning process. The experiment will be carried out using an implemented prototype of an e-learning system. The application field will be part of a course of driving regulations. The educational material focuses on intersections, road signs related to intersections and right-of-way laws. Students who take part in the experiment are offered different learning scenarios. Learning scenarios differ from each other in the lessons order and presentations methods, for example, some scenarios contain more graphical elements such as pictures, graphs, films, other contain interactive elements whereas yet another some more text. Figure 6 presents two

lessons: one in a graphical and the other in an interactive form.

The experiment starts after the registration process. A student fills in the Index of Learning Styles Questionnaire (Soloman and Felder, 2010) and provides information such as login, password, sex, age, educational level. Users could be assigned to two groups: a control group and an experimental group, based on sex, educational level, age and driving license. The distribution of women, men, young people, old people, middle-aged people, educated people, uneducated people, with and without the driving license should be even in both groups. If a student is assigned to the control group, he/she is offered a universal learning scenario and the same as the other students in this group. The universal learning scenario is similar to lessons prepared by a teacher in school. There is no possibility to personalize the learning scenario according to student preferences and needs, so the universal learning scenario consists of a few pictures, texts, films, tasks, etc. and is a combination of dedicated learning scenarios. Users from the experimental group are proposed educational material suitable for their learning style. After learning, a student has to pass a test. The test consists of 10 questions chosen randomly from a question bank consisting of 30 items. Test time is limited to 10 minutes. Figure 7 presents example questions from the test.

Students have 3 chances of passing the test. The method of modification of a learning scenario described in Section 5 was simplified. If a user fails the test for the first time, he/she is offered the same learning scenario. If the test score is still not sufficient after the second attempt the system chooses the learning scenario which results in the student's achieving best results. The third failure finishes the experiment without successful graduation. The described e-learning system prototype is available at <http://brylant.iit.pwr.wroc.pl/~koziarkiewicz/system>.

7. Preliminary experimental result

Preliminary research was conducted using a prototype of an e-learning system described in Section 6. In our experiment, 38 users were assigned to an experimental group and 32 to a control group. We want to verify the hypothesis that those from the experimental group achieve better test scores than those from the control group.

The mean scores of experimental and control groups satisfy the following assumption: $H_0: \mu_{\text{exp}} = \mu_{\text{cont}}$, or the student's alternative assumption: $H_1: \mu_{\text{exp}} > \mu_{\text{cont}}$. There is no information on the standard deviation, and the sizes of groups are different. That is why, for testing the null hypothesis, the following statistic is assumed:

$$u = \frac{\mu_{\text{exp}} - \mu_{\text{cont}}}{\sqrt{\frac{S_{\text{exp}}^2}{n_{\text{exp}}} + \frac{S_{\text{cont}}^2}{n_{\text{cont}}}}}, \quad (19)$$

where μ_{exp} is the average score value for the experimental group, μ_{cont} is the average value for the control group, S_{exp} is the standard deviation for the experimental group, S_{cont} is the standard deviation for the control group, n_{exp} is the size of the experimental group, n_{cont} the size of the control group. Using data collected during the experiment, the following values are calculated:

$$\begin{aligned} n_{\text{exp}} &= 55, \\ n_{\text{cont}} &= 67, \\ \mu_{\text{exp}} &= 59.818, \\ \mu_{\text{cont}} &= 52.239, \\ S_{\text{exp}} &= 23.08, \\ S_{\text{cont}} &= 22.248. \end{aligned}$$

The tested statistical value equals

$$u = \frac{59.818 - 52.239}{\sqrt{\frac{23.08^2}{55} + \frac{22.248^2}{67}}} = 1.834.$$

The significance level of 0.05 is assumed. The confidence interval is equal to $[1.64, +\infty)$.

The tested statistical value belongs to the confidence interval, and hence the null hypothesis is rejected and the alternative hypothesis is assumed. From our experiment we can draw a conclusion that the mean score of the experimental group is greater than that of the control group. Therefore students who were offered a personalized learning scenario achieve better learning results than those who were proposed a universal learning scenario.

8. Conclusions and future work

In this paper, the learning path generating process based on users' profiles is described. A student is suggested an opening learning scenario which is the best way of participating in the course for this learner. He or she learns according to an opening learning scenario until the first test failure. The system modifies the learning scenario based on all collected data and the learner's current knowledge level. A student continues the learning process according to the new learning scenario. Such an approach allows adapting the learning path in each step of the learning process and providing educational material that will be best assimilated.

The preliminary experimental results showed that such a solution is efficient. Students who were offered a personalized learning scenario achieve better learning results than those who were offered a universal learning scenario. We can draw a conclusion that taking into account

a student's preferences and learning style increases the effectiveness of learning process and shows the correctness of the developed algorithms.

We plan to collect more experimental results and introduce them in future works. It is also planned to carry out statistical tests and provide a proof of the correctness of algorithms described in this paper.

The proposed method might be applied in an e-learning system used for conducting training, supporting traditional learning in schools (the presented material can be shown in an interesting, multimedia form) or in a remote coaching system for weaker students who require more material to be shown.

Acknowledgment

This research was financially supported by the European Union within the European Social Fund under the fellowship no. MK/SN/87/II/2010/U, and by the Polish Ministry of Science and Higher Education under the grants 0419/B/T02/2009/37 and N N519 407437.

References

- Bouzeghoub, A. Defude, B., Ammour, S., Duitama, J.F. and Lécocq, C.(2004). A RDF description model for manipulating learning objects, *Proceedings of the International Conference on Advanced Learning Technologies, Joensuu, Finland*, pp. 81–85.
- Brusilovsky, P. (1999). Adaptive and intelligent technologies for web-based education, *Kunstliche Intelligenz* **13**(4): 19–25.
- Brusilovsky, P., Schwarz, E. and Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web, in C. Frasson, G. Gauthier, A. Lesgold (Eds.), *Intelligent Tutoring Systems*, Springer-Verlag, Berlin, pp. 261–269.
- Essalmi, F., Jemni, L. and Ayed, L. (2007). A process for the generation of personalized learning scenarios based on ontologies, *Proceedings of International Conference on Information and Communication Technology and Accessibility, Hammamet, Tunisia*, pp. 173–175.
- Gamboa, H. and Fred, A. (2001). Designing intelligent tutoring systems: A Bayesian approach, *ICEIS 2001—International Conference on Enterprise Information Systems, Setubal, Portugal*, pp. 452–458.
- Grigoriadou, M., Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H. and Magoulas, G.D. (2002). INSPIRE: An intelligent system for personalized instruction in a remote environment, in S. Reich, M. Tzagarakis and P.M.E. De Bra (Eds.), *Hypermedia: Openness, Structural Awareness, and Adaptivity*, Lecture Notes in Computer Science, Vol. 2266, Springer-Verlag, Berlin/Heidelberg, pp. 215–225.
- Hewahi, N. (2007). Intelligent tutoring system: Hierarchical rule as a knowledge representation and adaptive pedagogical model, *Information Technology Journal* **6**(5): 739–744.
- Kelly, D. and Tangney, B. (2002). Incorporating learning characteristics into an intelligent tutor, in S.A. Cerri, G. Gouarderes and F. Paraguaçu (Eds.), *Intelligent Tutoring Systems*,

- Lecture Notes in Computer Science, Vol. 2363, Springer-Verlag, Berlin/Heidelberg, pp. 729–738.
- Kelly, D. and Tangney, B. (2004). Predicting learning characteristics in a multiple intelligence based tutoring system, in J.C. Lester, R.M. Vicari and F. Paraguaçu (Eds.), *Intelligent Tutoring Systems*, Springer-Verlag, Berlin, pp. 9–30.
- Klaus-Dieter, S., Thalheim B., Binemann-Zdanowicz A., Kascsek R., Kuss T. and Tschiedel B. (2005). A conceptual view of web-based e-learning systems, *Education and Information Technologies* **10**(1): 83–110.
- Kobsa, A., Koenemann, J. and Pohl, W. (2001). Personalized hypermedia presentation techniques for improving online customer relationships, *Knowledge Engineering Review* **16**(2): 111–155.
- Koziarkiewicz, A. (2008a). Content and structure of learner profile in an intelligent e-learning system, in N.T. Nguyen, G. Kolaczek and B. Gabrys (Eds.), *Knowledge Processing and Reasoning for Information Society*, Exit, Warsaw, pp. 101–116.
- Koziarkiewicz, A. (2008b). Determination of opening learning scenarios in intelligent tutoring systems, in A. Zgrzywa, K. Choroś and A. Siemiński (Eds.), *New Trend in Multimedia and Network Information Systems*, IOS Press, Amsterdam, pp. 204–213.
- Koziarkiewicz-Hetmańska, A. (2009). A conception for modification of learning scenario in an intelligent e-learning system, in N.T. Nguyen, R. Kowalczyk and S.-M. Chen (Eds.), *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, Lecture Notes in Artificial Intelligence, Vol. 5796, Springer-Verlag, Berlin/Heidelberg, pp. 87–96.
- Koziarkiewicz-Hetmańska, A. and Nguyen, N.T. (2010). A method for scenario modification in intelligent e-learning systems using graph-based structure of knowledge, in N.T. Nguyen, R. Katarzyniak, Ch. Shyi-Ming (Eds.), *Advances in Intelligent Information and Database Systems*, Studies in Computational Intelligence, Vol. 283 Springer-Verlag, Berlin, pp. 169–179.
- Kukla, E. E. Nguyen, N.T., Danilowicz, C., Sobacki, J. and Lenar M. (2004). A model conception for optimal scenario determination in an intelligent learning system, *ITSE—International Journal of Interactive Technology and Smart Education* **1**(3): 171–184.
- Nguyen, N.T. (2002). Consensus systems for conflict solving in distributed systems, *Information Sciences* **147**(1): 91–122.
- Nguyen, N.T. (2008). *Advanced Methods for Inconsistent Knowledge Management*, Springer-Verlag, New York, NY.
- Nguyen, V.A., Nguyen, V.H. and Ho, S.D. (2008). Constructing a Bayesian belief network to generate learning path in adaptive hypermedia system, *Journal of Computer Science and Cybernetics* **24**(11): 12–19.
- Popescu, E., Badica, C. and Trigano, P. (2008). Learning objects' architecture and indexing in WELSA adaptive educational system, *Scalable Computing: Practice and Experience* **9**(1): 11–20.
- Rius, A., Sicilia, M. and Garsia-Barriocanal, E. (2008). An ontology to automate learning scenarios? An approach to its knowledge domain, *Interdisciplinary Journal of E-Learning and Learning Objects* **4**(1): 151–165.
- Soloman, B.A. and Felder, R. (2010). *Index of Learning Styles Questionnaire*, <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>.
- Stankov, S., Glavinic, V. and Rosic, M. (2000). On knowledge representation in an intelligent tutoring system, *Proceedings of INES'2000—International Conference on Intelligent Engineering Systems, Portoroz, Slovenia*, pp. 381–384.
- Weber, G. and Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for web-based instruction, *International Journal of Artificial Intelligence in Education* **6**(5): 351–384.



Adrianna Koziarkiewicz-Hetmańska received the M.Sc. degree in computer science from the Wrocław University of Technology in 2007. She is currently a Ph.D. student of computer science at the same university. Her research interests include intelligent tutoring systems, rough classification, consensus theory and other artificial intelligence applications. She is an author and co-author of several publications at national and international conferences and workshops. For her research project she has been awarded a few scholarships and grants.



Ngoc Thanh Nguyen currently works at the Wrocław University of Technology, Poland, where he is the head of the Knowledge Management Systems Division at the Institute of Informatics. He received his M.Sc. degree in computer science in 1986, Ph.D. in 1989, D.Sc. in 2002, and the professorial title in 2009. He has edited 20 special issues of international journals and 10 conference proceedings. He is the author or editor of 14 books and more than 200 other publications. His latest monograph, entitled *Advanced Methods for Inconsistent Knowledge Management* was published by Springer in 2008. He is the editor-in-chief of two international journals: *LNCS Transactions on Computational Collective Intelligence* (Springer) and the *International Journal of Intelligent Information and Database Systems* (Inderscience), and of two book series for the IGI Global Publisher (USA). He is an associate editor of four international journals and a member of editorial boards of several other international journals.

Received: 12 April 2010

Revised: 29 November 2010