UDC 681.5.09

J. P. MARTINEZ BASTIDA, A. G. CHUKHRAY, E. V. GAVRILENKO

National Aerospace University named after N. E. Zhukovzky «KhAI», Ukraine

MODELS AND METHODS FOR IMPLEMENTING PEDAGOGICAL INTERVENTIONS IN MODEL-TRACING COGNITIVE TUTORS

This paper presents some models and methods for generating pedagogical interventions in model-tracing cognitive tutors. They use Bayesian networks for assessment and making decisions, this feature allows managing uncertainty reasoning based on a formal foundation. This technique combines the rigorous probabilistic formalisms with a graphical representation and efficient inference mechanisms. It is explained how Bayesian networks are employed as an inference engine to assess the degree of learning of the relevant knowledge components in the learning domain and determine the proper pedagogical interventions for performing a productive learning process.

Keywords: *information technologies, Bayesian network, pedagogical intervention, model-tracing, cognitive tutors, Bayesian assessment.*

Introduction

Model-tracing cognitive tutors (MTCT) have successfully been applied on different knowledge domains, and have proved positive results on students while acquiring skills and knowledge under specific learning domains [1-4]. Moreover, MTCT can track students' actions to provide pedagogical interventions such as hints and feedbacks in a task-structured curriculum [2]. This feature, also called cognitive model needs determining the student's thinking and the required skills and knowledge for the learning domain. Cognitive models (CM) are an integral part of developing Intelligent Tutoring Systems (ITS) [4]. Thus, a CM requires a proper understanding of the knowledge involved in student's actions in a given learning domain, problem-solving strategies or principles and it should also be able to interpret student's recurrent behavioral patterns and tendencies that reflect a way of thinking in order to provide constructive pedagogical interventions. A MTCT is "interested" on the way the student processes and assimilates the relevant knowledge components, this can be tracked by analyzing the behavior when the student attempts to commit actions to satisfy the requirement(s) of a task, and it can be recurrent in terms of the way that knowledge is required, in other words; how tasks are graphically presented. An approach discussed on this paper is based on the hypothesis that some students are less able to seek for help when they needed or get close to a person to get it, e.g. the teacher or other means of information, communication or learning support, due to the lack of meta-cognitive skills for "help-seeking" [3]. This approach gives students support for developing skills like help-seeking and self-regulatory by means of adaptive pedagogical interventions. These interventions may be in the context of an interactive learning environment that leads them to learn the knowledge and skills of certain domain.

So according to the hypothesis that a help-seeking student becomes a better learner [3], the MTCT uses a task specific pedagogical intervenor. Mainly cognitive tutors support the base of learning by doing, help-seeking instructions and self-analyzing. These features in learning platforms and cognitive tutors have been tested and they prove to raise student's scores [3-5]; the models and methods explained in this work are developed to perform pedagogical interventions according to student's actions and performance. For implementing and testing them, a MTCT named TITUS was developed [6]. It employs Bayesian Networks (BNs) with diagnostic models for assessing students [5, 6, 8]. Therefore, the aim of this work is to present some models and methods for implementing into the development of Model-tracing cognitive tutors that make use of Bayesian networks with diagnostic models for performing inferences in order to produce adaptive pedagogical interventions.

1. Generic description of the two loops pedagogical interventions framework

It is expressed on [5] that tutors behaves similarly despite of their different structures, thus a common two loops structural blocks are proposed, the outer and inner loops. Despite the essence and the main common features of the two loops structure had been described in detail in the literature [5], there still exist the absence of the related models and methods to understand their internal functionality and guide designers to build algorithms or solutions for developing them. A basic explanation of the common functionalities in the two loop structure is presented as following.

In short words, the outer loop has the responsibility to define the task that the student should do next. The main challenges this loop presents are selecting intelligently a task as well as developing a rich enough set of tasks to select from. On the other hand, inner loop is focused on the student's actions while he attempts to complete a determined task; whereas the outer loop deals with the tasks, the inner loop deals with the steps related to complete a task and offers the students some "services" as they use the tutor. These services will be deeper explained ahead on the inner loop related section.

Furthermore, in basis of Bloom's taxonomy [7] that is employed on different spheres of pedagogy, including the computer-oriented training, an ITS can attain the first three lowest levels: knowledge, understanding and application. Assessment and acquirement of new knowledge or skill based on the trained knowledge components by the students can be carried out when they attempt to complete different practical tasks, computing solutions of tasks, fulfillment of individual laboratory practices, etc. Higher levels assume creativity moments and ambiguity; thus, these aspects cannot be realized without the expert assessment of the teacher. Learning automation under such circumstances lies beyond this work.

2. Models for implementing the outer loop

Assessment model for determining the degree of learning of knowledge components in the domain. A key factor to aid the student to navigate through the learning domain is to be able to model the prior degree of learning he has and to keep track of each relevant knowledge component, and Bayesian networks (BN) can help to manage this uncertainty [8-10].

The basic structure for the BN that models the degree of learning in a student is depicted on Fig. 1.

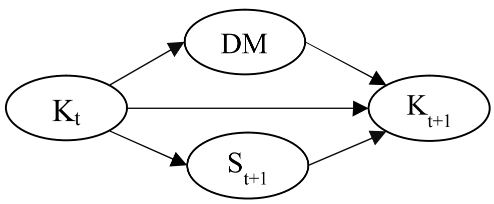

Fig. 1. BN assessment basic structure

This BN consists of four nodes: K_t, S_{t+1}, DM and K_{t+1}, where K_t is the probability of learning of certain knowledge component or skill at t time; S_{t+1} is the step or student's action at moment t+1 after he attempts to complete certain task; DM is a diagnostic model [9] that is directly linked to the student's actions and influences the probability of the degree of learning of the relevant knowledge component at t+1 moment; and K_{t+1} is the probability of learning of the knowledge component at t+1 moment. $\neg K_t$, $\neg S_{t+1}$, $\neg DM$ and $\neg K_{t+1}$ are the respective complementary probabilities.

Probability $P(K_{t+1})$ of learning certain knowledge component at t+1 moment, after a student's correct action is obtained with (1), in this situation, evidences in a student's action are denoted by $P(S_{t+1}) = 1$ (correct action) and $P(DM) = 0$ (deactivated).

$$\begin{aligned}
P\left(K_{t+1}\right)=&[P\left(K_t|S_{t+1},\neg DM\right)\cdot P\left(K_{t+1}|K_t,S_{t+1},DM\right)\cdot\\
&\cdot P\left(S_{t+1}\right)P\left(DM\right)]+[P\left(\neg K_t|S_{t+1},\neg DM\right)\cdot\\
&\cdot P\left(K_{t+1}|\neg K_t,S_{t+1},DM\right)\cdot P\left(S_{t+1}\right)\cdot P\left(DM\right)]+\\
&+[P(K_t|S_{t+1},\neg DM)\cdot P(K_{t+1}|K_t,S_{t+1},\neg DM)\cdot P(S_{t+1})\cdot\\
&\cdot P(\neg DM)]+[P(\neg K_t|S_{t+1},\neg DM)\cdot P(K_{t+1}|\neg K_t,S_{t+1},\neg DM)\cdot\\
&\cdot P(S_{t+1})\cdot P(\neg DM)]+[P(K_t|S_{t+1},\neg DM)\cdot\\
&\cdot P(K_{t+1}|K_t,\neg S_{t+1},DM)\cdot P(\neg S_{t+1})\cdot P(DM)]+\\
&+[P(\neg K_t|S_{t+1},\neg DM)\cdot P(K_{t+1}|\neg K_t,\neg S_{t+1},DM)\cdot P(\neg S_{t+1})\cdot\\
&\cdot P(DM)]+[P(K_t|S_{t+1},\neg DM)\cdot P(K_{t+1}|K_t,\neg S_{t+1},\neg DM)\cdot\\
&\cdot P(\neg S_{t+1})\cdot P(\neg DM)]+[P(\neg K_t|S_{t+1},\neg DM)\cdot\\
&\cdot P(K_{t+1}|\neg K_t,\neg S_{t+1},\neg DM)\cdot P(\neg S_{t+1})\cdot P(\neg DM)].
\end{aligned} \tag{1}$$

Conditional probabilities $P(K_t|S_{t+1},\neg DM)$ and $P(\neg K_t|S_{t+1},\neg DM)$ in (1) are obtained with (2) and (3) respectively,

$$\begin{aligned}
P\left(K_t|S_{t+1},\neg DM\right)&=\alpha \sum_{K_{t+1}} P\left(K_t,S_{t+1},\neg DM,K_{t+1}\right)=\\
&=\alpha \sum_{K_{t+1}} P(K_t)\cdot P(S_{t+1}|K_t)\cdot P(\neg DM|K_t)\cdot\\
&\cdot P\left(K_{t+1}|K_t,S_{t+1},\neg DM\right).
\end{aligned} \tag{2}$$

$$\begin{aligned}
P\left(\neg K_t|S_{t+1},\neg DM\right)&=\alpha \sum_{K_{t+1}} P\left(\neg K_t,S_{t+1},\neg DM,K_{t+1}\right)=\\
&=\alpha \sum_{K_{t+1}} P(\neg K_t)\cdot P(S_{t+1}|\neg K_t)\cdot P(\neg DM|\neg K_t)\cdot\\
&\cdot P\left(K_{t+1}|\neg K_t,S_{t+1},\neg DM\right).
\end{aligned} \tag{3}$$

where α is a normalization coefficient,

$$\alpha\cdot\left(P\left(K_t|S_{t+1},\neg DM\right)+P\left(\neg K_t|S_{t+1},\neg DM\right)\right)=1.$$

This model assumes that each task depends on individual knowledge components. That is, the set of relevant knowledge components in a task are individual cognitive processes, thus when a student attempts to complete a task, they can be applied independently one from another, so their posterior probability must be assessed separately.

Therefore, a step analyzer assesses each relevant knowledge component in the actual task and passes it to the outer and inner loops in order to determine the proper pedagogical interventions. Thus, outer and inner loops directly depend on this assessment.

Model for selecting tasks. For implementing this model, a set of several tasks for the learning domain must be developed and separated into complexity levels and sequential modules. Three or five levels of complexity are commonly instantiated under the macroadaptation approach as standard for educational proposes [4-6] (e.g. very easy, easy, average, difficult, and very difficult). Modules should be created so that in each of them there were two tasks as minimum from each level of complexity in order to have alternatives for a choice. Moreover, all the set of tasks in a module must cover the complete set of relevant knowledge components related to it, and they should be trained more than once in each level of complexity. Set of tasks in every module should be developed as an interwoven network over the relevant knowledge components that it contains. Thus, it is preferable that every knowledge component should be trained at least by two different tasks; this relation between a knowledge component and tasks increases the probability of learning it by increasing the times of possible situations that students might employ it, this is well known because it is the classic approach that is commonly implemented in the classrooms. Task model (MT) above explained, is represented by (4) and its boundaries in (5). Fig. 2 depicts and example of the MT.

$$MT : \{T_{ijk}\} \to \{KW_{kl}\}, \qquad (4)$$

where T is a task, KW defines a knowledge component, i is the task identifier, $j \in [1, 5]$ represents the levels of complexity, k is the module of the task T, and l is the identification number for the knowledge component.

$$\begin{gathered} \forall_k, \forall_j, \{T_{ijk}\} = 0, \ \|T_{ijk}\| \ \geq 2, \\ \forall_k, \forall_j, \forall_l \ \{T_{ijk}\} = MT^{-1}(KW_{kl}) \neq 0, \|T_{ijk}\| \geq 2, \\ \forall_k \cdot \bigcup MT(T_{ijk}) = \{KW_{kl}\}, \end{gathered} \qquad (5)$$

where MT^{-1} - $\{KW_{kl}\} \to \{T_{ijk}\}$.

The parameters related to the learning domain are fixed, and the number or tasks included in the MTCT may be added but this process is performed out-of-working time, thus the MT is static. On the other hand, the student model (MS) is constantly updated while the student is working with the MTCT, for this reason, MS is a dynamical representation of the student.

This "representation" might include name, surname, user, password, learning progress, performance, right attempts, and another key information that outer loop in join with the inner loop may use for assessing the student's degree of learning of each knowledge component in the learning domain. The above concepts can be represented in (6) and (7), where S represents the student, q is his identification number, $P \subset \mathbf{R}$ are real numbers in the interval [0, 1], that represents the probability of learning certain knowledge component, N is the attempts of completing a determined tasks.

$$MS1 : \{S_q\} \times \{T_{ijk}\} \to N. \qquad (6)$$

$$MS2 : \{S_q\} \times \{KW_{kl}\} \to P. \qquad (7)$$

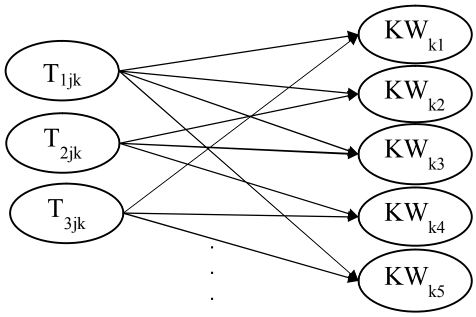

Fig. 2. Task model structure (example)

The prior information is initialized if a student S_q uses the tutor for the first time, thus for each S_q: $\forall_i$, $\forall_j$, $\forall_k$ $MS1(S_q, T_{ijk})=\{0\}$; $\forall_l$, $\forall_k$ $MS2(S_q, KW_{kl}) =\{0.5\}$. After this, first module is selected and complexity level is set to the middle one: k = min(k), j = [max(j)/2]. As it was commented above, the outer loop selects the next tasks (NT) in a certain module represented by (8), for a determined student, containing knowledge components with lower degree of learning.

$$NT=MT^{-1}\left(KW_{kl} \middle| MS2(S_q, KW_{kl}) \to \min\right). \qquad (8)$$

Because many knowledge components might have a lower degree of learning and tasks may contain several of them, (8) can return more than one choice ($\|NT\|>1$), thus the outer loop searches the next task based on attempts (NT'), represented by (9).

$$NT'=NT\left(KW_{kl} \middle| MS1(S_q, T_{ijk}) \to \min\right). \qquad (9)$$

In case that $\|NT'\|>1$, the outer loop will implement (10), and randomly will realize a model imitation for selecting a task (NT*). This case is certainly possible at the first time a student uses the MTCT.

$$NT^* = RAND(NT). \qquad (10)$$

Processes described by (8)-(10) repeats in accordance with the situations above explained, meanwhile the student has not probably learned the knowledge components in the current module; only then, the outer loop passes to the closest upper module: k + 1 and again it repeats the process of choosing the next task, until k < max(k). In case the complete set of knowledge components in the domain and the max(k) have reached, the training program comes to the end.

3. Models and method for implementing the inner loop

Models for defining complexity level and assessing degree of learning of knowledge components. The most common services that an inner loop may offer, according to (5), are following listed:

– minimal feedback on a step. Thus, the tutor indicates whether the step is correct or incorrect;

– error specific feedback on incorrect steps. This information is intended to help the student to focus on which particular step or knowledge component is wrong and how to avoid making it again;

– hints on misconceptions or errors on specific knowledge components;

– assessment of knowledge.

It is important to emphasize that, the main aim of the services above listed is to assist students in learning the knowledge components in the learning domain. Designers may probably learn from the misused services under classic educational activities. Inner loop implements a “step analyzer” that is used for other services and interwoven with the outer loop. However, many other services could be implemented inside the inner loop structure. Once the outer loop has chosen a task, the MTCT waits the student’s action (step). After the student has committed it, the inner loop mechanism is triggered; the step analyzer in particular which is one of the services listed above, and assesses the degree of learning of the relevant knowledge components in the task (Sol_i):

$$Sol_i(NT) \in \{0,1\},$$
$$NT \in \{ NT, NT', NT^* \},$$

then updates the information about the attempt as well. The complexity level is adjusted according to the piecewise model in (11).

$$j=\begin{cases} j+1, \text{ if } (Sol_i(NT)=1) \ \& \ (j < \max(j)); \\ j-1, \text{ if } (Sol_i(NT)=0) \ \& \ (j < \min(j)); \\ j, \text{ other cases.} \end{cases} \quad (11)$$

A module is completed when the set of knowledge components that conform it are “learned”, thus a threshold pKW helps to estimate this. Statements (12) and (13) are used for determining if certain knowledge component has been probably learned.

$$MIN[MS2(S_q, KW_{kl})] > pKW. \quad (12)$$
$$AVG[MS2(S_q, KW_{kl})] > pKW. \quad (13)$$

According to [5] over a threshold pKW = 0.85, it can be determined that student probably has learned a certain knowledge component; besides, this value can be adjusted. Statement (12) corresponds to a more “rigid” tracing for all knowledge components, whereas (13) permits to scatter degree of learning of knowledge components.

Method for pedagogical feedback support. Inner loop is also responsible of supplying a “service” for pedagogical feedback; this service may be offered at the moment the student makes actions during attempting or after completing a task. Although, a hint could be supplied before, during or after attempting to complete the assigned task to support or assist the student in completing it as well. Hints are intended to avoid frustration or remarking repetitive misconceptions or mistakes. However, in this work, it is only proposed a general method for supplying pedagogical feedback after the student has submitted his answer. It can be used as a base for developing other supporting pedagogical methods, but this may increase complexity of the software to make it capable of tracking every minimal student’s action even over the MTCT’s graphic users interface for interpreting and “translate” it into a pedagogical intervention.

$\forall_k \ Sol_i(NT), NT \in \{ NT, NT', NT^* \}$

<u>Start</u>

Analyze: $\forall_l\{KW_{kl}\} : \{Sol_i(NT)\} \rightarrow [T_{ijk}]$

$\{Sol_i(NT)\} \leftrightarrow 1$

MS1 : $\{S_q\} \times \{T_{ijk}\} \rightarrow (\{N_{ikr}\}+1)$

Give: $\{\min(FB_l)\} : \{Sol_i(NT)\} \rightarrow 1;$

$\{Sol_i(NT)\} \leftrightarrow 0$

MS1 : $\{S_q\} \times \{T_{ijk}\} \rightarrow (\{N_{ikw}\}+1)$

$(\{N_{ikw}\} = 1) \rightarrow \{\min(FB_l)\}, \{T_{ijk}\} \rightarrow [k]$

Give: $\forall_l \{FB_l\} = 2 : \{N_{ikw}\} \in [2, 3], \{T_{ijk}\} \rightarrow [k]$

Give: $\forall_l \{FB_l\} = 3 : \{N_{ikw}\} > 3, \{T_{ijk}\} \rightarrow [k]$

<u>End</u>

The method for the pedagogical feedback support is above presented and following explained. When the student’s step is submitted, the step analyzer assesses the relevant knowledge components in the current task. In addition, it computes how many times the student has correctly employed a specific knowledge component (N_{ikr}); how many times he has misused it (N_{ikw}), and accordingly the inner loop returns some classification of feedback, (FB_l) ∈ {1: minimal feedback, 2: hint about error, 3: specific error feedback}.

For the first time a knowledge component has been misused the inner loop, after employing the step analyzer, will return a minimal feedback ($FB_l \rightarrow 1$), such as "correct" or "incorrect". For the second and third time that the step analyzer determines certain knowledge component has been misused in the current task, it will return an error-specific hint or feedback ($FB_l \rightarrow 2$). For instance in a fault-tolerant learning domain, "You should pay more attention on the value of the transfer coefficient" or "The class of fault you have chosen is not correct", "Static characteristics for this class of fault are depicted on the figure, identify them", etc. It has been determined that the inner loop will give second level feedback twice as a very simple mechanism to minimize feedback abuse. Nevertheless, other more advanced mechanisms may be implemented.

On the fourth and over of wrong attempts or misuse of a relevant knowledge component, the inner loop will return and error-specific feedback, leading the student to review and study the corresponding theory or related information to overcome the deficiencies on the corresponding knowledge components in order to prevent this from occurring again and supporting a constructive learning process. The inner loop gives only delayed feedbacks and hints in accordance with the policies explained above and it will only give them right after the student had submitted his step(s).

4. Experimental results and analysis

A MTCT was developed to implement and test the performance of the proposed models and methods [6].

The training program has been classified into three sequential modules, 29 relevant knowledge components were defined. As it was explained above, macroadaptation knows which knowledge components are required for each task.

Thus, for training the complete set of knowledge components, 43 tasks were developed. Moreover, some of these tasks have more than one variant; this feature increases the set of tasks up to 212 different tasks that the MTCT may present to the student and they are grouped by level of complexity as well.

Experimental results for evaluating the effectiveness of the models and methods were obtained by means of the analysis of 38 students' performance, separated in two groups as follows:

1) 19 students used the MTCT without the implementation of the outer and inner loops during the learning process (Group A);

2) 19 students used the MTCT with a fully implementation of the outer and inner loops (Group B);

Experimental results from the group of students that used the tutor without implementing the outer and inner loops mechanism are depicted on Fig. 3.

Average degree of learning of the Group A for each of the 29 knowledge components in the learning domain is clearly below the threshold pKW on the Fig. 3, and it states that the degree of learning of the knowledge components in the learning domain is less probably.

On the other hand, when the Group B used the tutor with the outer and inner loops implemented and obtaining adaptive pedagogical interventions the probability of learning every knowledge component considerably increased, and this result is shown on Fig. 4.

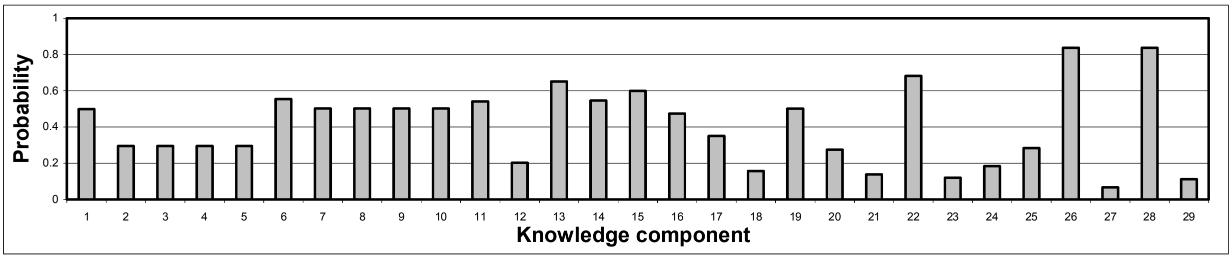

Fig. 3. Probability of learning of knowledge components without outer and inner loops mechanisms

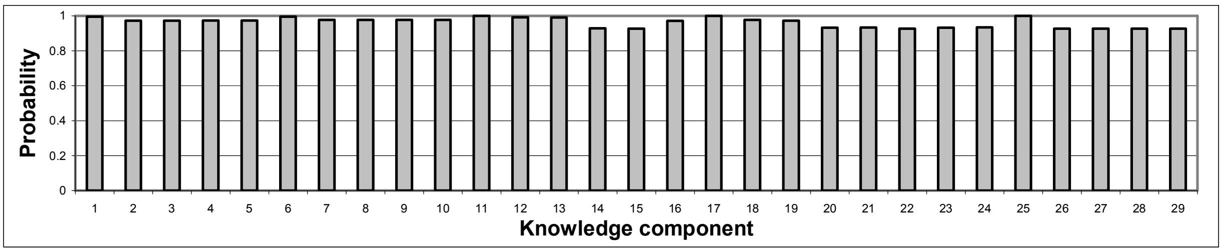

Fig. 4. Probability of learning of knowledge components with outer and inner loops mechanisms

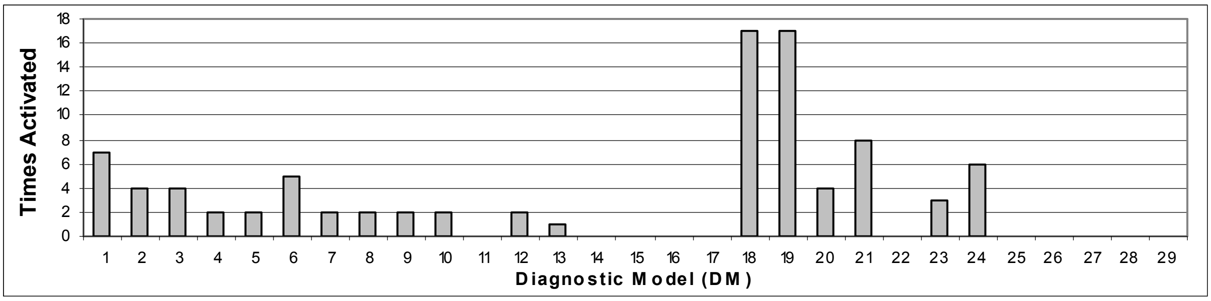

Fig. 5. Times knowledge components' diagnostic models (DM) were activated

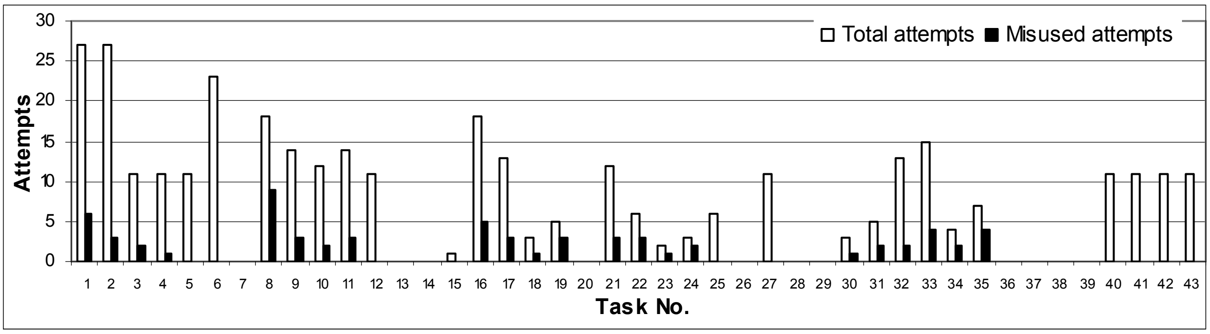

Fig. 6. Total and misused attempts for each task

Times that students misused a knowledge component is shown on Fig. 5, it shows times the DMs for each KW were activated when it was misused and states on it that knowledge components 18, 19 were difficult ones for students. On base of that, adjustments in the educational role must be done and pay attention on them in order to compensate these deficiencies.

Attempts of completing a task depicted on Fig. 6 say which tasks resulted problematic for students, but also demonstrates the adaptability of the outer loop in accordance to the student's performance and because of that, some tasks were not showed at all, however others were more often required on basis of their relevant KWs.

The students' final results from each group are depicted on Fig. 7. Students' numbers are just for a generic identification but there is not any relationship between the groups.

As Table 1 shows that by implementing the models and methods for pedagogical interventions Group B obtained an average ~0.56 higher probability of learning the knowledge components in the task domain than the Group A which did not obtained any pedagogical intervention.

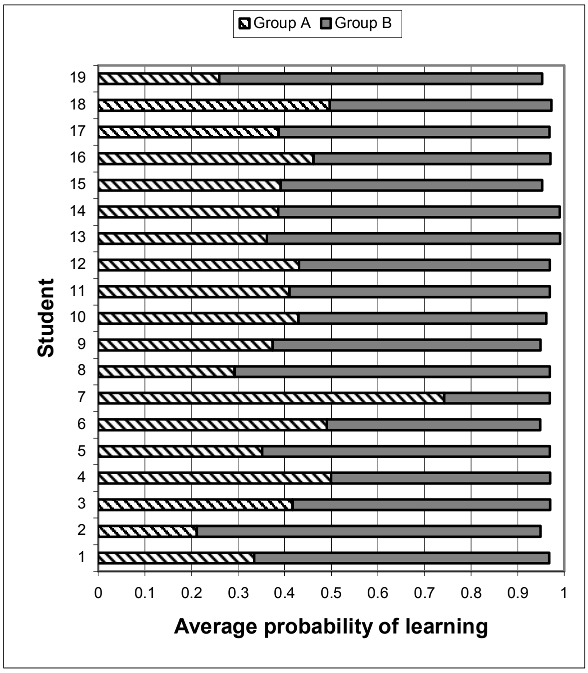

Fig. 7. Students' average probability of learning for the learning domain

Table 1

Average probability of learning of knowledge components in the learning domain

Group	Average probability
A	0.4068
B	0.9662

Conclusions

This paper proposes some models and methods for implementing into information technologies means for education, specifically in MTCT. An assessment model based on Bayesian networks with diagnostic models for making inferences for generation of pedagogical interventions was presented as well. A two-loop structure was described and the content of each component in it was in detail explained. This provides the learners with cognitive pedagogical support, like hints and feedback. It has the ability to build a student model from each student and generate individual pedagogical interventions based on it, in order to actively adapt the learning process according to the student's performance.

The implementation of the proposed models and methods demonstrates their own effectiveness based on the increment of the degree of learning of the relevant knowledge components in the learners. This effectiveness was obtained by implementing them into a MTCT that was employed with regular students in a master degree program of the learning domain. Students that received pedagogical interventions from the MTCT obtained a 42% better performance than those ones that did not receive any kind of assistance from the cognitive tutor. So students with better performance have a higher probability of having learned the relevant knowledge components and it proves the positive educational impact in students when the proposed approach is implemented in a MTCT.

References (GOST 7.1:2006)

1. The Affective Meta-Tutoring project: Lessons learned [Text] / K VanLehn, W. Burleson, S. Girard, M. E. Chavez-Echeagaray, Y. Hidalgo-Pontet, L. Zhang // Intelligent Tutoring Systems, 12th International Conference, ITS 2014. – Berlin, 2014. – P. 94-103.

2. Designing a Knowledge Representation Approach for the Generation of Pedagogical Interventions by MTTs [Text] / L. Paquette, J. F. Lebeau, G. Beaulieu, A. Mayers // International Journal of Artificial Intelligence in Education. - 2015. - Vol. 25. - P. 118-156.

3. Toward Meta-cognitive Tutoring: A model of Help-Seeking with a Cognitive Tutor [Text] / V. Aleven, B. McLaren, I. Roll, K. Koedinger // International Journal of Artificial Intelligence in Education. – 2006. - Vol. 16. – P. 101-130.

4. Aleven, V. Rule-Based Cognitive Modeling for Intelligent Systems [Text] / V. Aleven // Advances in Intelligent Tutoring Systems / Eds. R. Nkambou, J. Bourdeau, R. Mizoguchi. – Berlin : Springer-Verlag, 2010. - P. 33-62.

5. VanLehn, K. The Behavior of Tutoring Systems [Text] / K. VanLehn // International Journal of Artificial Intelligence in Education. – 2006. - Vol. 16, № 3. - P. 227-265.

6. Martinez Bastida, J. P. A learning platform for developers of faults-tolerant systems based on the signal-parametric approach [Text] / J. P. Martinez Bastida, A. G. Chukhray, E. V. Gavrilenko // Радіоелектронні і комп'ютерні системи. – 2012. - № 4 (68). - С. 40-46.

7. Bloom, B. S. Taxonomy of Educational Objectives: The Classification of Educational Goals, Volumen 1 [Text] / B. S. Bloom. – New York : David McKay Co. Inc., 1956. – 207 p.

8. Conati, C. Using Bayesian Networks to Manage Uncertainty in Student Modeling [Text] / C. Conati, A. Gertner, K. VanLehn // User Modeling and User-Adapted Interaction, Kluwer Academic Publishers. – Netherlands, 2002. - Vol. 12. – P. 371-417.

9. Чухрай, А. Г. Методологические основы интеллектуальных компьютерных программ, обучающих решению алгоритмических задач [Текст] : дис. … д-ра техн. наук : 05.13.06 ; защищена 14.03.14 / Чухрай Андрей Григорьевич. – Х., Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», 2013. – С. 104-118.

10. Russel, S. Artificial Intelligence: A Modern Approach [Text] / S. Russel, P. Norving. – Third Ed. – USA, Prentice Hall, 2010. – 1132 p.

References (BSI)

1. VanLehn, K., Burleson, W., Girard, S., Chavez-Echeagaray, M. E., Hidalgo-Pontet, Y., Zhang, L. The Affective Meta-Tutoring project: Lessons learned. *Intelligent Tutoring Systems, 12th International Conference, ITS 2014*, 2014, pp. 94-103.

2. Paquette, L., Lebeau, J. F., Beaulieu, G., Mayers, A. Designing a Knowledge Representation Approach for the Generation of Pedagogical Interventions by MTTs. *International Journal of Artificial Intelligence in Education*, 2015, vol. 25, pp. 118-156.

3. Aleven, V., McLaren, B., Roll, I., Koedinger, K. Toward Meta-cognitive Tutoring: A model of Help-Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education,* 2006, vol. 16, pp. 101-130.

4. Aleven, V. *Rule-Based Cognitive Modeling for Intelligent Systems.* Nkambou, R., Bourdeau, J. & Mizoguchi, R. (Eds.) Advances in Intelligent Tutoring Systems, 2010, pp. 33-62.

5. VanLehn, K. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education,* 2006, vol. 16, no. 3, pp. 227-265.

6. Martinez Bastida, J. P., Chukhray, A. G., Gavrilenko, E. V. A learning platform for developers of faults-tolerant systems based on the signal-parametric approach. *Radioelektronni i komp'uterni sistemi - Radioelectronic and computer systems*, 2012, no. 4 (68), pp. 40-46.

7. Bloom, B. S. *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain.* New York, David McKay Co. Inc., 1956. 207 p.

8. Conati, C., Gertner, A., VanLehn, K. Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User Modeling and User-Adapted Interaction, Kluwer Academic Publishers*, Netherlands, 2002, vol. 12, pp. 371-417.

9. Chukhray, A. G. *Metodologicheskie osnovy intellektual'nyh komp'juternyh programm,*

obuchajushhih resheniju algoritmicheskih zadach. Dys. … d-ra tekhn. nauk [Methodological bases of intelligent computer programs, training solving algorithmic problems. Dis. … Dr. techn. Sciences]. Kharkov, National Aerospace University N. E. Zhukovsky "KhAI", 2013, pp. 104-118.

10. Russel, S., Norving, P. *Artificial Intelligence: A Modern Approach*. USA, Prentice Hall, 2010. 1132 p.

Поступила в редакцию 8.01.2017, рассмотрена на редколлегии 15.02.2017

МОДЕЛИ И МЕТОДЫ ДЛЯ ВНЕДРЕНИЯ ПЕДАГОГИЧЕСКОГО ВЛИЯНИЯ В СЛЕДЯЩИХ КОГНИТИВНЫХ СИСТЕМАХ

Х. П. Мартинес Бастида, А. Г. Чухрай, Е. В. Гавриленко

В данной работе предложены формализованные методы и модели для внедрения в информационные технологии в области образования. Их отличительной особенностью является использование байесовских сетей для оценки и принятия решений, что позволяет управлять неопределенностью на формальной основе. Рассмотрено, каким образом байесовские сети целесообразно использовать в качестве механизма создания логического вывода для оценки уровня владения знаниями студентов и определения соответствующих педагогических мероприятий для продуктивного процесса обучения. Приведено подробное объяснение формализации процессов в области образования, а также некоторых методов для достижения этой цели.

Ключевые слова: информационная технология, байесовская сеть, педагогические мероприятия, модель трассировки, обучающие комплексы, байесовская оценка.

МОДЕЛІ ТА МЕТОДИ ДЛЯ ВПРОВАДЖЕННЯ ПЕДАГОГІЧНОГО ВПЛИВУ НА ВІДСТЕЖУВАЛЬНИХ КОГНІТИВНИХ СИСТЕМАХ

Х. П. Мартінес Бастіда, А. Г. Чухрай, О. В. Гавриленко

В даній роботі запропоновано формалізовані методи і моделі для впровадження в інформаційні технології в сфері освіти. Їх особливістю є використання баєсівських мереж для оцінки і прийняття рішень, що дозволяє управляти невизначеністю на формальній основі. Розглянуто яким чином баєсівські мережі доцільно використовувати в якості механізму створення логічного висновку для оцінки рівня володіння знаннями студентів і визначення відповідних педагогічних заходів для продуктивного процесу навчання. Приведено докладні пояснення формалізації процесів у галузі освіти, а також деяких методів для досягнення цієї мети.

Ключові слова: інформаційна технологія, баєсова мережа, педагогічні заходи, модель трасування, навчальні комплекси, баєсова оцінка.

Мартинес Бастида Хуан Пабло – аспирант кафедры «Системы управления летательными аппаратами», Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: jpbastida@hotmail.com.

Чухрай Андрей Григорьевич – д-р техн. наук, доц., профессор кафедры «Системы управления летательными аппаратами», Национальный аэрокосмический университет им. Н. Е Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: achukhray@gmail.com.

Гавриленко Елена Владимировна – канд. техн. наук, доц., доцент кафедры «Системы управления летательными аппаратами», Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: l.gavrilenko@mail.ru.

Martinez Bastida Juan Pablo – Ph.D. student of aircraft control systems, National Aerospace University named after N. E. Zhukovzky “KhAI”, Kharkiv, Ukraine, e-mail: jpbastida@hotmail.com.

Chukhray Andrey Grigoryevich – Doctor of Technical Science., Prof. of Aircraft Control Systems Department, National Aerospace University named after N. E. Zhukovzky “KhAI”, Kharkiv, Ukraine, e-mail: achukhray@gmail.com.

Gavrilenko Elena Vladimirovna - Candidate of Technical Science, Assistant Professor of Aircraft Control Systems Department, National Aerospace University named after N. E. Zhukovzky “KhAI”, Kharkiv, Ukraine, e-mail: l.gavrilenko@mail.ru.