# Developing a Pedagogical Intervention Support based on Bayesian Networks

J. P. Martínez Bastida[1], E. V. Gavrilenko[1], A. G. Chukhray[1]

[1] National Aerospace University, KhAI, Kharkiv, Ukraine
(jpbastida, lm77191220, achukhray)@gmail.com

**Abstract.** This paper proposes an approach for developing pedagogical interventions support in information technologies for education based on Bayesian networks. In this paper, we show how the presented approach is able to automate pedagogical interventions in Model-tracing cognitive tutors (MTCTs). The paper discusses a novel Bayesian network topology to assess student's mastery to provide pedagogical interventions. Preliminary results to assess effectiveness of the proposed approach were obtained by implementing it in a MTCT called TITUS.

**Keywords:** Bayesian networks, model-tracing, cognitive tutor, pedagogical intervention.

**Key Terms.** Modeling Systems in Education.

## 1 Introduction

Cognitive models (CM) are an integral part of developing Model-tracing cognitive tutors (MTCTs) [1, 2]. Various MTCTs have successfully been applied over the last decades, they are capable to trace the student's steps while he is interacting with the cognitive tutor and their implementation has proved a positive impact in the learners [1-4]. CMs require a proper understanding of the knowledge involved in a step (student's action), problem-solving strategies or principles in a given learning domain.

A CM should be able to interpret student's recurrent behavioral patterns and tendencies that reflect a way of thinking in order to provide constructive pedagogical interventions. Therefore, a MTCT is always "interested" on the way a student processes and assimilates the relevant knowledge components, the result of this can be called as the learner's meta-cognition model. This model is built by tracing and analyzing the actions when a student commits steps to accomplish certain task, but steps can be recurrent in terms of the way that knowledge is required, in other words; how tasks are presented. Interpretation for assessing mastery in students is a very important feature in an intelligent tutoring system (ITS) that involves uncertainty information. Moreover, assessment of mastery in a student and keep track of it require uncertainty reasoning, since this assessment leads to monitor cognitive processes that are not always explicitly observable. Bayesian Networks (BNs) are a widely used approach for uncertainty modeling in ITSs. This technique combines the rigorous probabilities formalism with a graphical representation and efficient inference mechanisms [5-7]. For implementing and testing the pedagogical interventions support proposed in this work, a Technical Intellectual Tutoring System (TITUS) [8] was developed. The curriculum in TITUS has been built in accordance with the signal-parametric approach for fault-tolerant systems [9].

This work is based on the hypothesis that some students are less able to look for help when they need it or get closer 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", besides a help-seeking student becomes a better learner [3]. Mainly, TITUS supports the base of learning by doing, help-seeking instructions and self-analyzing. These features have been tested in learning platforms and cognitive tutors and they prove to raise student's scores [2, 3, 6, 7].

## 2 Assessment model for determining mastery

Bayesian networks are a formalization to manage uncertainty and they have widely been employed in ITSs [5, 6]. BNs based on the Knowledge Tracing approach affect prior probabilities of mastery in Knowledge Components (KC) equally. Thus, when multiple KCs are involved in a step and the step is incorrect, all probabilities of mastery will equally decrease in every KC involved in the step, without taking in account if they were or were not misused. BN presented on Fig. 1 implements a Diagnostic Model (DM) that improves assessment of mastery in the case above exposed. This topology assumes that each step depends on individual KCs. Thus, the set of relevant KCs in a step are individual cognitive processes; when a student attempt to complete a task, KCs can be applied independently one from another, so their posterior probability of mastery should be assessed separately. This BN consists of four nodes: $K_t$, $S_{t+1}$, $DM$ and $K_{t+1}$, where $K_t$ is the probability of mastery of certain KC or skill at $t$ time; $S_{t+1}$ is a step at moment $t+1$; $DM$ is a diagnostic model that is directly linked to the step and influences the assessment of mastery; and $K_{t+1}$ is the probability of mastery at $t+1$ moment. $\neg K_t$, $\neg S_{t+1}$, $\neg DM$ and $\neg K_{t+1}$ are the respective complementary probabilities of mastery.

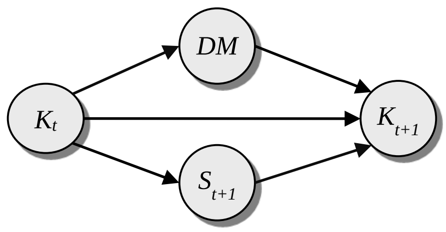



**Fig. 1.** BN with Diagnostic Model for Knowledge Tracing

The probability $P(K_{t+1})$ of mastery certain knowledge component at $t$ moment after a student's correct step is obtained with (1).

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

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, where α is a normalization coefficient. The evidences in a student's action are denoted by $P(S_{t+1}) = 1$ (correct step) and $P(DM) = 0$ (deactivated).

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

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

Therefore, a step analyzer assesses each relevant KC in the actual step in order to determine the corresponding pedagogical actions.

## 3 Model for selecting the next to do

Implementation of the model for selecting a task requires a set of tasks separated by sequential learning modules and complexity levels. Under the macroadaptation approach, three or five levels of complexity are commonly instantiated as standard for educational proposes [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 in each level of complexity, with the aim to have alternatives of choice. Moreover, all the set of tasks in a module must cover the complete set of relevant KCs included in it, and they should be trained more than once at each level of complexity.

Set of tasks in every module should be developed as an interwoven network over the relevant KCs that it contains. Thus, it is preferable that every KC should be trained at least by two different tasks. This relationship between a KC and tasks increases the probability of mastering 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) is represented in (4) and its boundaries in (5)-(7), 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 for the task $T$, and $l$ is the identification number for the knowledge component. An example of the MT above explained is depicted on Fig. 2.

$$MT: \{T_{ijk}\} \rightarrow \{KW_{kl}\} \tag{4}$$

$$\forall k, \forall j \; \{T_{ijk}\} \neq \emptyset, \|T_{ijk}\| \geq 2 \tag{5}$$

$$\forall k, \forall j, \forall l \; \{T_{ijk}\} = MT^{-1} \; (KW_{kl}) \neq \emptyset, \|T_{ijk}\| \geq 2 \tag{6}$$

$$\forall k \; \cup MT \; (T_{ijk}) = \{KW_{kl}\} \tag{7}$$

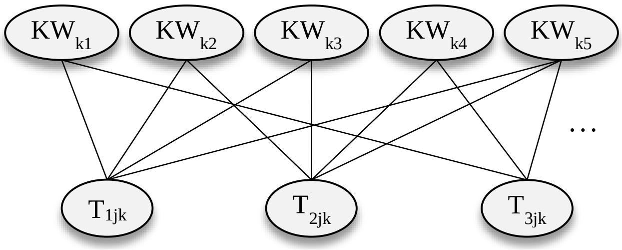



**Fig. 2.** Task model structure (example)

On the other hand, the student model (*MS*) is constantly updated while the student is working with the ITS, for this reason, *MS* is a dynamic representation of the student. *MS* can be represented by (8) and (9), where $S$ represent the student, $q$ is his identification number, $P \subset \Re$ in the interval [0, 1] that represents the probability of mastery, $N$ are the attempts (steps) realized.

$$MS1: \{S_q\} \times \{T_{ijk}\} \rightarrow N \tag{8}$$

$$MS2: \{S_q\} \times \{KW_{kl}\} \rightarrow P \tag{9}$$

The prior information is initialized if a student $S_q$ uses the ITS 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. Therefore, a next task (*NT)* with KCs that have lower probabilities of mastery among tasks in a module (*MZ*) is chosen by means of (10).

$$NT = MT^{-1}\left( KW_{kl} \middle| MS2\left(S_q, KW_{kl}\right) \rightarrow min \right) \tag{10}$$

If $\|NT\| > 1$, thus, search of the next task will be based on attempts *NT'* and represented by (11). In case $\|NT'\| > 1$, it will be implemented by (12) and a task will randomly be selected (*NT**). This case is certainly possible at the first time a student uses the ITS.

$$NT' = NT\left( KW_{kl} \middle| MS1\left(S_q, T_{ijk}\right) \rightarrow min \right) \tag{11}$$

$$NT^* = RAND(NT) \tag{12}$$

Processes described by (10)-(12) will repeat meanwhile the student has not mastered the KCs in the current module; only then, the ITS passes to the closest upper module: $k + 1$ and again it accordingly repeats the processes of choosing a next task until $k < max(k)$.

# 4 Models for defining complexity level and assessing probability of mastery

Once a task has been chosen, the ITS waits a step. After the student has committed it, the step analyzer is triggered and assesses probability of mastering the relevant KCs in the task: $Sol_i(NT) \in \{0,1\}$, $NT \in \{ NT, NT', NT^* \}$, and updates the attempt as well. The complexity level is adjusted according to the piecewise model in (13).

$$j = \begin{cases} j + 1, if\left(Sol_i(NT) = 1\right)\left(j < max(j)\right) \\ j - 1, if\left(Sol_i(NT) = 0\right)\left(j < min(j)\right) \\ \qquad j, other\ cases \end{cases} \tag{13}$$

A module is completed when the KCs that conform it are mastered, thus a threshold value ($pKW = 0.85$) helps estimating it [5]. Expressions (14) and (15) are used for determining probability of mastery.

$$MIN\left[MS2\left(S_q, KW_{kl}\right)\right] > pKW \tag{14}$$

$$AVG\left[MS2\left(S_q, KW_{kl}\right)\right] > pKW \tag{15}$$

# 5 Method for pedagogical feedback support

Pedagogical feedback is a “service” that may be offered at the moment the student makes steps. Although, a hint could be supplied before, during or after committing a step to support or assist the student. Hints are intended to avoid frustration or remarking repetitive misconceptions or error patterns.

However, in this work, it is only proposed a general method for supplying pedagogical feedback after the student has submitted a step. Nevertheless, it can be used as a base for developing other supporting pedagogical methodologies, but this may increase complexity of the software to make it capable of tracking every minimal student’s action even over the tutor’s GUI for interpreting and “translate” it into a pedagogical intervention. The method for the pedagogical feedback support is executed when the student’s step is submitted and the step analyzer already assessed the relevant KCs involved in the current task.

```
  ∀k Sol_i (NT), NT ∊ { NT, NT', NT*}
Start
  Analyze: ∀l{KW_kl} : {Sol_i (NT)} → [T_ijk]
     {Sol_i (NT)} ↔ 1
        MS1 : {S_q}x{T_ijk}→({N_ikr}+1)
        Give: {min(FB_l)} : {Sol_i (NT)} →1

     {Sol_i (NT)} ↔ 0
        MS1 : {S_q}x{T_ijk}→({N_ikw}+1)
        ({N_ikw} = 1) → {min(FB_l)}, {T_ijk} → [k]
        Give: ∀l {FB_l} = 2 : {N_ikw} ∈ [2, 3], {T_ijk}→[k]
        Give: ∀l {FB_l} = 3 : {N_ikw} > 3, {T_ijk} → [k]
End
```

In addition, it computes how many times the student has properly employed a specific KC ($N_{ikr}$); 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 KC is misused, a minimal feedback ($FB_l \rightarrow 1$) is returned, such as “correct” or “incorrect”. For the second and third time, it will return an error-specific hint or feedback ($FB_l \rightarrow 2$), i.e. “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 second level feedback should be given twice as a very simple mechanism to minimize feedback abuse. Nevertheless, other more advanced mechanisms may be implemented.

On the fourth and over a misuse of a relevant KC has occurred, the tutor 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 KCs in order to prevent this from occurring again and supporting a constructive learning process. The tutor gives only delayed pedagogical feedback support in accordance with the policies explained above and it will only give them right after the student had submitted his step.

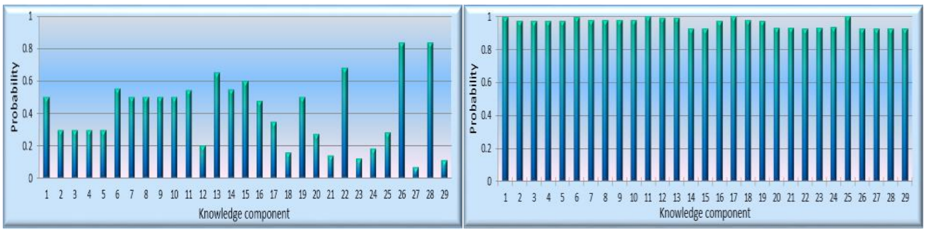
## 6 Implementation and experimental results

TITUS [8] was developed to implement and test the performance of the proposed approach. The training program has three sequential modules and 29 relevant knowledge components. Thus, for training the complete set of KCs, 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 TITUS may present to the student and they are grouped by level of complexity as well.

Experimental results for evaluating the effectiveness of the pedagogical interventions provided by TITUS, were obtained by means of the analysis of 38 students' performance, separated in two groups as follows:

1. 19 students used TITUS without any kind of pedagogical support during the learning process (Group A);
2. 19 students used TITUS with a full implementation of the pedagogical support (Group B);

Experimental results from Group A are depicted on Fig.3(a). Average probability of mastery for KCs is clearly below the threshold *pKW*. On the other hand, when Group B used TITUS, the probability of mastery for every KC considerably increased, and this result is shown in Fig. 3(b). Times when student has misused a knowledge component are shown in Fig. 4.




(a) (b)

**Fig. 3.** Probability of mastery of Group A (a) and Group B (b)

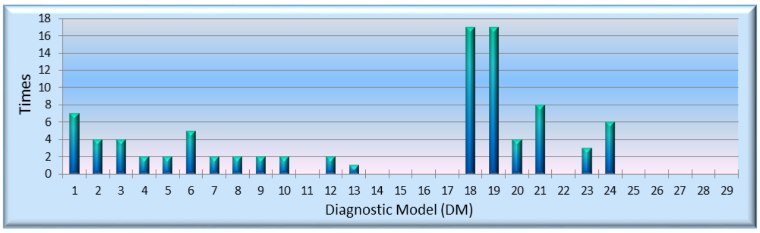



**Fig. 4.** Times DMs were activated for each Knowledge Component

Attempts in Fig. 5 say which tasks resulted problematic for students, but also shows the adaptability of the proposed approach and how it was developed according to the student's performance.

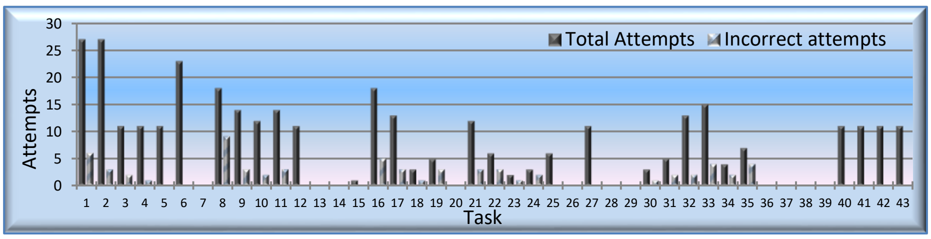



**Fig. 5.** Total and incorrect attempts for each task

# 7 Conclusions

This paper proposes an approach for developing pedagogical interventions support in information technologies for education. A novel assessment model based on Bayesian networks for providing pedagogical interventions was presented as well. It provides learners a cognitive pedagogical support, like hints and feedback. It has the ability to build a student model from each student and provide individual pedagogical interventions based on it, in order to actively adapt the learning process according to the student's performance.

Results demonstrate effectiveness of the approach based on the increment of mastery in learners. This effectiveness was obtained by developing a MTCT called TITUS that was employed with regular students in a master degree program of the task domain. Students that received pedagogical interventions obtained a 42% better performance than those ones that did not receive any kind of assistance, and it proves the positive educational impact in students when the proposed approach is implemented in a MTCT. In the near future, we expect to develop an extended version of the BN model and pedagogical feedback support by including for instance help abuse among others.

# References


1. Paquette, L., Lebeau, J. F., Beaulieu, & G., Mayers,: A. Designing a Knowledge Representation Approach for the Generation of Pedagogical Interventions by MTTs. In: International Journal of Artificial Intelligence in Education, vol. 25, pp. 118-156. (2015)
2. Aleven, V., Nkambou, R., Bourdeau, J. & Mizoguchi, R.: Rule-Based Cognitive Modeling for Intelligent Systems. In: Advances in Intelligent Tutoring Systems, pp. 33-62. (2010)
3. Aleven, V., McLaren, B., Roll, I., & Koedinger, K.: Toward Meta-cognitive Tutoring: A model of Help-Seeking with a Cognitive Tutor. In: International Journal of Artificial Intelligence in Education, vol. 16, pp. 101-130. (2016)
4. Pelánek R., & Jarusek P.: Student Modeling Based on Problem Solving Time. In: International Journal of Artificial Intelligence in Education, vol. 25, pp. 493-519. (2015)
5. Conati, C., Gertner, A., & VanLehn, K.: Using Bayesian Networks to Manage Uncertainty in Student Modeling. In: User Modeling and User-Adapted Interaction, Kluwer Academic Publishers. vol. 12, pp. 371-417. Netherlands (2002)
6. VanLehn, K.: Intelligent Tutoring Systems for Continuous, Embedded Assessment. In: The future of assessment: Shaping teaching and learning. (eds.) Carol Anne Dwyer. pp. 113–138. Lawrence Erlbaum Associates, New York (2008)
7. 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. In: 12th International Conference, ITS 2014, pp. 94-103. Berlin (2014)
8. Martinez Bastida, J.P., Chukhray, A.G.: An Adaptive Learning, Technical Intelligent Tutoring System for Signal-Parametric Fault-Tolerant Systems. In: Radioelectronic and Computer Systems. vol. 4(74), pp. 139-144. (2015)
9. Kulik, A.S., Fault diagnosis in dynamic Systems via signal-parametric approach. In: IFAC/IMACS Symposium of fault detection, supervision and a technical process, SAFE PROCESS 91. vol. 1, pp. 157-162. Baden-Baden (1991)