An Open and Inspectable Learner Modeling with a Negotiation Mechanism to Solve Cognitive Conflicts in an Intelligent Tutoring System

Evandro Costa[1], Priscylla Silva[1], Jonathas Magalhães[2], and Marlos Silva[2]

[1] Federal University of Alagoas - Alagoas - Brazil
{evandro,pmss}@ic.ufal.br
[2] Federal University of Campina Grande - Paraíba - PB - Brazil
{jonathas,marlos}@copin.ufcg.edu.br

Abstract. Some researchers have developed relevant and diverse proposals for improving the content quality of the learner model in Intelligent Tutoring Systems, mainly reducing its uncertainty. Following this aim, this paper proposes an open learner modeling approach using Bayesian networks, focusing on negotiation mechanism to solve detected cognitive conflicts that can emerge when the learner inspects information of his model inferred by the system. Therefore, we addressed some issues concerning the provision of inspectable model and negotiated updating of this model. Its contribution lies in the fact that the learners attempt to change the learner model is met with a challenge, leading to a decision if the learner claims to know more (or less) than the model represents.

Keywords: Open Learner Model, Negotiation mechanism, Bayesian Networks, Intelligent Tutoring Systems

1 Introduction

The field of Intelligent Tutoring Systems (ITSs) claims to be adaptive, yielding software systems that provide individualized instruction by creating and maintaining model of their learners. Some proposals for these models have been hidden from the learner. However, other research approaches were expanded to open the learner model, allowing the learner to inspect his model thereby facilitating reflection, which is known to enhance the learning process [6]. Following this trend many challenges have been faced in this research direction in order to improve even more the accuracy of the model. Open learner modeling in ITSs has been the focus of several studies in the literature. However, little work has been done on developing interactive updating procedures of the learner model, involving the learner and the system. Proposals for open learner modeling have been considered by some work in the literature (e.g., [1, 6, 8]), but some of them do not include negotiation mechanism to solve potential conflicts in their approaches. However, experiments performed in [2, 5, 7] have shown that the provision of negotiated student models are potentially useful because they can help the student to reflect on their knowledge.

This paper proposes an open learner modeling approach using Bayesian networks, focusing on negotiation mechanism to solve detected cognitive conflicts that can emerge when the learner inspects information of his model inferred by the system. Alternatively, these conflicts can be also detected by the system. Here, it is important to say that the mentioned conflicts are identified in the context of problem solving situations when the learner and the system have conflicting evaluations about the solution of a given problem. Then, some negotiation method is necessary to solve these conflicts. To this end, we addressed some issues concerning the provision of inspectable model and negotiated updating of this model.

2 The Learner Model and The Negotiation Mechanism

The model is divided into two parts, one to represent the tutor's belief in learner's knowledge M_t and the other part represents the learner's belief in himself, denoted by M_s. The M_t and M_s are represented by Bayesian Networks (BNs), they have the same structure, their difference is how the evidence is inputted. In addition, these BNs are constructed taking into account the curriculum structure proposed in [4]. We utilize one Dynamic Belief Network (DBN) to represent the learner model with two parts [3]: i) *Domain-general* knowledge, to represent the entire learner's domain knowledge, and ii) *Task-specific*, to represent the learner's knowledge in a specific problem.

The part of *Domain-general* is built from the domain definition for each learner, the probabilities of this part of the model are utilized by the tutor to the assessment of the learner and to the process of negotiation. The part of *Task-specific* is generated in run time and it is built from the problem characteristics. When a learner finishes to solve a problem, the probabilities of the part of *Task-specific* are updated, so, the nodes related with the task-specific part are pruned and the marginal probabilities are used to update the part of *Domain-general* [3].

The knowledge of the learner is modeled in the BN of domain-general knowledge. The Figure 1 presents the general structure of the BN, its nodes and edges. The BN contains two types of node: i) *Pedagogical Unit (PU)* node represents a skill of the learner. This node can be the parent of a set of other *Pedagogical Unit* nodes or a collection of *Problem Set* nodes; ii) *Problem Set (PS)* node that represents the learner's knowledge in a specific set of problems. Every node has two values, mastered or not mastered, that measure the learner's knowledge.

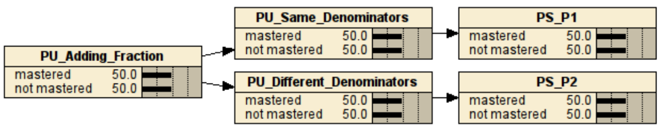

Fig. 1. The BN Structure of the *Domain-general* part of DBN Model

The part of *Task-specific* models the learner's knowledge in a specific problem. This part of DBN contains a *Problem (P)* node that represents the probability of learner solve it and one *Problem Set (PS)* node, that is borrowed from *Domain-general* BN. Both nodes have two values, mastered or not mastered. The *Problem Set* node is parent of a set of *Problem* nodes. When the learner finishes a problem, the *Problem* node is pruned and the probabilities of the *Problem Set* node are roll-up and the *Domain-general* BN is updated.

The weights of the Bayesian Network are given by the Knowledge Engineer (responsible for modeling the domain knowledge). To set the prior probabilities, we established five degrees of knowledge: *No idea*=0.05, *Basic*=0.25, *Good*=0.50, *Very good*=0.75 and *Expert*=0.95. Intuitively, this indicate the prior knowledge that the system/teacher puts on a learner. To set the table of conditional probabilities, we follow the approach proposed by Zapata-Rivera and Greer [9]. The M_t takes the learners performance in solving problems, so the evidence is captured and propagated of the following way. When a learner submits a solution for a problem p_1, the tutor verifies his solution and returns a value $v \in [0, 1]$ referents the learner's grade in p_1. So, v is used as evidence for the *Task-specific* part of M_t, the calibration of the Problem node P_1 referents to the problem p_1 is set of the following way: $P(P_1 = mastered) = v$. The M_s model captures the learners belief in themselves, this belief indicates the trust in the solution that he submitted. The evidence is captured of the following way: after solve a question, the tutor ask to learner what is his belief in his solution. The learner marks his belief according to these degrees: *Very unsure*=0.05, *Unsure*=0.25, *Almost sure*=0.5, *Sure*=0.75 and *Very Sure*=0.95. This approach is based on the previous work presented in [2]. So, the propagation of evidence is the same of the M_t model. The learner will have access to his model, to achieve a good visualization of the learner Model, we utilize the same approach used in ViSMod [9], the Figure 2 presents a visualization of the learner Model.

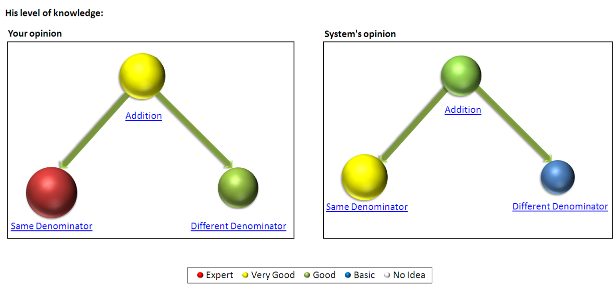

Fig. 2. The Open Learner Model, the Ms on the Left and Mt on the Right.

The negotiation process can be started by the learner or by the system. The learner might initiate a negotiation process when he disagrees with tutor's belief about a pedagogical unit, this can occur for various reasons, e.g. the learner learned outside the tutors environment the knowledge about some Pedagogical Unit. The tutor always starts a negotiation process when it detects a cognitive conflict. A cognitive conflict occurs when the difference between tutor's belief and the learner's belief exceeds some threshold. By interacting with the learner in a process of negotiation, the tutor must select one of the pre-defined negotiation strategies. This strategy determines the behavior of the system during the negotiation. It depends of two factors: i)student's credibility; and ii)the new belief required by the student, if the student is trying to change the system's belief or if the tutor is trying to change the student's belief.

The learner's credibility is modeled by a DBN, see Figure 3. Where the *Prob-*

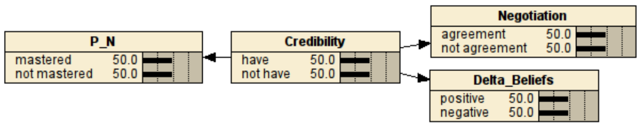

Fig. 3. The DBN of the Learner's Credibility.

lem node P_N represents the problem that the student has solved, this node is borrowed from *Task-specific* part of the M_t; the *Negotiation* node represents the negotiation status when it finalized, it has two binary values, agreement or not agreement; the *Delta-Beliefs* node represents the difference between tutor's beliefs and student's beliefs, it has two values: positive and negative. The calibration of *Delta-Beliefs* node is given by: P(Delta-Beliefs = positive) $= tanh(\delta b)$, where *tanh* is the hyperbolic tangent that serves to map the δb to [0,1] and δb is defined by: $\delta b = \sum_{pu \in M_t} P(pu = Mastered)^2 - \sum_{pu \in M_s} P(pu = Mastered)^2$.

The prior and the conditional probabilities are set as we explain in Subsection 2. The prior of the *Credibility* node is set with the threshold *Good*. The conditional probabilities are calculated according the degrees of influence between nodes, the *Credibility* node has a *very weak* influence on *Problem* node and a *Weak* influence on the *Delta-Belief* and *Negotiation* nodes. The credibility is updated every time a new evidence is captured, a problem has been solved or a negotiation has been finished.

The interaction between the student and the system is through dialogues menu and are controlled by dialogue rules, this movements of dialogue are based on previous work presented in [5]. These rules also define which movements of dialogue will be shown in the menu, so the student can interact in each movement of the negotiation. For each movement of dialogue executed by the student, the system is going to answer with another movement of dialogue which will be determined by the negotiation strategy adopted by the system. There are three cases to the learner starts a dialogue: requires an explanation about a belief, requires a change of the tutor's belief or suggests a change of the tutor's belief. If the learner requires the tutor an explanation, the tutor will inform him about the reasons that led it to establish that belief to him. The explanation can be, e.g.

statistical data about learner's performance. During the process of negotiation, the tutor may request to the learner prove that the tutor's belief is wrong, alternatively, the learner will be able to ask the tutor the opportunity to prove.

The proof process consists of two problems: i) an analysis problem (where the solution of a problem will appear to the learner and it should indicate whether the solution is right or wrong), ii) a problem of synthesis (where a problem will be given to the student to solve). The learner will have two chances to solve each problem. The problems are chosen according with the similarity to problems that the learner had made mistakes. After the test process in the time $T+1$, the learner will have a grade g in the range [0,1] associated with his performance. The tutor's belief $P(PU)_{T+1}^{(M_t)}$ will be update by the following Equation: $\mathrm{P(PU)}_{T+1}^{(M_t)} = P(PU)_T^{(M_t)} + \alpha * g * (P(PU)_T^{(M_s)} - P(PU)_T^{(M_t)}), where \mathrm{P(PU)}_T^{(M_t)}$ is the tutor's belief before the learner submits the test, $P(PU)_T^{(M_s)}$ is the learner's belief before test and $\alpha = 0.5$ is a smoothing parameter. The learner will have two opportunities to submit the solutions. After that, the tutor's new belief will be exposed to the learner, and he will be able to agree or not. In the case of agreement, the negotiation will be finalized, otherwise, the negotiation will continue until the learner accepts the tutor's beliefs through the proof process or until one the part involved quit, finalizing negotiation.

Likewise for the student, the tutor has three possible ways to start a negotiation: i)requires a justificative for a belief, ii)requires the change of a belief or iii)suggests a change of a belief. In the case of the learner's belief be lower than tutor's belief, the tutor will adopt the strategy of *Support* during negotiation. This strategy consists in try to prove to the learner that he has more knowledge about a Pedagogical unit than he suppose have and will try to convince him to change his belief to equal to the tutor's belief. In the case of learner's belief be greater than tutor's belief, the strategy will be *Persuasion*. So, the tutor will try to prove that the learner is overestimating his knowledge and will try to convince him to decrease his belief until it equal to the tutor's belief. In both cases, to convince the learner, the tutor will be able to utilize statistical data and/or show problem solutions that the student has submitted with their respective diagnosis. When the learner inspect his model, the tutor will show his learner model like showed in Figure 2. If the learner disagrees with the system's opinion about his knowledge in any pedagogical unit, he can start a negotiation dialogue with the system. During the dialogue, the system will adopt a strategy of negotiation, which may be modified during the dialogue. The Figure 4 shows an example of negotiation dialogue started by the learner.

3 Conclusion and Future Work

Overall, the main contribution of our work is two-fold: the representation of the learner through a Bayesian networks associated with the curriculum structure and the provision of an updating learner model method that includes a negotiation mechanism to solve cognitive conflicts that emerge from problem solving

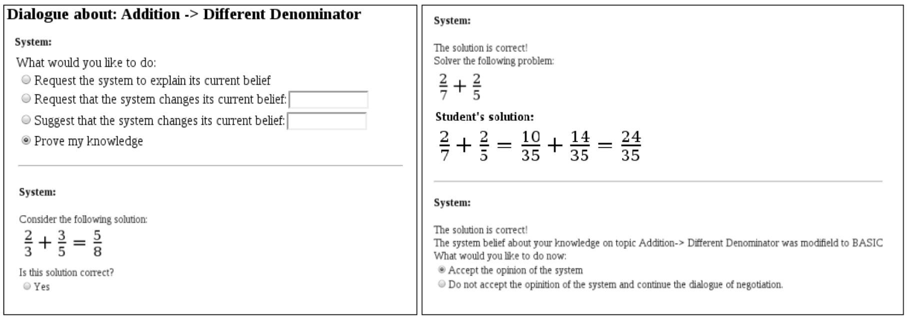

Fig. 4. Example of Negotiation Dialogue Started by the Learner.

evaluation processes. Therefore, we make a contribution to open learner modeling, advancing to existing approaches in the literature. The preliminary results obtained in a case study by using a complete scenario worked on our developed fraction tutor, revealed the value of our modeling approach. For future work, we will conduct experiments in a basic math classroom to better evaluate the effects of our approach on reflection and learning processes.

References

1. Peter Brusilovsky, Elmar W. Schwarz, and Gerhard Weber. Elm-art: An intelligent tutoring system on world wide web. In *Proceedings of the Third International Conference on Intelligent Tutoring Systems*, pages 261–269. Springer-Verlag, 1996.
2. Susan Bull, Paul Brna, and Helen Pain. Extending the scope of the student model. *User Modeling and User-Adapted Interaction*, 5(1):45–65, 1995.
3. Cristina Conati, Abigail Gertner, and Kurt Vanlehn. Using bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417, 2002.
4. Evandro de Barros Costa, Angelo Perkusich, and Edilson Ferneda. From a tridimensional view of domain knowledge to multi-agent tutoring system. In *SBIA*, pages 61–72, 1998.
5. Vania Dimitrova. Style-olm: Interactive open learner modelling. *International Journal of Artificial Intelligence in Education*, 13(1):35–78, January 2003.
6. Antonija Mitrovic and Brent Martin. Evaluating the effect of open student models on self-assessment. *International Journal of Artificial Intelligence in Education*, 17(2):121–144, April 2007.
7. David Thomson and Antonija Mitrovic. Preliminary evaluation of a negotiable student model in a constraint-based its. *Research and Practice in Technology Enhanced Learning*, 5(1):19–33, 2010.
8. Kalina Yacef. The logic-ita in the classroom: A medium scale experiment. *International Journal of Artificial Intelligence in Education*, 15(1):41–62, January 2005.
9. Juan-Diego Zapata-Rivera and Jim E. Greer. Interacting with inspectable bayesian student models. *International Journal of Artificial Intelligence in Education*, 14(2):127–163, 2004.