Hierarchical Representation and Evaluation of the Student in an Intelligent Tutoring System

Joséphine M.P. Tchétagni and Roger Nkambou

Université du Québec à Montréal
tchetagni.josephine@uqam.ca nkambou.roger@uqam.ca

Abstract. In this paper, we present an approach to hierarchical knowledge representation for the student's evaluation in propositional logic. The *hierarchical evaluation* consists in assessing the student's state of knowledge at several levels of granularity. The relevance of the method is justified by the need for a precise and flexible diagnosis of the learner's skills in a given domain. For that purpose, we shall model the propagation of the evaluation from a specific level of knowledge content to more general levels, using Bayesian inferences and neural networks classifications.

1 Introduction

"Intelligence" in an intelligent tutoring system (ITS) is ensured with tools which enable efficient management of the available information. One of the keys is the system capability to provide the learner with a personalized, adaptive but effective teaching. Thus, to adopt a suitable strategy, these teaching features require the system to be aware of the cognitive and behavioural skills associated to a particular student, to diagnose the student's errors or misconceptions, and to adjust the system beliefs about his current state of knowledge. This paper proposes a hierarchical representation of the student's model in an ITS (McCalla et al. [5]). We aim at showing that this approach can provide an efficient support for a global evaluation and a precise diagnosis. Our objective here is to validate this assertion by defining and justifying this idea, using propositional logic as the matter to be learned. In what follows, section 1 reviews student modeling methods, section 2 explains our approach and we show why it is relevant in section 3. Section 4 gives an outline of our future works concerning the validation aspects.

2 The Student Model: Related Works

The aim of the student's model (SM) should be to guide the tutor in taking the teaching decisions that are best adapted to a learner.However, a comprehensive student modeling is a difficult task, as these decisions must take into account several factors: the learner's current knowledge and behavioural characteristics, the goals of the training session, etc. These parameters are not easily assessable

S.A. Cerri, G. Gouardères, and F. Paraguaçu (Eds.): ITS 2002, LNCS 2363, pp. 708–717, 2002.

from a man-machine interaction, thus various possibilities may be available when designing the SM. In the SM, the first question to be answered is *what* is to be represented. Here two philosophies of knowledge representation exist: *state models* where the student's knowledge is represented at given stages of the learning process and *procedural models* (Anderson et al. [1]) where the process through which the student solves problems is represented. State models are adapted in learning concepts while procedural models are more appropriate in acquiring skills. Overlay models (Carr and Goldstein [2]) and buggy models (Fung [4]) are knowledge representation approaches that determine *how* to express the student's knowledge. In overlay models, the student's knowledge is considered as a subset of the domain knowledge which should be incremented. However, buggy models further enable the modeling of faulty information in system knowledge. A more recent approach consists in representing knowledge as a set of contextual constraints that the student behaviour or responses should comply with (Mitrovic et al [6]). When implementing these concepts, Bayesian graphs (Conati et al.[3]) provide an intuitive approach to diagnosis, our main concern in this paper. They make it possible to represent in specific contexts, the dependencies between different elements of knowledge in term of posterior probabilities.

3 Our Approach

Our aim here is not to propose a new domain knowledge representation approach, but rather, to *position the learner's state of knowledge* with respect to a knowledge element, that intervenes in a given domain. Assuming that a knowledge base already exists in the system, we have developed from this base, a hierarchical structure including the items of interest for a given training session, in order to view the student knowledge with respect to this hierarchy, by mimicking it.

3.1 Hierarchizing the Learning Content

When converting domain knowledge into a hierarchical representation, five levels of granularity can be identified, namely the domain (level 1), the subjects (level 2), the sub-subjects (level 3), the concepts (level 4) and the exercises (level 5) through which the student acquires concepts. Level 4 comprises *primitive or basic concepts* and *generic or composite concepts.* For instance, concepts such as *logical conjunction, logical disjunction* and *logical negation* are primitives while the *logical implication* is generic. In fact, a composite concept is a knowledge unit the definition (intuitive or formal) of which is based on a combination of several basic concepts. In this hierarchy, the nodes in each level refer to a knowledge unit. For example, Figure 1. shows at level 1 the domain of *mathematics*, at level 2, the subjects of *logics* and *integral calculus*, at level 3 the sub-subjects of *propositional logic*, *predicate calculus, fuzzy logic*, at level 4 the concepts of logical *OR, AND, NOT* and at level 5, there are some exercises linked to these concepts. Indeed, the links represent the decomposition of a given knowledge unit into more specific ones (*hierarchical or external links*), but they can also model the dependencies between elements of a single granularity (*internal links*). For example at the sub-subject level, the forward link between *fuzzy logic* and *propositional logic* models the fact that the latter is a pre-requisite to learn the former, while the forward links between *propositional logic* the logical concepts *OR, AND, NOT* model the decomposition of the sub-subject into concepts. Furthermore, the primitive concepts will be connected to basic exercises, while the generic ones will be connected to more elaborate exercises, which are characterized by a synthesis of the basic knowledge acquired by solving related basic exercises. It is therefore necessary to express the pre-requisite relations even at this bottom level. For example, problems concerning *implication* or *conjunctive normal form* (CNF) should be considered only after the solving of problems involving basic logical operators.

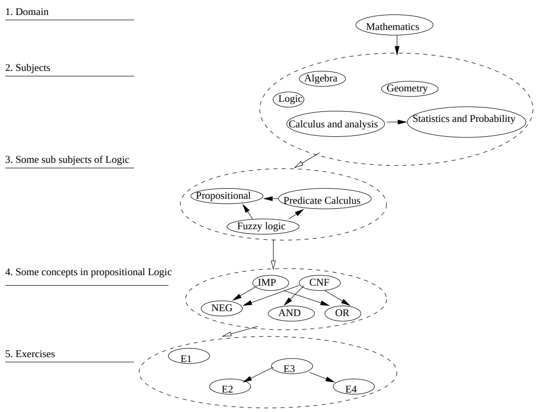

Fig. 1. A hierarchical domain knowledge view

Table 1. A hierarchical model for domain knowledge representation

Level	Designation	Information			
1	Labels	Mathematics			
2	Labels	Logic			
3	Labels	Proposition		Predicate	
	Meta-K[1]	*Probability*(Masters Predicate Level_2)/(Masters Boole Level_2) = 0.5			
4	Labels	Primitive(OR)	Primitive(NEG)	Generic(IMP)	Generic(FNC)
	Meta-K	*Probability*(Masters IMP Level_1) / (Masters (OR, NEG) Level_1)= 0.8 *Probability* (Masters EQU Level_1) / (Masters IMP Level_1) = 0.8			
5	Labels	Depend on the knowledge representation used			
	Meta-K	(Do Exercises (Generic)) → (Do Exercises (Primitive)			

3.2 Domain Knowledge Representation in the Hierarchy

As we noticed earlier, the representation of domain knowledge in this hierarchy does not relate to its real contents. Thus, we have used labels to designate the elements in each level and posterior probabilities to express their dependencies. In Table 1, level 4, the representation for concepts linked to the sub-subject of *propositional logic* is shown. But level 5 deals with the exercises to be solved by the learner, thus the representation paradigm to be used depends on the nature of the knowledge involved (procedural, analytical or declarative).

Table 2. Student model on the basis of the domain knowledge hierarchy

Level	Designation	Information		
1	Mastery	(Mathematics, LEVEL)		
2	Mastery	(Logic, LEVEL)		
3	Mastery	*Probability*(Propositional, LEVEL)	*Probability*(Predicates, LEVEL)	
4	Mastery	*Probability*(OR, LEVEL)	*Probability*(NEG, LEVEL)	*Probability*(IMP, LEVEL)
5	Trace	Depend on a predefined format		

3.3 Modeling the Student in the Knowledge Hierarchy

The framework we are proposing here concerns the student's performance in solving problems. As stated earlier, the SM will mimic the hierarchical representation, thus at each level and for each label there will be indicators expressing the student's *LEVEL* of mastery with respect to the corresponding contents (Table 2). Student assessment at the fifth level will directly use data from a trace of the training session. At higher levels, it will be carried on inferences based on performance values from the lower levels. These inference rules indicate the level of mastery to be assigned for a concept, sub-subject or subject, when the levels of mastery in related finer granularities are given. Here, we have used posterior probabilities relations to model those dependencies (as in Table 1).

3.4 Assessing the Student State of Knowledge

Since the student's evaluation is based on inferences, this process should begin at the fifth level. We do not intend to perform a pre-test prior the launching of a training session for the SM initialisation; we will rather collect information related to problem solving in a training session. Thereafter, cognitive and/or behavioural skills to be considered in the evaluation process have to be defined. The skills are sometime slightly linked to the nature of the problem. For example in natural sciences (medicine, biology, etc.), memory and attention are important, while in pure sciences (mathematics, physics), abstraction capability (cognition) and concentration may be needed. Hereon, we shall only consider criteria related to the student's knowledge and his assessment will be simply based on his performance during the exercises solving activities. To this end, a statistical approach, which only takes into account the student's final solution, may be adopted. In this case, we have considered the proportion of solved exercises over the total number of tried exercises. A more refined approach allows the follow up and the assessment of the student on the basis of the number of correct steps performed in the solving process. A last approach will consist in taking into account the coherence of the student reasoning. Here, the question is: *what is the relevance of step (i+1), considering steps* $\{$*i, i-1, i-1 , . . . 1*$\}$? A paradox appears when a student adopts a perfectly coherent, but completely incorrect path right from the first step of the problem solving process.

Information not directly linked to the knowledge content may also help in the student's assessment and may further favour its accuracy. Hence, criteria such as the time spent in solving the associated exercises, the number of steps in solving exercises (compared to the maximum or average number of steps fixed by a domain expert), the numbers of variables used, the time spent in solving similar problems presented successively and the number of tutor's interventions, are parameters which may be involved in assessing the student.

Once these criteria are chosen, one should infer the student's level or class of mastery (from levels 1, 2 and 3 as defined earlier) for the concept related to the exercises considered. For each class, conditions of membership are defined in different ways. They may be deterministically stated by listing some features that the student's performance should have; for example $\{$*(NUMBER of EXERCISES SOLVED = HIGH, TIME FOR SOLVING = MEDIUM)*$\} \rightarrow S \in level\_2$. Alternatively, more flexible conditions define a bound on the probability that a student will master a concept. For example, these conditions may be expressed as *If probability[*$\{$*S masters LOGICAL AND* $\}$*]* > 0.8 *thenS* $\in$ *level_1 performance class*. A clear and precise definition of these conditions will require domain expert advice and empirical data sets of former students. The next challenge is to determine how to obtain these values for a particular student. Neural network computation provides a classification of students when membership conditions are deterministic while statistical techniques provide an estimate of the probability that the learner masters.

Indeed, a multi-layer artificial neural network trained with a back-propagation algorithm classifies a student on the basis of the values of the criteria listed above, for his particular data. In Figure 2, the network will have m inputs where m is the number of criteria chosen among the ones stated above. In this illustration example, m=3 and the inputs to the network are the criteria "*NUMBER OF SOLVED EXERCISES*", "*AVERAGE TIME to solve EXERCISES*" and "*NUMBER OF TUTOR INTERVENTIONS*". The output layer contains three units, each corresponding to a performance level. Note that the network is in fact a 2-stage network since it first assigns the student's parameters to one of the classes *HIGH, LOW* or *MEDIUM*, before the final classification on the basis of the class of those parameters.

Statistical calculations on the other hand evolve with the frequence at which a parameter is observed or not. For example, if the evaluation criterion is the "*NUMBER OF SOLVED EXERCISES*", a learner's level of knowledge with re-

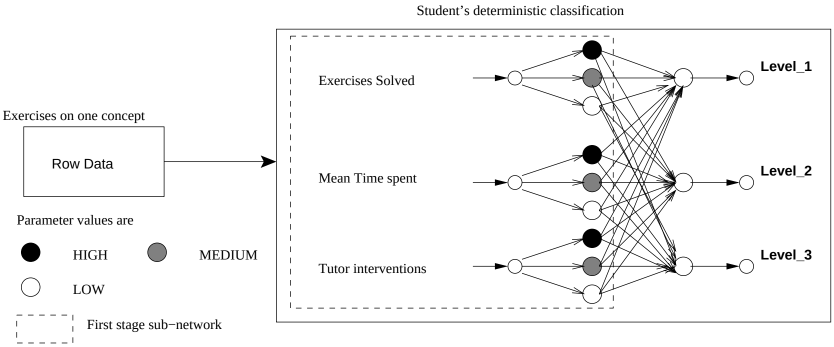

Fig. 2. A neural network architecture for student assignment to a performance class

spect to a concept I may be estimated by taking the frequence p_i of exercises that he successfully solved. Now, the question is to determine how these inferences can be carried on, from the level of exercises to the top level of subjects when the conditions of membership to a performance class are bounds on the probability of mastering a concept, a sub-subject or a subject.

3.5 Updating the Learner Model Using Bayesian Reasoning

Diagnosing the student's level of mastery for a sub-subject is carried out from the aggregation based on his level of mastery for the concepts related to this sub-subject. The same process will happen by aggregating performance values from sub-subjects to subjects and so on. Furthermore, as stated above, inferences may be performed from a knowledge unit to another one in the same level, allowing the tutor or an adaptive agent of the student's model to deduce the student's level concerning a knowledge not yet learned and to eventually optimize the learning process by skipping some basic exercises. The architecture proposed in this paper may be seen as a Bayesian network, since the diagnosis of the learner knowledge at each level in the model is built from *a priori* information related to his state of knowledge at the level below (Figure 3).

Therefore, one should first set posterior probabilities for each knowledge unit in the architecture from the level of the domain to the levels below. This may be achieved by using a neural network classifier similar to the one in Figure 2. In fact, as Ruck et al. [8] showed, a multi-layer perceptron trained with any algorithm minimizing the mean square error between an output unit and the target output approximates the posterior distribution of classes corresponding to that output, *given* the actual network input. In this case, the network output may be interpreted as the probability of mastering the concept, given a *prior knowledge* of former results on subjects (and composite concepts if necessary) related to that concept. To better understand how, lets examine the case where we are looking for an estimate of *Probability[(Mastery of < Basic Concept >) = Level_1/(Mastery of < Composite concept >) = Level_j]* (for example *Prob-*

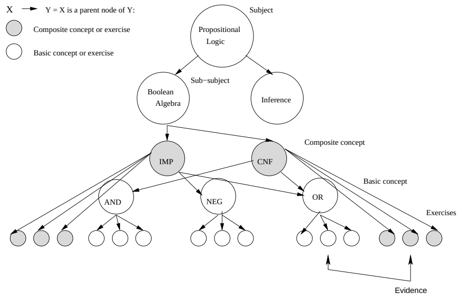

Fig. 3. The student model as a Bayesian network

ability[(Mastery of OR) = Level_1)/(Mastery of IMP) = Level_1)]). We argue that the input patterns to the neural network should be those that were used to evaluate *Probability[(Mastery of < Composite concept >)]*. Those input patterns consist of some features of exercises *directly* linked to the parent composite concept, for example, the number of times that the basic concept intervenes in the solved exercises. Furthermore, since sub-subjects and subjects are not directly linked to exercises in the architecture, posterior probabilities of the form *Probability[(Mastery of <Concept>)=*$Level_i$*/(Mastery of <Sub-subject>)=*$Level_j$*]*, *Probability[(Mastery of <Sub-subject>)=*$Level_i$*/(Mastery of <Subject>) =* $Level_j$*]* cannot be derived from exercises resolution data. We think that those values should rather be established on the basis of an expert judgment for the following reason: from the granularity of concepts and above, the learning matter is more and more dense while our main goal was in fact to avoid defining exercises covering a whole sub-subject or subject.

Once the Bayesian network is set with prior and posterior probabilities, a particular student assessment can take place based on evidence from his exercises solving activities: once an exercise is solved, this evidence is propagated throughout the network in the bottom-up direction. In this architecture, there are restrictions on the way evidence propagates throughout the network. This is due to the fact that two child nodes may influence their (same) parent, without influencing each other: they are not *d-connected*. This is especially true for primitive concepts node, which should not be related, since they refer to a minimal content to be learned.

3.6 Nodes Interactions and Evidence Propagation

Now, our goal is to illustrate two restrictions on the propagations throughout the network. For this demonstration example, the SM probabilities are initialized with arbitrary values. Since inferences will be triggered each time an exercise is solved, "concept nodes" are considered as *observation nodes* via the resolution of one associated exercise. Nodes at the upper levels are considered as *target nodes* since assessment will apply with respect to a sub-subject or a subject.

Exercises linked to a generic concept comprise some elements of basic concepts, but those linked to a basic concept require only knowledge related to it. Therefore, evidence from a generic concept will influence *backward* its ancestors and *forward* all the corresponding children nodes. Nonetheless, evidence from a basic concept will *only* propagate backward to its ancestors. This is intuitive since for example solving a generic exercise is likely to improve the knowledge of *all* the basic concepts which appear in that exercise, but solving an exercise associated to a primitive concept only improve the knowledge of a parent generic concept.

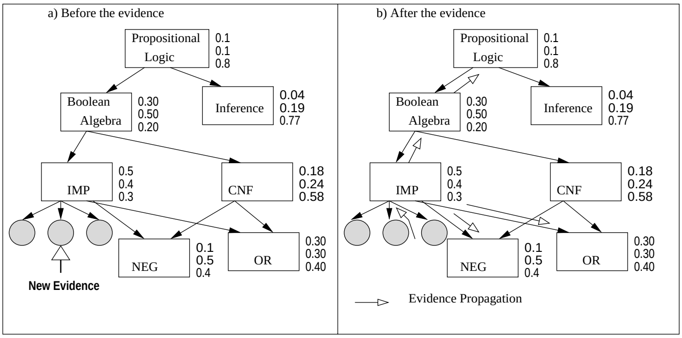

Fig. 4. Network Initial state and evidence propagation at the node *IMPLICATION*

An exercise related to the "*IMPLICATION*" node has been solved in figure 6.a) and arrows in figure 6.b) show the resulting propagation. Beliefs concerning its chidren, "*OR*" and "*NOT*" have changed through the forward propagation of $\pi(OR)$ and $\pi(NOT)$ messages, while the beliefs about its ancestors have changed through the backward propagation of $\lambda(BOOLEANALGEBRA)$ and $\lambda(PROPOSITIONALLOGIC)$ messages. Furthermore, the update in the node "*BOOLEAN ALGEBRA*" does not propagate back to the node "CNF" since any change in the knowledge of "*BOOLEAN ALGEBRA*" due to "*IMPLICATION*" should not influence the knowledge of the CNF concept, unless it is explicitly specified by linking the "*IMPLICATION*" node to the "*CNF*" node. Figure 7.b) shows the propagation resulting after the solving of an exercises related to "*NEG*". Beliefs about "*IMPLICATION*" changed, as well as beliefs about the other ancestors of "*NEG*". But there is no change on the belief about the "*OR*" node albeit the fact that it has a parent node in common with "*NEG*" node. This means that basic concepts with the same parent node should not be *d-connected*. This could be achieved using an *OR-GATE* between basic concepts and their parent node.

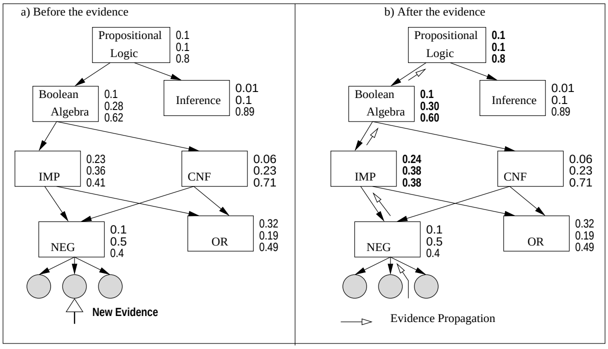

Fig. 5. Network Initial state and evidence propagation at the node "*OR*"

4 Relevance of the Approach

In a learning context, it is useful to partition the domain knowledge in order to ease the evaluation process. The student's assessment can be simplified by designing his knowledge state at various levels of granularity. The hierarchical presentation of learner's knowledge can be justified from the fact that the domain knowledge generally covers several subjects, and learning will involve a part of the whole. Furthermore, for long-term training purposes, it is necessary *to assess the student's state of knowledge with respect to several elements of the domain matter*, in case these elements influence learning content in later training stages. In the example of learning of the sub-subject of *propositional logic*, while examining what happens when the student's evaluation is made in a global way, we notice how difficult it is to take into account the student's preliminary state of knowledge. For instance, if the *conjunction* and *disjunction* concepts have already been learned, there is no mean through which the corresponding exercises can be ignored in the session planning process. Thus, the hierarchical approach to the learner's model simplifies the evaluation process since the learning content at each level of granularity can be considered independently. Furthermore, it makes training more flexible because, if one wishes to learn a subject or a particular concept, the appropriate structure is already available. The students state of knowledge on the whole learning domain can be easily stated in terms of his state of knowledge vis-à-vis the different parts of this domain. Finally, in case of incoherence or misconception about a knowledge item which was assumed to be well mastered, it is easier to track back the evaluation process.

5 Conclusion

The main contribution of this work is the proposition of an approach that takes advantage on the fact that the student's assessment in finer grain knowledge content is more tractable. Thus, the learner's performance is effectively evaluated with respect to more specific knowledge units while his state of knowledge for coarser grained knowledge content is inferred using a Bayesian network. Future works will focus on the validation process where our main concern will be the implementation of the student solving process representation and the estimation of posterior probabilities, using the CyberSciences ITS (Nkambou and Laporte [7]) as our testbed.

References

1. R. Anderson, J. and R. Pelletier. A development system for model-tracing. *Journal of Artificial Intelligence in Education*, 4(4):397–413, 1992.
2. B. Carr and I. Goldstein. *Overlays: a Theory of Modeling for CAI.* AI Memo 406, MIT, 1977.
3. C. Conati, A. Gertner, K. VanLehn, and M. Druzdzel. On-line student modeling for coached problem solving using bayesian networks. In A. Jameson, C. Paris, and C. Tasso, editors, *User Modelling: Proceedings of the Sixth International Conference, UM97*, pages 231–242, Vienna, July 1997. Springer Verlag.
4. P. Fung. Do-it-yourself student modelling. *Computer Education*, 20(1):81–87, 1993.
5. I. McCalla, G. and E. Greer, J. *Granularity-Based Reasoning and Belief Revision in Student Models in Student Models: The Key to Individualized Edu cational Systems, J. Greer and G. McCalla (eds.).* New York: Springer Verlag, 1994.
6. A. Mitrovic, M. Mayo, P. Suraweera, and B. Martin. Constraint-based tutors: a success story. In Laszlo Monostori, József Váncza, and Moonis Ali, editors, *IEA/AIE*, volume 2070 of *Lecture Notes in Computer Science*, pages 931–940. Springer, 2001.
7. R. Nkambou and Y. Laporte. Cooperating agents in a virtual laboratory for supporting learning in engineering and science. In V. N. Alexandrov, Jack Dongarra, Benjoe A. Juliano, René S. Renner, and Chih Jeng Kenneth Tan, editors, *International Conference on Computational Science*, volume 2074 of *Lecture Notes in Computer Science*, pages 366–376. Springer, 2001.
8. W. Ruck, D., K. Rogers, S., M. Kabrisky, E. Oxley, M., and W. Suter, B. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1:296–298, 1990.