A Qualitative Comparison of Techniques for Student Modeling in Intelligent Tutoring Systems

Carolina González¹, Juan C. Burguillo, and Martín Llamas
Departamento de Ingeniería Telemática, Universidad de Vigo, Vigo 36200, Spain
{cgonzals, jrial, mllamas}@det.uvigo.es

Abstract - Intelligent Tutoring Systems (ITS) are interactive learning environments based on instruction assisted by computers. The intelligence of these systems is largely attributed to their ability to adapt to a specific student during the teaching process. In general, the adaptation process can be described by three phases: (i) getting the information about the student, (ii) processing the information to initialize and update a student model, and (iii) using the student model to provide the adaptation. In this paper we studied aspects related with student modeling (SM) in Intelligent Tutoring Systems. First we make a qualitative comparison of two techniques: Bayesian Networks (BN) and Case-based Reasoning (CBR) for SM. We apply both techniques to a case study concerning the development of an ITS for e-learning in the medical domain. Finally, we discuss the results obtained.

Index Terms – Bayesian Networks, Case-based Reasoning, Intelligent Tutoring Systems, Student Model.

INTRODUCTION

Adaptive curriculum sequencing and interactive problem solving support are important aspects of intelligent tutoring systems (ITS). Student modeling (SM) is mainly used in order to adapt the ITS to each student. SM involves creating an individual model for each student that identifies the current knowledge of the student and adapts the learning sequence accordingly to help him/her in navigating through the course. In general, the adaptation process can be described by three stages: getting the information about the student, processing the information to initialize and update a student model, and using the student model to provide the adaptation.

Some student models are built for recognizing student plans or solution paths [1], some are built for evaluating student performance or problem solving skills [2], and some are created for describing constraints that the student has violated [1] [2]. But there is one question that must be answered to build a new student model: what aspects of the student should we model in a specific intelligent tutoring system?

In this paper we address what to model and how to divide the student model into components in a context of a Intelligent Tutoring System for training in infectious diseases, ITS-TB. Two modeling techniques are used: Case-Based Reasoning and Bayesian Networks. We analyze the student modeling process with these techniques and we make a qualitative comparison of BN and CBR.

BAYESIAN NETWORKS

Numerical techniques for reasoning under uncertainty have been applied to student modeling, especially in the last few years. Quantitative approaches can be used in conjunction with qualitative techniques in order to handle uncertainty. Numerical techniques, such as Bayesian Networks (BN) [4], the Dempster-Shafer theory of evidence [19], and fuzzy logic [20] are used for generating student models. Other techniques, although computationally cheap (e.g., the model tracing approach [21]) can only record what a student knows and not the student’s behavior and characteristics. BN is the most broadly used approach for reasoning under uncertainty in learning environments. Probabilistic or causal relationships among variables are represented as a directed acyclic graph. Using prior and conditional probabilities attached to each node, it is possible to propagate changes in probability values on receipt of evidence. The casual information encoded in BN facilitates the analysis of action sequences, observations, consequences, and expected utility [5].

Several authors in different areas have explored the use of Bayesian Networks to represent student models. Mayo [6] classifies Bayesian student modeling approaches based on how the structure of the network and prior, conditional probabilities are elicited. Mayo identifies three types of Bayesian student models: expert-centric, efficiency-centric, and data-centric. Expert-centric models use experts to specify the structure of the network and its corresponding initial prior and conditional probabilities. Resulting networks usually contain a big number of variables, making difficult to evaluate the model. Efficiency-centric models restrict the structure of the network in order to maximize efficiency. There are several risks with this approach, such as: oversimplifying the model and/or introducing incorrect assumptions. Finally, data-centric models use data from previous experiment and/or pre-tests to generate the network and its probabilities. Although the resulting network and probabilities can be used efficiently in implementing ITS, it does not guarantee that the final structure can be easily reviewed and understood by a human.

¹ Carolina González, Departamento de Sistemas, Universidad del Cauca, cgonzals@unicauca.edu.co

© 2006 IEEE

October 28 – 31, 2006, San Diego, CA

36th ASEE/IEEE Frontiers in Education Conference

T1F-13
Some systems using Bayesian Networks in student modeling are: OLAE, POLA, ANDES and others [7]. These systems store information about students in different knowledge areas. The purpose of the OLAE system [8] is the student model assessment. It uses Bayesian Networks to observe student behavior and compute the probabilities that the student knows and uses each of the rules in a given knowledge domain. Additionally, it acts when the student has solved the problem. POLA [1] extended the OLAE system. It determines the rules known and the road followed by the student in the problem solving. Finally, ANDES [9] determine the student prior probabilities of knowing a set of knowledge elemental items.

CASE BASED REASONING

Case-based reasoning (CBR) is a relatively new problem-solving strategy and machine learning technique. CBR is an effective paradigm for problem solving in many aspects. Instead of relying solely on the general knowledge of a problem domain or making associations about the generalized relationships between problem descriptors and conclusions, it is able to utilize the specific knowledge of previously experienced, concrete problem situations or cases [3].

The CBR paradigm covers a range of different methods for organizing, retrieving, utilizing and indexing the knowledge retained in past cases. Cases may be kept as concrete experiences, or a set of similar cases may form a generalized case. Cases may be stored as separate knowledge units or divided into subunits and distributed within the knowledge structure. Cases may be indexed by a prefixed or open vocabulary, and within a flat or hierarchical index structure. The solution from a previous case may be directly applied to the present problem, or modified according to differences between the two cases. The matching of cases, adaptation of solutions, and learning from an experience may be guided and supported by a deep model of general domain knowledge, by more shallow and compiled knowledge, or only be based on an apparent, syntactic similarity. CBR methods may be purely self-contained and automatic, or they may interact heavily with the user to support and guide his choices. Some CBR methods assume a rather large amount of widely distributed cases in its case base, while others are based on a more limited set of typical ones. Past cases may be retrieved and evaluated sequentially or in parallel.

Student modeling using CBR technique is established as an important paradigm in developing Intelligent Tutoring Systems. The idea is based on the notion that student's problem solving capabilities can be evaluated by looking at how the student accesses past solved problems (cases) which are similar to their current situation. This technique is simple and does not require a complex inference algorithm. Furthermore, it can be applied to obtain various types of information related to the knowledge of the learner, such as the knowledge level, the capabilities (analogy and adaptation), and the solution known by the learner. The Case-Based Reasoning approach to educational application has already been explored. CBR is often used in teaching, planning, design and argumentation. Recently, there has been research focused on Case-based intelligent tutoring systems [23]. In such systems a set of solved problems (cases) is presented to the student who is expected to learn through these cases to solve a new problem. In [17] CBR is used to construct exclusively the knowledge component of the student model. Other approaches do not concern itself with the specific training, but solves problems found in a generic teaching process.

ITS-TB: A CASE STUDY

We are building an intelligent tutoring system, ITS-TB, which helps medical students in the problem solving and decisions making process. The structure of the course is presented in Table 1.

Each student has different knowledge, abilities, preferences and academic background. The student model in ITS-TB can be used for different purposes:

- To determine if the student is ready to continue with the next curriculum topic, and to choose this topic.
- To generate explanations according to the student knowledge.
- To advise and help the student. The tutor does not interrupt the students frequently, and allows them to learn from their mistakes.
- To generate problems according to the student knowledge level. Once the weak points of the student are identified, the system generates a problem. This problem is solved simultaneously by the expert module in order to diagnose the student solution. In this sense, each student that interacts with the system will solve a set of problems adapted to his knowledge level.
- To generate the adequate teaching strategy according to the student knowledge.

### 1. Student Model Content

A comprehensive student model should contain information about the previous students’ knowledge, the student’s progress, preferences, interests, goals, personal information and any other information related to the student. Based on the dependence upon the subject domain, the content held in student models consists of two parts:

<table>
<thead>
<tr>
<th>Assignature</th>
<th>Signature Duration (Months)</th>
<th>Topic</th>
<th>Weight</th>
<th>Time (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PYC</td>
<td>5.</td>
<td>Infectious Disease</td>
<td>$\lambda = 0.4$</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Non infectious disease</td>
<td>$\lambda = 0.6$</td>
<td>4.5</td>
</tr>
</tbody>
</table>

**TABLE 1**

**HEALTH CARE COURSE**
Domain specific information (DSI): it is also denoted as student knowledge model (SKM) which represents a reflection of the student's state and level of knowledge on a particular subject domain.

Domain independent information (DII): it is slightly different from system to system. The domain-independent information about a student may include learning goals, cognitive aptitudes, measures for motivation state, preferences about the presentation method, factual and historic data, etc. We propose a student model that includes individual and cognitive characteristics grouped in a component named knowledge component. This component contains information related to the (1) knowledge level of the student, (2) personal information, (3) learning preferences, and (4) psychological characteristics.

II. Student Modeling with Bayesian Networks

In order to construct the Bayesian Network for modeling the student, we use GENIE [11], which is a development environment for building graphical decision theoretic models. This tool implements several BN inference algorithms, including clustering and stochastic algorithms.

In the student model, the cognitive state of the student is inferred from two parts: the previous data about the student and the student's behavior during the interaction with the system [12], both involving uncertainty. The diagnosis process is made by means of propagating the probabilities contained in the nodes of the network. In ITS-TB the knowledge is divided into a hierarchic structure integrated by a set of nodes: concepts, topics and tutorial. The concepts nodes represent the knowledge level that the student has acquired in every concept of the tutorial. The teach-concept nodes indicate the grade of viability for studying a concept in the tutorial. The topic nodes store the knowledge score acquired by the student in a specific tutorial topic. Finally, the tutorial node is used for storing the knowledge score acquired by the student in a specific tutorial topic. Figure 1 shows the Bayesian network used.

III. Student Modeling with Case-Based Reasoning

Our approach for student modeling with CBR includes a representation of the knowledge and reasoning of the student, and the way how the student acquires new knowledge in order to perform intelligent learning. The student model built with CBR is structured as a multi-agent system [13] and follows the steps of a CBR cycle.

Retrieval Phase: The information is acquired when the student interacts with the system for the first time. The system models the student as a new case and searches for a similarly solved case by comparing this new case with the existing case base. The system uses a table of similarity to select a similar case optimally. Generally the algorithm for computing similarity uses a simple Nearest Neighbor Algorithm [14]. The algorithm formula is:

\[ \text{Similarity} (N, S) = \sum_{i=1}^{n} f(N_i, S_i) \times w_i \]  

where:

- \( N \) is the new case (new student)
- \( S \) is the source case (past student history)
- \( n \) is the number of features in each case
- \( i \) is an individual feature from 1 to \( n \)
- \( f \) is a similar function for feature \( i \) in cases \( N \) and \( S \)
- \( w \) is the weight of feature \( i \)

The Procedure Matching is implemented to check the corresponding features in the cases stored. A tutor agent performs this task. This procedure is presented in figure 3.
Procedure Matching (features)
begin
  reviseFeatureCaseBase ();
  computeDegreeSimilarity ();
  computeDegreeMatch ();
  rankingCases ();
End

FIGURE 3. PROCEDURE MATCHING

- **Reuse Phase**: If a similar case is selected, then the system estimates the degree of similarity of the searched case by using fixed values as thresholds. If the degree of similarity is higher than the fixed threshold value, then the diagnosing and solving method, of the similar case found in the case base, is applied to the new one. Otherwise, the system regards the new case as nonexistent in the case base and it considers the new case as completely new. An adaptation agent performs this task. In the reuse phase five groups of rules are proposed for: (1) analyzing the student’s misconceptions, (2) diagnosing the student’s knowledge, (3) creating an optimal problem case, and (4) inference over an incomplete case.

- **Revise Phase**: The selected case is revised by an adaptation rule to remove unsuitable elements and to correct inconsistent attributes. The revise phase may change attributes or values within the case. This phase is controlled by an orientation agent. This agent performs two tasks: (1) evaluates the case solution generated by reuse, and (2) repairs the case solution using domain-specific knowledge.

- **Retain Phase**: If the degree of similarity is lower than the fixed threshold value the system regards the new case as nonexistent in the case base and proceeds to retain its details. The system in this phase checks the results of the problem solving procedure. If the results are correct, the system inserts them into the case base; otherwise the system assumes that the student does not perform well and infers the student’s misconceptions. Lastly, the system provides the student an optimal problem solving solution.

Implementing the CBR Student Model

In order to implement the student model with CBR we have used CBR Works [15]. The agent paradigm [13] has been chosen because of its autonomy and proactivity in order to take on the human role of a mentor. As we stated before, information about the students is regarded as cases. When the student starts learning, the information about him is extracted from the student model and is converted into a new case. Figure 4 describes the student modeling with agents. The cases are stored in a case base and the model is described in terms of concepts, attributes, and types. Figure 5 shows the concept manager. The cases stored in the case base can be in four states:

- **Unconfirmed**: Case is incomplete or not yet validated. Cases of this mode will not be retrieved.
- **Confirmed**: Case is complete and validated. Cases of this mode are allowed for retrieval.
- **Protected**: Case is complete and validated, but protected against unwanted changes.
- **Obsolete**: Case contains old data, may be interesting for statistics. Cases in this mode are not retrieved.

FIGURE 4. AGENTS IN CBR STUDENT MODELING

FIGURE 5. CONCEPT MANAGER

Figure 6 presents the generation of a new case and its addition in the case buffer. All the cases have been set to be in confirmed and protected mode, in order to be retrieved and to avoid modifying their contents.
In the retrieval process, the case navigator creates new queries where new values are assigned to specific concepts to create a new case. Moreover, in this process, we can use an existing case as a query. The similarity of the retrieved cases is obtained by using attributed based similarity. Figure 7 shows a similar case.

After modeling with CBR, the SM of ITS-TB we realize that:

- The evaluation of the student performance helped to decide when to give hints or answers if the student cannot answer a question.
- The student reply history allows the tutor to end a dialogue and return to the original plan when the student could not continue along a causal link.
- The category student answer, a part of the student reply history, is effective in helping to decide on different retry strategies. It recognizes near misses and other categories of answers that could be previously treated as totally incorrect answers.
- The tutoring history prevented the tutor from giving the same hint repeatedly.

**DISCUSSION**

To determine what information must be included in the student model, a system-dependent task, we have to consider several constraints. The first one is the nature of the domain: Is it a quantitative or a qualitative domain? The second constraint is the structure of a tutorial session. Basically, we need to consider the student-system interaction; how the system presents the problem to the student, and how the system can observe about the student’s evolution. A third constraint is the tutorial decisions that the system needs to make. Does the system need to plan the curriculum, to switch between tutoring protocols, to plan tutoring dialogue, or just give a simple feedback without multi-step plans? Furthermore, the technique or techniques used for approximated reasoning must also be considered. We consider CBR as an adequate approach once it has been compared with BN because of the results showed in Table 2 explained next.

- **Theoretical Basis:** BNs have been very used in student modeling by their great versatility and theoretical solidity. In addition, BN are considered as a powerful tool to make additive and predictive inferences. On the other hand, CBR is an effective paradigm for problem solving using incremental learning since, each time a problem is solved it is stored as a case in the CBR memory for further use.
- **Structure Definition:** This aspect is one of the most difficult to solve. When using BN it is assumed that the structure of dependencies and the parameters are provided by the human expert. Nevertheless, it can be impossible for a teacher to specify the great number of conditional probabilities required. This has motivated the research in techniques for parameter simplification. It is easier for the expert to describe his opinions in terms of data sets stored in case bases or rules than he quantifies them in the form of probabilities.
- **Model Initialization:** In BN and CBR the initial data of the students can be obtained by means of previous questionnaires or tests. Some approaches rely on a mixed mechanism where BN and CBR are combined to acquire initial knowledge on the students. In this approach the information necessary to construct the BN is taken from cases stored in the case bases.
- **Diagnosis Process:** The probability propagation algorithms use the implicit relations of independence in the structure of a BN to calculate the probabilities of every node given the evidence available. After calculating these probabilities, it is possible to make predictive and additive inferences. These calculations can be carried out applying the Bayes’ Theorem, the law of total probability and the conditions of conditional independence. In this sense, the number of operations grows exponentially with the number of variables of the network becoming a computational intractable task. Using CBR the diagnostic task can be made in a simpler way, following the process of case recovering with similarity algorithms of low complexity. Besides, the inference process can be comparable with the one used by a human expert.
- **Use:** CBR allows to select the adequate teaching strategy and, at the same time, to diagnose the misconceptions committed by the students. An ITS based on CBR improves the precision at the time of determining the knowledge level and the errors of the students during the learning process.
- **Complexity:** CBR allows easy maintenance of the case base: modify an operation, changing the type of an attribute, adding a new case, and easily updating the model. Some problems found with CBR were solved using the agent’s technology (p.e., to manage the big size of case library in complex domains). The difficulty to
obtain the initial parameters or add a new node in BN is bigger due to: the complex knowledge acquisition process, the conditional and unconditional dependencies, the update of the model, and the computational complexity of the inference algorithms.

## TABLE 2.

<table>
<thead>
<tr>
<th>Theoretical Basis</th>
<th>Bayesian Networks</th>
<th>Case-based Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure Definition</td>
<td>Variables, relations and probabilities</td>
<td>Cases</td>
</tr>
<tr>
<td>Model Initialization</td>
<td>Not specific</td>
<td>Tests, stereotypes, mixed</td>
</tr>
<tr>
<td>Diagnosis Process</td>
<td>Propagation algorithms</td>
<td>CBR stages</td>
</tr>
<tr>
<td>Use</td>
<td>Selection of the teaching strategy</td>
<td>Teaching strategy, student misconceptions</td>
</tr>
<tr>
<td>Complexity</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

## CONCLUSIONS

In this paper we have discussed how to build a student model considering the nature of the domain, the structure of the tutoring session, and the tutorial decisions that an ITS needs to make. We compared qualitatively the student modeling with two techniques: Bayesian Networks and Case Base Reasoning.

Bayesian Networks is an approach broadly used in student modeling, but there is a high grade of complexity in their design. The main reasons for avoid the use of BN are the computational complexity of the algorithms and the difficulty of the knowledge acquisition process.

Case-based Reasoning technique as an easier approach for constructing a student model because it presents numerous advantages: (a) it is easier to handle, to update and to maintain the student model, beneficial for both the tutor and the student; (b) it promotes student reflection because they report student's misconceptions and the reasons why they have happened; and (c) it facilitates the supervision of the students by enabling the tutor to have a solid and continuous view of the student performance, including both qualitative and quantitative information.

## ACKNOWLEDGMENT

The authors want to thank anonymous reviewers and Prof. Ubaldo Garcia-Palomares for their helpful comments.

## REFERENCES


