We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,200
Open access books available

116,000
International authors and editors

125M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Learner Modeling Based on Bayesian Networks

Anouar Tadlaoui Mouenis, Aammou Souhaib and
Khaldi Mohamed

Additional information is available at the end of the chapter

Abstract

The work presented in this chapter lies within Learner modeling in an adaptive educational system construed as a computational modeling of the learner. All actions of the learner in a learning situation on an adaptive hypermedia systems are not limited to valid or invalid actions (true and false), but they are a set of actions that characterize the learning path of his formation. Thus, we cannot represent the information from the system of each learner using relative data. It requires putting our work in a probabilistic context due to the changes in the learner model information during formation. We propose in this work to use Bayesian networks as a probabilistic framework to resolve the issue of dynamic management and update of the learner model. The experiments and results presented in this work are arguments in favor of our hypothesis, and can also promote reusing the modeling obtained through different systems and similar modeling situations.

Keywords: Learner model, adaptive Hypermedia educational systems, Bayesian networks, Cognitive diagnosis, Uncertainty

1. Introduction

First of all, to clarify our purpose, it seems important to note that the work presented in this chapter lies within learner modeling in an adaptive educational system, construed as a computational modeling of the learner; that is to say, the representation and specification of the learner's knowledge. Different approaches have been taken to manage modeling of the

learner with multiple objectives, from the evaluation of the learner's knowledge to the recognition of the plan followed in problem solving.

Despite these various attempts at modeling learning characterized by a dynamic aspect, we always find that there are difficulties in achieving this goal. The proposed approaches provide us with only a static view of the learner model, yet this model is always in development (the learner's knowledge is evolving within the same module). Therefore, a dynamic view is essential. In order to monitor the behavior of the learner in real time and during formation, we must adopt a dynamic modeling approach when managing learner modeling.

The actions of the learner in a learning situation are not limited to valid or invalid actions (true and false), yet it is the actions that characterize the formation of the learning path. From this observation, we cannot represent information from the system of each learner using relative data. Rather, we must place our work in a probabilistic context due to changes in the learner model during formation.

The problems presented in this chapter can be summarized as follows: How should we represent the different functions of a learner model? And what approaches can be used to perform updates on the different characteristics of such a model?

In this work, we propose the use of Bayesian networks as a probabilistic formalism to resolve the issue of management and dynamic update of the learner model. To resolve this issue, we must first ask: Why and how can we represent a learner model with Bayesian networks? How can we go from a dynamic representation of the Unified modeling language diagram of the model to a probabilistic representation with Bayesian networks? Is this consideration experimentally justified?

2. Theoretical approaches

The purpose of this section is to provide the readers with knowledge required in the field of learner modeling. In this section, we address the definitions and terminologies of the chapter's key words.

2.1. Model of the learner

2.1.1. Definition

Learner modeling is the modeling of all the important features that affect the learner (knowledge, preferences, goals, etc.). It identifies relevant information, and structures, initializes, updates and exploits it. By replacing the word "learning" with the term "user", this definition is also applicable to the model of the user. An application other than the learner's educational model is called the user model.

The main goal of a learner model is to store learner information, such as the learner's level of knowledge or skill pertaining to a given topic, and his or her personal information, such as psychological characteristics and preferences.

Zaitseva [1] defines the learner model as a set of structured information about the learning process, in which the characteristics of the learner are considered to be the values of this structure. According to Beck [2], the learner model acts as the key to system adaptation by providing the necessary data to other modules.

The uncertainty of the information contained within the learner model and the intention behind its creation have been the focus of many studies. Thus, a learner model represents system beliefs about learners' beliefs, accumulated during the diagnostic process.

The learner model can be an integral part of adaptive hypermedia systems, as it can be shared with multiple systems. In this last case, we discuss user modeling servers [3]. This type of server is used in environments where more distributed adaptive systems access the server to query or update user information. CUMULATE is one of the most known and used systems for user modeling servers.

2.1.2. Foundations of the learner model

Self [4] defined a formalization of the learner model that is based on the beliefs and knowledge of the system and the learner. Beliefs are represented by formulas in propositional calculus. Objects of belief are called propositions. Beliefs are related to the behavior of an agent (A), a user (U) or a system (S). BA = {p/BAP} is the set of beliefs of agent A. BAP are the proposals themselves.

BSU = {p / BSBUp} is the set of proposals that system S believes are believed by user U (see Fig. 1).
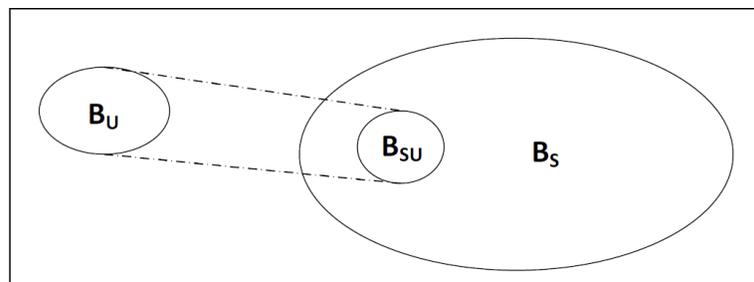
Figure 1. The representation of the system and user beliefs

The learner model can be defined by a set of proposals that the system S thinks about learner U: UM = BS (U) = {p / BSp (U)}

Belief can be replaced by knowledge; therefore, KAP = BAp.

To distinguish between the different aspects of the learner model, Self distinguishes the following proposals:

- Proposals that are dependent on a field that a learner acquires in a system.

- Independent clauses of the system domain. These clauses are also called the background.

This proposal describes the cognitive and personal characteristics of the learner, also known as behavioral skills, which include preferences, tasks, goals and experience.

2.2. Bayesian networks

Before describing our investigation of the use of Bayesian networks in learner modeling, we'll define such networks and address the meaning of inference in this context.

In the rest of this section, we'll take a typology of nodes inspired by Conati [5], and found in different terms in the literature. The field layer is the set of nodes modeling epistemic knowledge of the learner, and the task layer is the set of nodes modeling the actions of the learner.

2.2.1. Definition

Numerous models have been created through the representation of knowledge. Probabilistic graphical models, and especially Bayesian networks initiated by Pearl [6] in the 1980s, have proven to be useful tools for representing uncertain knowledge and reasoning from incomplete information.

A Bayesian network is a directed acyclic graph in which the nodes correspond to the variables (user properties), and the links represent probabilistic relationships of influence. These variables can belong to the field of knowledge, the base knowledge and / or the cognitive model. Each node represents the system's belief about possible values (levels, states) of the variable. Thus, the conditional probability distribution must be specified for each node. If the variables are discrete, they can be presented as a table.

The graph is also called the "structure" of the model, and the probability tables are its "parameters". They can be provided by experts, or calculated from data; generally speaking, the structure is defined by experts and the calculated parameters are from experimental data.

Consider a Bayesian network $B = (G, N)$ defined by
$G = (X, E)$, an acyclic directed graph with various vertices associated with a set of random variables $X = (X, ..., Xn)$; $N = \{P(Xi \mid Pa(Xi))\}$ All the probabilities of each node Xi are conditional to the state of its parents $Pa(Xi)$ in G.

According to Mayo [7], a Bayesian network allows compact representation of the joint probability distribution over a set of variables:

$$P(X1, X2, \cdots, Xn) = \prod_{i=1}^{n} P(Xi \mid Pa(Xi))$$

These methods obviously use the concept of conditional probability, i.e., what is the probability of Xi knowing that I have observed Xj ; but they also use the Bayes theorem, which calculates, conversely, the probability of Xj knowing Xi, when $P(Xi \mid Xj)$ is known.

2.2.2. Bayesian network construction

To specify a Bayesian network in a comprehensive way, it is necessary, as we have seen in the definition, to specify the network structure (the acyclic graph) and the network parameters

(the probability tables). To reach this specification, there are two approaches: 1) the collection of expertise, and 2) the machine learning, which is one of the attractions of Bayesian networks. A combination of these two approaches is also possible.

In the first approach, the collection of expertise, we must begin by defining the network structure, starting with identifying the possible nodes, and then we distinguish between hypothetical (unobservable) variables and informational (observable) variables. The next step concerns the analysis of the existing arc in terms of the influence of one variable upon another. Traditionally, if an arc is directed from A to B, A is a cause of B; however, in the case of learner modeling, we will see that the interpretation is not so simple. The parameters are in turn attached to approximations using qualitative or frequentists' information.

A Bayesian network is considered as a probability distribution. By using maximum likelihood as a statistical learning parameter criterion, the result is a Bayesian network with a fixed structure and with E as a comprehensive basis of example. If the parameters of the Bayesian network are equal to the frequencies of the same features observed in E, the maximum likelihood will be achieved. A test is necessary to determine the conditional independence of random variables in the statistical learning structure.

3. Learner modeling

In this section, we present the steps to follow when modeling the learner in an adaptive educational system, beginning with the user meta-model and then moving to use of the case diagram, and regrouping all actions of the learner in an adaptive system.

3.1. The metamodel

Here we discuss a specific user meta-model for e-learning, as presented by Aammou [8]. This model features a combination of models for e-learning and adaptive hypermedia. It takes into account elements, such as the history of actions that are missing in formal models. The construction of this model allows us to understand the user's creative process model for adaptive hypermedia, helping us to build our hypernym model.

In our user model for e-learning, we want to be able to:

• Define the characteristics attributes, essential and common to all users (name, username, password and age).

• Define attribute categories to separate the user's preferences, school / career and other attributes. This distinction will facilitate importing data, system maintenance, as well as communication with external systems; the attributes are differentiated according to their nature.

• Retain documents that have been covered by the user in either of these two ways: 1) by inclusion of the documents in a whole, greater course. Or by having documents specifically related to the concept that the user has investigated. The aim of this historical duplication

is such that when the user wishes to come back to a concept already brought to his attention, he is presented with documents that are the same as those from his first learning of the concept.

The UML class diagram representation of our user model is given in Fig. 2.

Figure 2. UML class diagram representing the user metamodel

- The **User Manager** class is responsible for interfacing with the other components of adaptive hypermedia systems. For this purpose, the Ask and Tell methods are used to ask questions and provide answers to the external components (domain model, adaptation model). The User Manager class is connected to all users, and is responsible for managing by an aggregation relationship.

- The **User** class is responsible for representing information pertaining to a particular user. It is composed of predefined attributes: name, username, password and age.

- The **Attribute Preference** class is responsible for representing the preferences of the user. These are view preferences: font size, color problems, contrasts, etc., as well as presentation preferences. The user may prefer textual or graphic elements, and may not want an audio element, for example.

- The **Attribute Background** class is responsible for representing the user attributes related to academic / professional background.

- The **Stereotype** class is responsible for representing the various categories of stereotypes to which the user belongs. By definition, a stereotype is an image or fixed design and schematic of an aspect of reality. In our model, a stereotype consists of a name and a value. The name sets the stereotype (e.g., "learning rate"), and the value is used to characterize the user (e.g., "quick" for the stereotype "learning rate"). The number of possible values is often reduced to a given stereotype. Values are often based on other attributes. Stereotypes differ from other attributes in their schematic characterization of the user, as they can represent much more granular elements.

- The **Other Attribute** class is responsible for representing user attributes that are not related to the user's career and are not preferences, e.g., a data encryption key. The purpose of this class is to ensure compatibility of the model with standard models like IMS or PAPI Learner, because some attributes do not fit into the other categories of attributes defined above.

- **Degree** is an association class that is responsible for giving a value to the knowledge of a concept by the user. The possible values are: very low, low, average, good, and excellent. This scale is a range of values that allows good precision with respect to a binary classification, and avoids a degree of accuracy that is too high, and therefore it is very useful for adaptation.

- The **Historical** class is in charge of representing a historical document covered in the learner path, allowing one to give the date of the course of a document, and the browsing history in the order of a historical path [it contains two methods, NextDocument() and Previous-Document()]. The historical class can be used to represent all the documents covered by a user's history, or to represent the historical documents covered to reach a certain degree of knowledge for a given concept.

The classes of Document and Concept are detailed in the model domain.

3.2. The use case diagram

Based on the meta-model, we were able to map out the functionality of the learner using the use case diagram (Fig. 3) to reflect a portion of the student's actions in an adaptive system. In this section, we will explain each of these actions, and consider the relationships of these actions with each other and within the system operation process.

Based upon the meta-model presented in the previous section, we have illustrated a learner's actions in a learning situation in an adaptive educational system (Table 1).

Learner's actions
- Follow courses
- Take pretest
- Take evaluation

Table 1. The mains actions of a learner in a learning situation in an adaptive system

In Fig. 3, a main actor is identified, named "the learner". The figure shows the generalization relationships between use cases and the learner, and the generalization relationships of inclusion and extension between use cases.
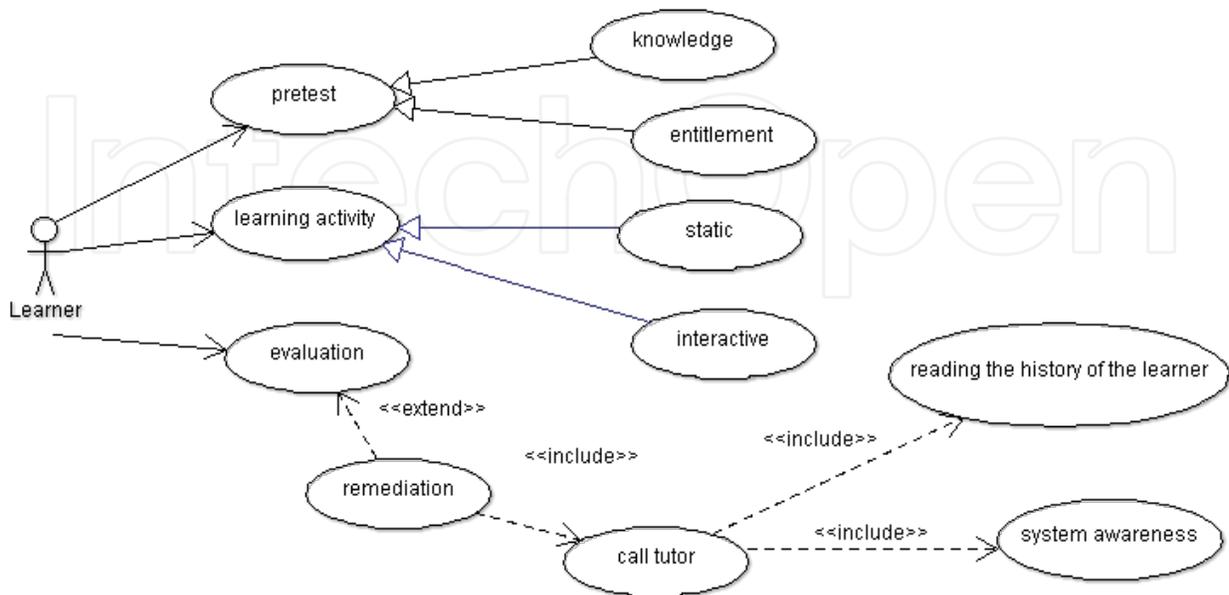

Figure 3. Use case diagram UML representing the learner actions

In particular, the functional requirement of "Learner" represents all information about the learner in the hypermedia system (the learner's knowledge, skills, personal information, etc.). This functional requirement is shown with a generalization relationship with three functional requirements:

- "**Pretest**" – this represents information about the pretest the learner has to take before entering the learning situation. The pretest is composed of two types of evaluation components: 1) tests of knowledge depicted with the functional requirement "*knowledge*", and 2) the functional requirement "*skills*", which represents the test through which we will evaluate the learner's skills.

- "**Learning Activity**" – this functional requirement represents information about the learning activities. Each learning activity in an adaptive educational hypermedia system is of two types: 1) static activities represented by the functional requirement "*Static*", and 2) interactive activities represented by the functional requirement "*Interactive*".

- "**Evaluation**" – this represents the information about the evaluation tests the learner has to take after completion of each learning activity. If the learner fails in the evaluation, the learner must pass to remediation; which is represented by the functional requirement "*Remediation*", which is connected to the functional requirement "*Evaluation*" through an extension of relationship.

In the case of remediation, the functional requirement "Remediation" involves activation of the functional requirement "Call Tutor" through an inclusion relation. This requirement

represents activation of the tutor to help the student to return to shortcomings in the learning activity.

Another inclusion relation is represented in Fig. 3. The actions of the learner in an adaptive system are represented, appearing in the relationship between the functional requirement "Call Tutor" and the requirement of "Reading the History of the Learner", which activates the return of the system to the profile and the learner's course information. The requirement "System Awareness" enables the system to follow the course of the learner after remediation.

4. Bayesian network development

In this section, we present the transformation of our use case diagram representing the learner model, as presented in [9], into a Bayesian network.

4.1. The development rules of a Bayesian network

4.1.1. The generalization relationship transformation

A generalized type of use case represents a functionality that allows all instances of specialized use cases. The transformation of this type of relationship to nodes of a Bayesian network is considered simple.

In Fig. 4, use case A is a generalization of use cases A1 and A2, and we represent the functional requirements of A1 and A2 as being descendant of the functional requirement A.
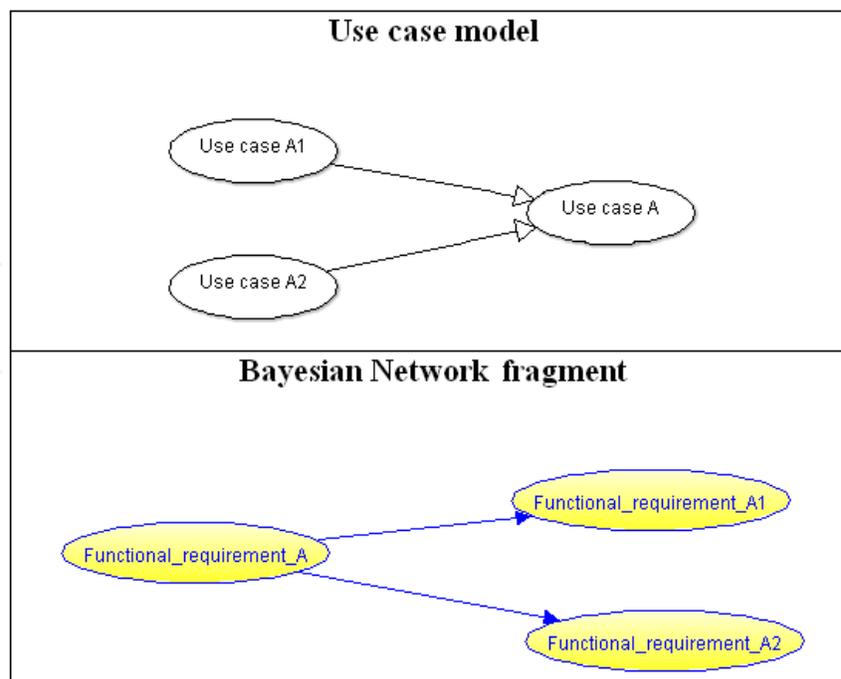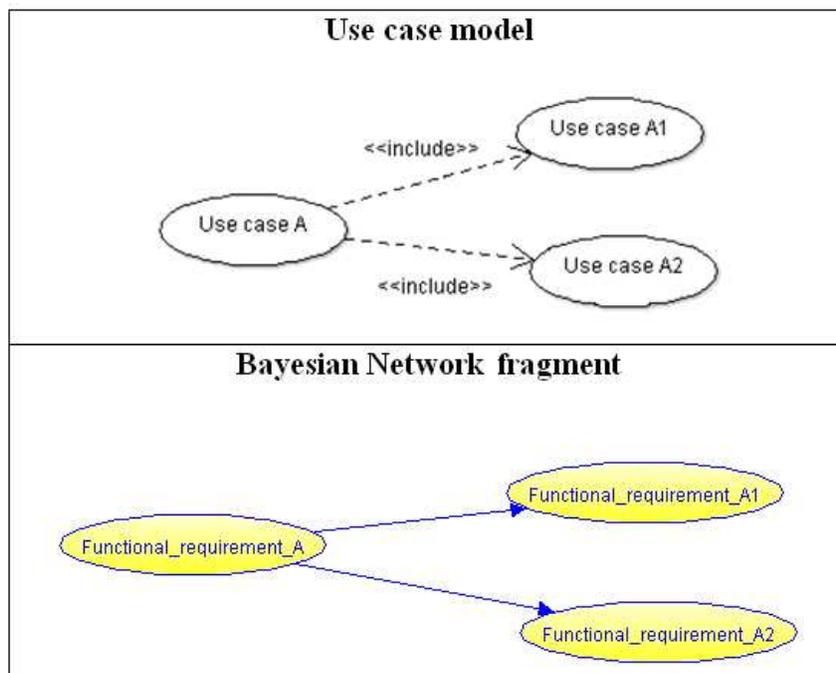
Figure 4. The generalization relationship transformed into a Bayesian network fragment

This results in a Bayesian network with a similar structure. The direction of the arc flows from A to A1 and A2 reflects top-down decomposition. This indicates that one is more likely to encounter a general case with specific functional requirements, including those in the Bayesian network, having developed the information that is represented by the arrows of the use case. Therefore,

$$P(A) = prior$$
$$P(A1 \mid A) = P(A \mid A1)P(A1) / P(A)$$
$$P(A2 \mid A) = P(A \mid A21)P(A2) / P(A)$$

4.1.2. The inclusion relationship transformation

The inclusion relation in a use case diagram represents the situation in which a use case is composed of a number of various use cases. For inclusion, a high level of use cases cannot be executed without the implementation of sub use cases.

Figure 5 represents this relationship, with case A including cases A1 and A2 if the behavior described by case A includes descendant behavior; that is, if A depends on A1. When A is pressed, the east must, as part of A.

Figure 5. The inclusion relationship transformed into a Bayesian network fragment

This results in a Bayesian network with a similar structure. The direction of the arc flowing from A to A1 and A2 represents a bottom-up composition. This indicates that one is more likely to encounter a general case with specific functional requirements, and the functional require-

ments, including those in the Bayesian network, have developed the information that is represented by the arrows of the use case. Therefore,

$$P(A) = prior$$
$$P(A1 \mid A) = P(A \mid A1)P(A1) \; / \; P(A)$$
$$P(A2 \mid A) = P(A \mid A21)P(A2) \; / \; P(A)$$

4.1.3. The extension relationship transformation

The extension relationship is probably the most useful because it has semantic meaning; it represents a particular use case branched additional behavior, given the satisfaction of certain conditions.

Figure 6 represents use case A, which extends to use case A1, when use case A can be called during execution in the case of A1 use. Run A1 can possibly lead to the execution of A; unlike the inclusion, the extension is optional.
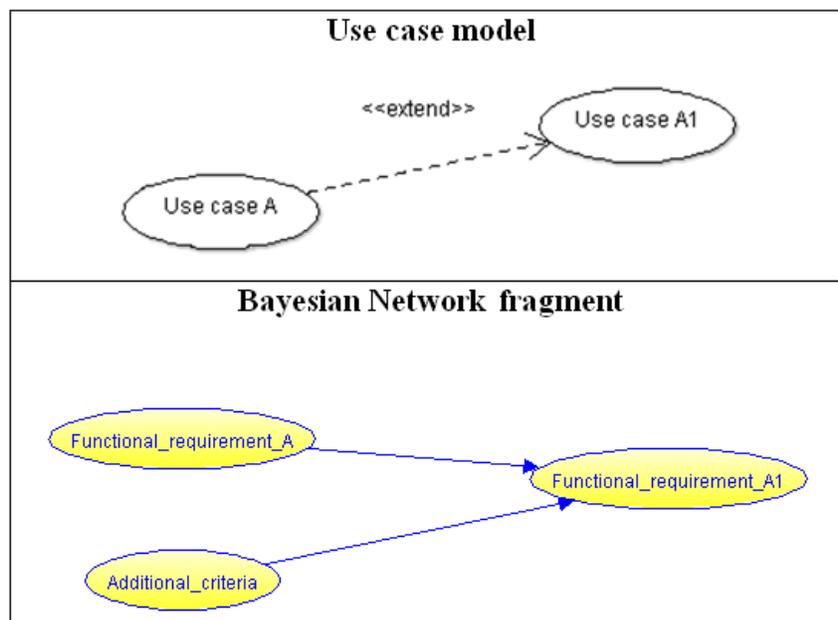
Figure 6. The extension relationship transformed into a Bayesian network fragment

The additional criterion is described in the flow of events in textual description as another functional requirement node. The direction of the implication is an additional criterion (AC) to the functional requirement A1, and functional requirements are included in the Bayesian network, with information represented in cases of use arrows. Therefore:

$$P(A1 \mid A, \; AC) = \frac{P(A \mid A1, \; AC)P(A1 \mid AC)}{P(A \mid AC)}$$

4.2. The Bayesian network developed

The development of a Bayesian network based on the use case diagram for modeling the learner in an adaptive educational system passes through two essential steps:

4.2.1. Specification of the model structure

Taking the case of the node "Learner" to illustrate the stages of development of our Bayesian network representing the learner model, note that this node has three parent nodes (Pretest, Learning Activity and Evaluation), and that each of these nodes is composed of child nodes. Links to these nodes are prerequisite relationships:

- **Learning Activity –** In this node, all students following the course must go through activities of two types, static and interactive, in the adaptive system.

- **Pretest –** All learners must take a pre-test before engaging in the learning activities of each course, The pre-test consists of two types of evaluations:

  ○ Knowledge: the student must answer more than ten questions to measure his or her wealth of knowledge. This type of evaluation reflects the evaluated portion of knowledge of the learner.

  ○ Skills: This is written proof of whether the student can apply the knowledge gained in the module. This type of evaluation reflects the skills portion of the learner.

- **Evaluation** –After the student follows the learning activity, an evaluation is conducted to determine the student's level of knowledge and skill within the module. The evaluation is essential to guide the course of the learner.

The value measuring the relative importance of each condition varies from 0 to 1, and the values of each evaluation element are defined by the teacher, who in this case is the teacher of the module "Database".

The relationship between the target variable (T) and the evidence variable (E) move from T to E, because the process that calculates the posterior probability of the target variable is the proof of knowledge of the diagnosis. Therefore, if the evidence variable has no children, the parents must be the target variables. There are two types of relationships:

- Prerequisite relations between target variables.

- Diagnostic relations of target variables to evidence variables. The control of concepts (targets) affects confidence of evidence. However, if the learner has failed a test, it is unclear if this is due to his lack of knowledge or ability, because there can be an unexpected error.

4.2.2. The specification of variable values

Once the use case diagrams have been created, it is easy to create the structure of the Bayesian network using the rules described in previous sections. Figure 7 represents the Bayesian network constructed from the use case diagram shown in the previous section. Notice how conditional independence was directly modeled by applying the rules as shown.

In the Bayesian network developed, we observe that the node learner (L) has three parents: Learning Activity (A), Evaluation (E) and Pretest (T), which in turn correspond to three weights of prerequisite relationship: w1 = 0.1, w2 = 0.5, w3 = 0.4. Conditional probability of (L) is computed as follows:

$$P\left(L| A,E,T\right) = w1*h1 + w2*h2 + w3*h3$$

Where

$$h1 = \begin{cases} \dfrac{1 \; if \; A = L}{0 \; otherwise} \end{cases}$$

$$h2 = \begin{cases} \dfrac{1 \; if \; E = L}{0 \; otherwise} \end{cases}$$

$$h3 = \begin{cases} \dfrac{1 \; if \; T = L}{0 \; otherwise} \end{cases}$$

We should state that {L, A, E, T} is a complete set of mutually exclusive variables, each of which is also a random and binary variable.
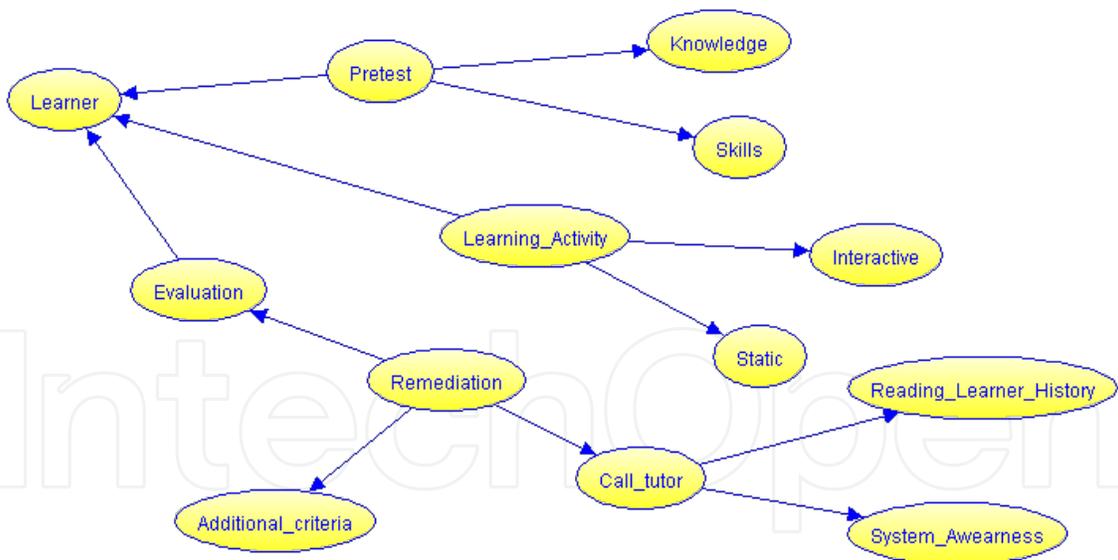
Figure 7. The Bayesian network developed of the learner model

Generalizing the formula below, we state that:

$$P\left(X = 1| Y1,Y2...Yn\right) = \sum_{i=1}^{n} wi*hi$$

where $h1=\begin{cases}1\ if\ Yi=X\\0\ otherwise\end{cases}$ with given random binary variables X and Yi. Obviously, $P(not\ X\ |\ Y1,\ Y2,\ ...,\ Yn)=1-P(X\ |\ Y1,\ Y2,\ ...,\ Yn)$.

a. The Conditional Probability Table of the Node "Learner"

Table 2 represents the CPT of each child node of the parent node Learner.

A	T	E	P(J=1)	1-P(J=1) P(J=0)
1	1	1	1.0 (0.1*1 + 0.4*1+ 0.5*1)	0.0
1	0	1	0.6 (0.1*1 + 0.4*0+ 0.5*1)	0.4
1	1	0	0.5 (0.1*1 + 0.4*1+ 0.5*0)	0.5
1	0	0	0.1 (0.1*1 + 0.4*0+ 0.5*0)	0.9
0	1	1	0.9 (0.1*0 + 0.4*1+ 0.5*1)	0.1
0	0	1	0.5 (0.1*0 + 0.4*0+ 0.5*1)	0.5
0	1	0	0.4 (0.1*0 + 0.4*1+ 0.5*0)	0.4
0	0	0	0.0 (0.1*0 + 0.4*0+ 0.5*0)	1.0

Table 2. The conditional probability table of "Learner" node

Because concepts A, E, and T have no prerequisite knowledge for understanding, their CPTs are specified as prior probabilities obeying uniform distribution, as stated in Table 3 (assigned medium value of 0.5 in most cases).

P(A=1)	P(A=0)	P(T=1)	P(T=1)	P(E=1)	P(E=1)
0.5	0.5	0.5	0.5	0.5	0.5

Table 3. The conditional probability table of "Learner" parents

b. The Conditional Probability Table of the Node "Pretest"

Table 4 represents the CPT of each child node of the parent node Pretest.

K	P(J=1)	P(J=0) 1-p(J=1)
1	0.8(0.8*1)	0.2
0	0.0(0.8*0)	1.0
Sk	P(J=1)	P(J=0) 1-p(J=1)
1	0.2(0.2*1)	0.8
0	0.0(0.8*0)	1.0

Table 4. The conditional probability table of "Pretest" node

c. The Conditional Probability Table of the Node "Learning Activity"

Table 5 represents the CPT of each child node of the parent node Learning Activity.

S	P(J=1)	P(J=0) 1-p(J=1)
1	0.6(0.6*1)	0.4
0	0.0(0.6*0)	1.0
D	P(J=1)	P(J=0) 1-p(J=1)
1	0.4(0.4*1)	0.6
0	0.0(0.6*0)	1.0

Table 5. The conditional probability table of "Learning Activity" node

5. Experiment and validation

In this section, we present the validation tests of the Bayesian network derived from our model of the learner.

The learners involved in the experiment presented herein are students of the module "Database", in the first year of DUT (Technical university diploma) at the Ecole Normale Superieure of Tétouan at Abdelmalek Essaâdi University.

5.1. UnBBayes software

UnBBayes [10] is a probabilistic network framework written in Java. It has both a GUI and an API with inference, sampling, learning and evaluation. It supports BN, ID, MSBN, OOBN, HBN, MEBN/PR-OWL, PRM, structure, parameter and incremental learning.

JAVA UnBBayes uses a technique to reason by odds in intelligent systems. Through a probabilistic network–graph where the nodes are likely variables representing domain knowledge and the arcs represent relationships between them, we can estimate probabilities conditioned to evidence that assists us in decision making. This calculation is called probabilistic inference. With the addition of tree techniques, inferences in probabilistic networks can be made with high efficiency.

To make this technique easy to use, we create the JAVA UnBBayes, which is a visual system that is interactive and platform independent, making it possible to edit, build networks, and show evidence of entry and probabilistic reasoning.

5.2. Metrics

In this section, before presenting the results of our tests, we introduce the metric through which we measure the performance of a learner module modeled using Bayesian networks. The UnBBayes software allows us to evaluate the performance of each node in our network dynamically and in real time. Here are the metrics we used to evaluate our Bayesian network:

- **The global confusion matrix (GCM),** computed for the selected target node and all the chosen evidence nodes.

- **Probability of Correct Classification (PCC):** The probability of correct classification calculated from the global confusion matrix considering all evidence nodes in the Bayesian network.

- **Marginal PCC (MPCC):** The probability of correct classification calculated from the global confusion matrix considering all evidence nodes in the Bayesian network other than the one presented in the row.

- **Marginal Improvement (MI):** The probability of correct classification calculated from the global confusion matrix considering all evidence nodes in the Bayesian network and gained by adding the node presented in the row to the rest of other nodes.

- **Individual PCC (IPCC):** The probability of correct classification computed from the LCM considering only the evidence presented in the row.

- **Cost Rate:** The individual probability of correct classification over the cost ratio.

5.3. The combined Bayesian network

Before presenting the evaluation results of each node of our Bayesian network modeling the learner model in an adaptive system, we begin by presenting the combined Bayesian network through the UnBBayes software.

Figure 8 provides a map of the combined network, in which marginal variables of each node of our network are developed. We can observe the change in the marginal variables of each node in our network, simply by changing one or more marginal variable of one or more parent nodes of the selected node.
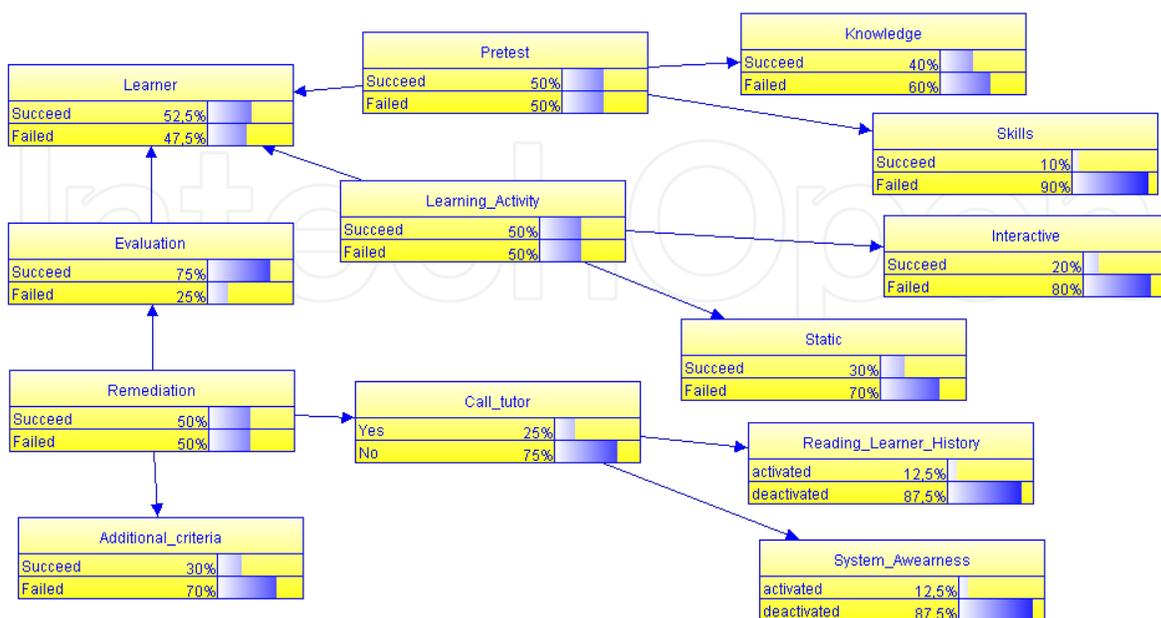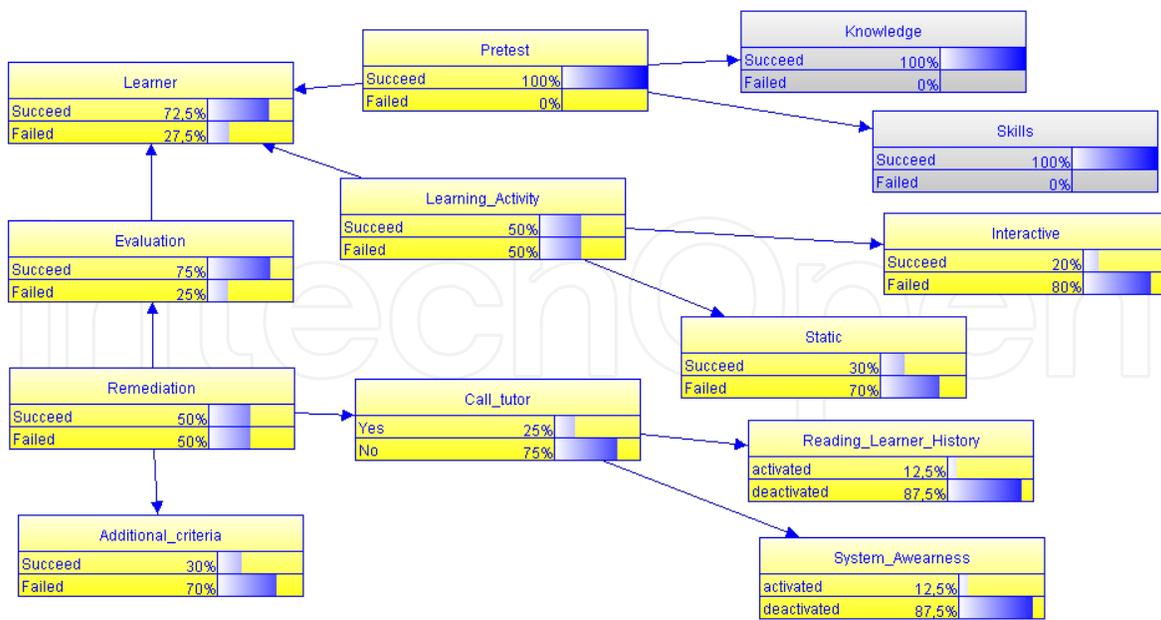

Figure 8. The combined Bayesian network of the learner model

Figure 9. The combined Bayesian network of the learner model

If we change the marginal variable "Succeed" of the node "Knowledge" from 40 % to 100 %, and the marginal variable "Succeed" of the node "Skills" from 10 % to 100 %, we notice that in Fig. 9, the marginal variable "Succeed" of the parent node "Pretest" will change from the initial state of 50 % into a total of 100 % completion. We also notice that a marginal variable of the parent's node of the node "Pretest"—the node "Learner"—will also change from 50 % to 72.5 %.

By changing the information of each node, and after compiling our network, all marginal variables will change automatically, giving us the ability to track in a dynamic way the flow of the learner's path, and to detect the causes of change during all stages of the learning situation.

6. Results

In this section, we present all the results of our experiments on our Bayesian network.

6.1. Node evaluation

To evaluate the performance of each node of our network and its contribution value within a single node or within the entire network, we first began by choosing the node we wanted to evaluate as an evidence node, and chose the parents of these nodes as target nodes. We then defined a sample size that represented how often the software would repeat the simulations.

Using the metrics presented in the previous section, we evaluated the influence of each node within its parent node and within our entire Bayesian network built.

6.1.1. Evaluation of the node "Pretest"

For the pretest node, there are two parent nodes: Knowledge and Skills. We chose the node Pretest as a target node and its parents as evidences nodes, and obtained the results shown in Fig. 10.
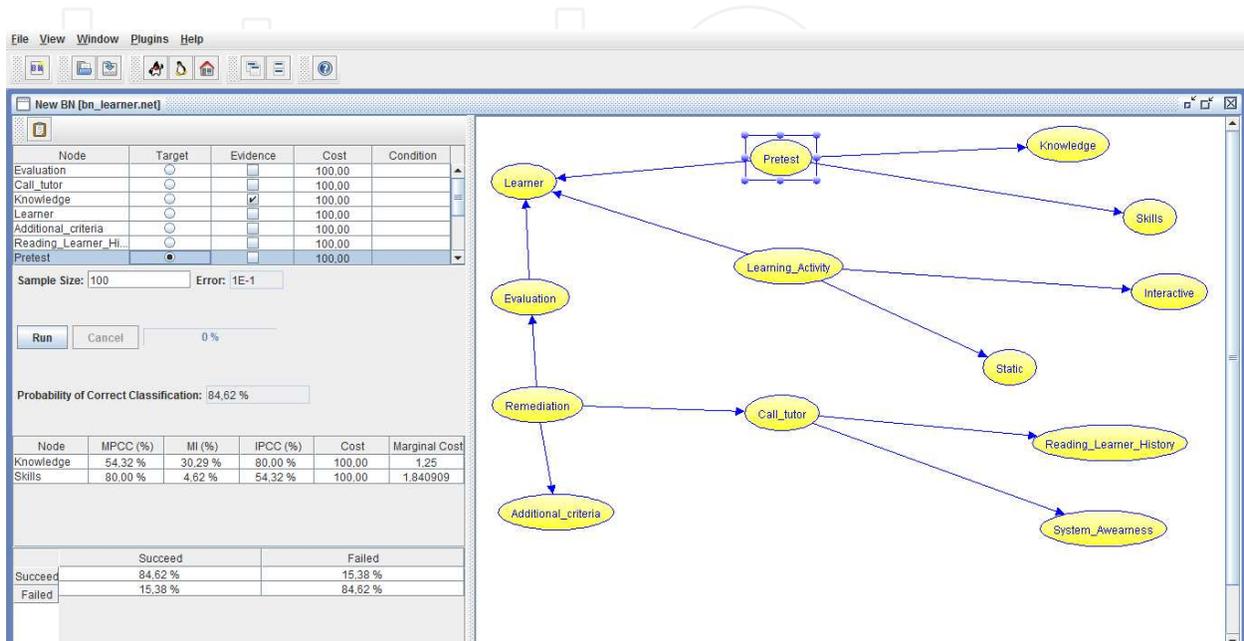

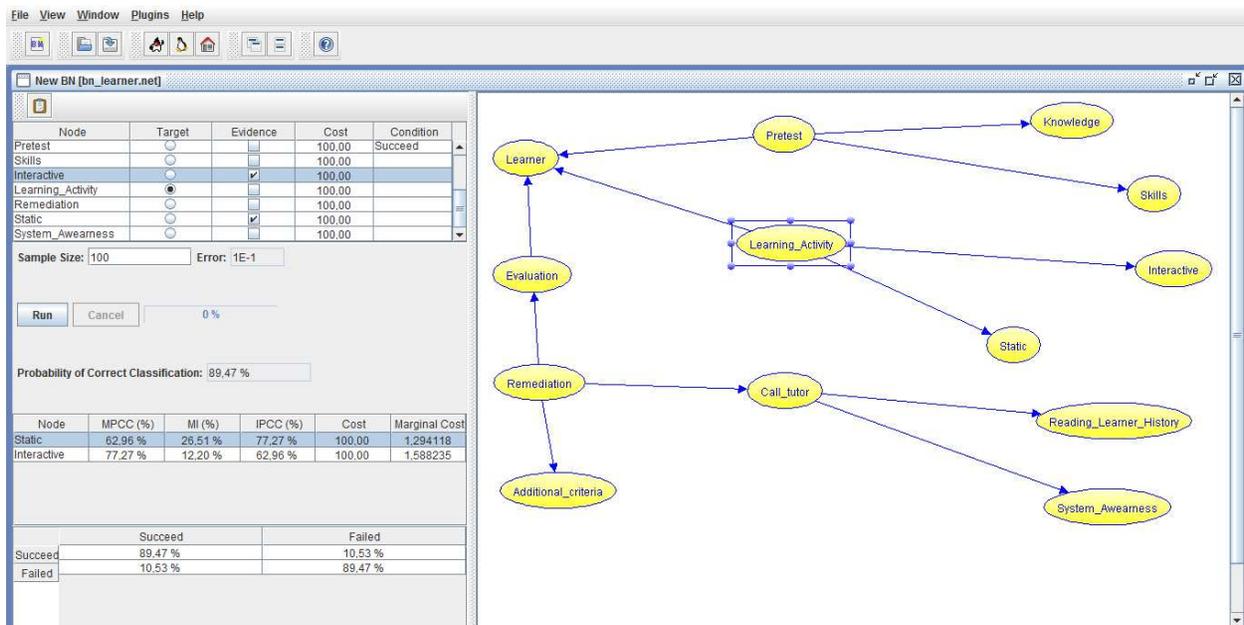**Figure 10.** The evaluation results of the node Pretest

According to the results presented in the table, we find the following. By adding evidence nodes into our evaluation of the target node, the percentage of the probability of correct classifications increases. Furthermore, by measuring the probability of correct classification of each node, we see how each node contributes independently to classification. In this evaluation, we find that the node "Skills" is the node that contributes the most.

We also find how each node contributes with respect to the set of nodes in front of it. In this evaluation, the marginal improvement of the node "Skills" mean that the influence of this node is larger compared to that of the target node. We also notice that even if the marginal cost of the two different sensors is the same, the sensor that is the most evolved reflects the marginal cost of the variables of the node "Skills".

All of this reflects that to pass the pretest, the learner in this learning situation must rely more on skills than on knowledge.

6.1.2. Evaluation of the node "Learning activity"

For the "Learning Activity" node, there are two parent nodes: Static and Interactive. We chose Learning Activity as a target node and its parents as evidence nodes, and obtained the results shown in Fig. 11.

Figure 11. The evaluation results of the node learning activity

According to the results in the table, we find the following. By adding evidence nodes into our evaluation of the target node, the percentage of the probability of correct classifications increases. Furthermore, by measuring the probability of correct classification of each node, we see how each node contributes independently to classification. In this evaluation, we find that the node "Static" is the node that contributes the most.

We also find how each node contributes with respect to the set of nodes in front of it. In this evaluation, the marginal improvement of the node "Static" means that the influence of this node is larger compared to that of the target node. We also notice that even if the marginal cost of the two different sensors is the same, the sensor that is the most evolved reflects the marginal cost of the variables of the node "Static".

All this reflects that the learner in the learning situation has followed a learning activity; the learner must focus on static activity grains more than the grains of interactive activities to increase the chances of succeeding in this learning activity.

6.1.3. Evaluation of the node "Learner"

For the Learner node, there are three parent nodes: Pretest, Learning Activity and Evaluation. By choosing Learner node as a target node and its parents as evidence nodes, we obtain the results shown in Fig. 12.

According to the results in the table, we find the following. By adding evidence nodes into our evaluation of the target node, the percentage of the probability of correct classifications increases. Furthermore, by measuring the probability of correct classification of each node, we see how each node contributes independently to classification. In this evaluation, we find that the node "Learning Activity" is the node that contributes the most.
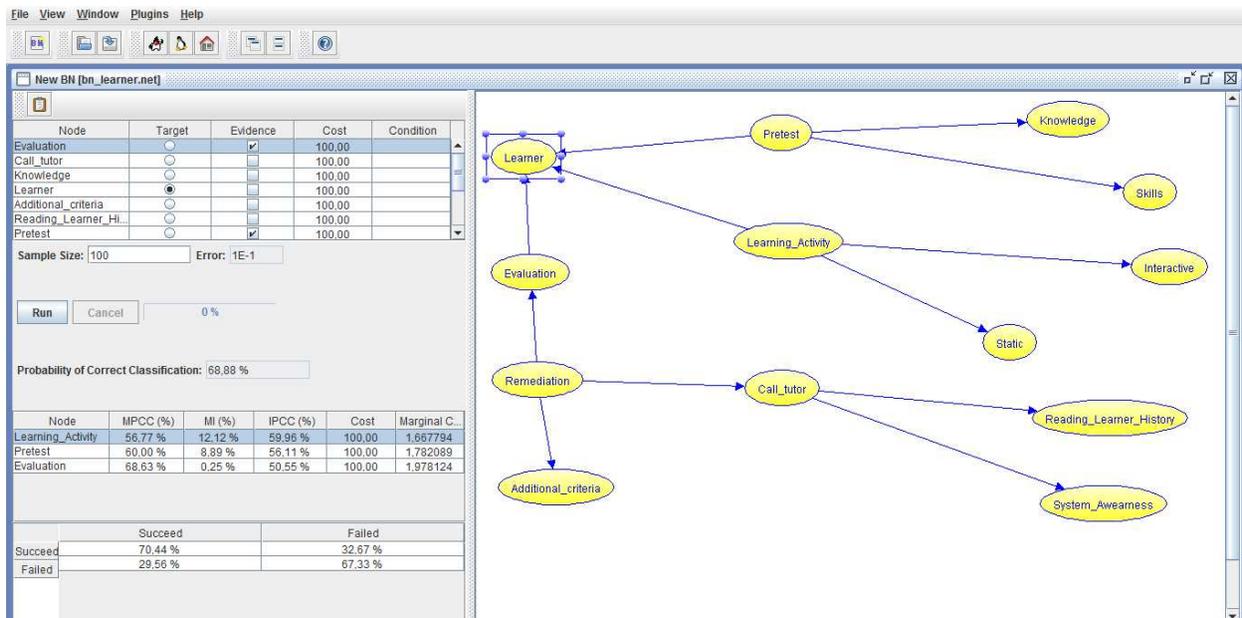
Figure 12. The evaluation results of the node Learner

We also find how each node contributes with respect to the set of nodes in front of it. In this evaluation, the marginal improvement of the node "Learning Activity" mean that the influence of this node is larger compared to the target node. We also notice that even if the marginal cost of the two different sensors is the same, the sensor that is the most evolved reflects the marginal cost of the variables of the node "Learning Activity".

All this reflects that the success of a learner in the learning situation pertains his success in the learning activity more than in the assessment or pretest.

6.2. Bayesian network evaluation

We validated each node of our learner model Bayesian network, and present in this section the validation results of the entire Bayesian network.

Figure 13 presents the entire Bayesian network validation results. In this evaluation of our network, we consider that the learner has successfully passed the pretest and the learning situation. The marginal variable of the node evaluation is 79.71 % in this case. A change in one of these two nodes will affect the marginal variables of our network in a probabilistic manner.

Based on the results and validation of each node of the Bayesian network, we were able to manage the operation of the network in a comprehensive manner.

When a learner begins to take a course in an adaptive hypermedia system, he must first successfully pass the functional requirement "Pretest", which is composed of two functional requirements that measure the learner's knowledge and skills in the chosen field. After validation of the pretest, the learner is automatically assigned to the functional requirement "Learning Activity", which is composed of two types, static and dynamic. At the end of the
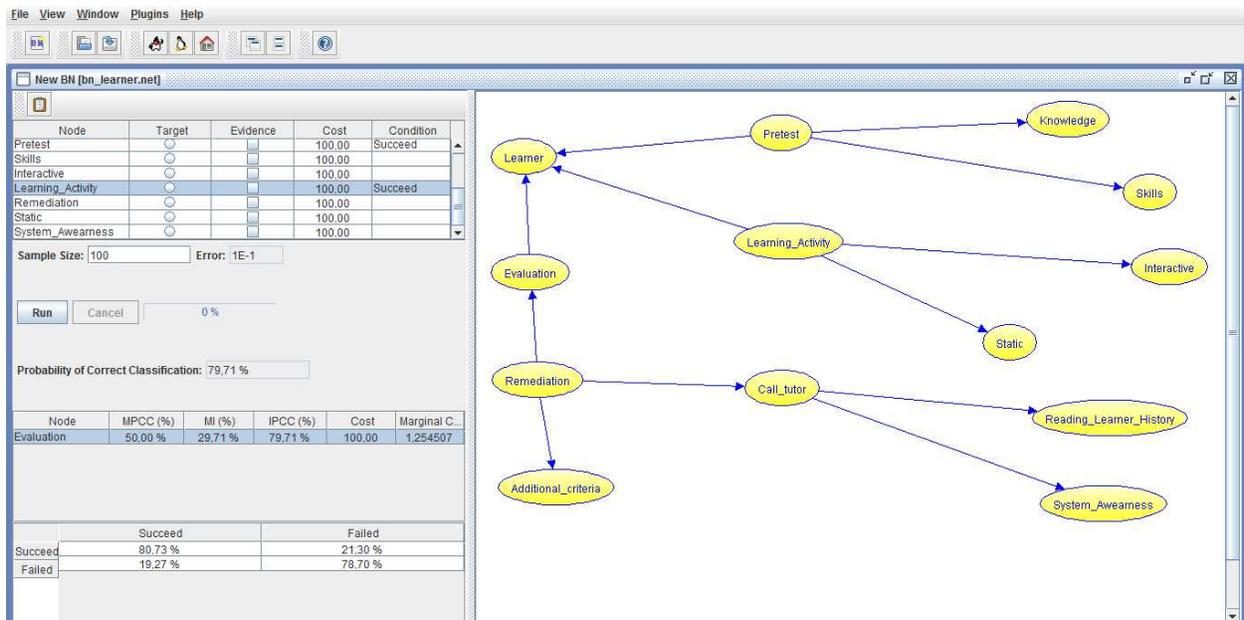
Figure 13. The evaluation of the entire learner model's Bayesian network

course, the learner takes an evaluation expressed in the functional requirement "Evaluation", and the result of this test takes the learner in the case of failure to the functional requirement "Remediation", to retake the learning activities in which the student could not succeed.

Failure in a learning situation requires calling a tutor by activating the functional requirement "Call Tutor", which then activate two functional requirements, "System Awareness" and "Reading History Learner". These two requirements are related to features of the hypermedia system.

7. Conclusion and perspective

We have shown how from a theoretical point of view and considering the analysis of the literature, it seems justified to select Bayesian networks as an effective tool to manage the learner model. The use of Bayesian networks to formally manage the problem of uncertainty in the learner model in an adaptive educational system gives us satisfactory results to address the problem of probabilistic and real-time management of all of a learner's actions in a learning situation.

The experiments presented in this article are arguments in favor of our hypothesis on the modeling of the learner model in a probabilistic way, using all the nodes as sensors to measure and evaluate the entire model.

The proposed rules for processing use case diagrams that schematize the actions of a learner in an adaptive system can be applied to many use cases in different systems.

We see two main directions in which to continue this work; on the one hand, by combining Bayesian networks with other modeling methods of the learner, such as overlay models; and on the other hand, by transforming the Bayesian networks developed for the management of the learner model into a machine-readable language, such as ontologies. Or, as we already proposed [11], by using probabilistic ontologies as a formalism that gives us the possibility to combine Bayesian networks with ontologies.

Author details

Anouar Tadlaoui Mouenis*, Aammou Souhaib and Khaldi Mohamed

*Address all correspondence to: tadlaouimouenis@gmail.com

LIROSA / Faculty of Sciences, Abdelmalek Essaâdi University, Morocco

References

[1] Zaitseva L., Boule C. (2005). Learning systems in professional training. Proceedings of Workshop "Industry meets research" within the conference Interactive Computer Aided Learning, ICL Villach, Austria.

[2] Beck J., Stern M., Haugsjaa E. (1996). Applications of AI in education. Crossroads, 3(1), 11–15.

[3] Brusilovsky P., Millan E. (1991). User models for adaptive hypermedia and adaptive educational systems. The Adaptive Web Book. Springer, Berlin/Heidelberg, 2007.

[4] Self J. Formal Approaches to Student Modeling. Technical Report AI-59, Lancaster University, England.

[5] Conati C., Gertner A., Vanlehn K. (2002). Using Bayesian networks to manage uncertainty in student modeling. Journal of User Modeling and User-Adapted Interaction, 12(4), 371–417.

[6] Pearl J. (1988). Probabilistic reasoning in intelligent systems. Morgan Kaufmann, San Mateo.

[7] Mayo M., Mitrovic A. (2001). Optimising ITS behaviour with Bayesian networks and decision theory. International Journal of Artificial Intelligence in Education, 12, 124–153.

[8] Souhaib A. (2013). Conception de l'intéropérabilité sémantique des systèmes hypermédia adaptatifs éducatifs, basée sur les ontologies. Ph.D thesis, Abdelmalek Essaidi University.

[9] Mouenis A.T., Souhaib A., Mohamed K. (2015). Development of Bayesian networks from unified modeling language for learner modelling. International Journal of Advanced Computer Science and Applications, 6(2).

[10] Costa P.C.G., Ladeira M., Carvalho R.N., Laskey K.B., Santos L.L., Matsumoto S. (2008). A first-order Bayesian tool for probabilistic ontologies. Proceedings of the 21st Florida FLAIRS UNBBayes-MEBN, Coconut Grove, FL, USA.

[11] Mouenis A.T., Mohamed K., Souhaib A. (2014). Towards probabilistic ontology based on Bayesian Networks. International Journal of Software and Web Sciences, 1(10), 102–106.