

An Authoring Tool based on Probabilistic Relational Models for Building Intelligent Tutor Systems

M.A. Romero-Inzunza, Pilar Gómez-Gil, L.Enrique Sucar
National Institute of Astrophysics, Optics and Electronics.
Department of Computational Science. Puebla, México, 72840
{m_romeroinzunza, pgomez, esucar}@inaoep.mx

Abstract. A novel authoring tool for aiding teachers in building intelligent tutoring systems (ITS) for virtual laboratories is presented. The main contribution of this tool, called HEDEA, is that it allows a professor to easily define the student model using a syllabus and variables associated to virtual experiments. The information given by the instructor is transformed automatically into a probabilistic relational model (PRM), which is the core of the ITS. The ITS, automatically generated by the tool, guides a student performing experiments in a virtual lab, evaluates the performance in each experiment, gives the student personalized help, and defines the next experiments. The virtual lab is connected to the ITS via a set of common variables that summarize the performance of each experiment. A teacher, which does not require expertise on ITS or probabilistic models, defines a set of experiments to be executed by the virtual lab, and the experiments variables that the ITS uses to deduce the student's knowledge by probabilistic inference in the PRM. A tutor that supports a small virtual lab for back-propagation neural networks was developed using this tool, obtaining promising results. An assessment of the usability of the proposed tool is presented.

1. Introduction

An intelligent tutoring system (ITS) provides personalized instruction that is adapted to each student [1]. ITS have a great potential to improve education, however the main challenges faced today for their use include the huge time required for the development of ITS [2] and the need of experts in AI, software engineering and education to develop them. To face these challenges, some authoring tools for ITS have been developed in the last years [2] [3] [4], several directed to users with no knowledge about programming or technical aspects related to ITS. In the other hand, most of these tools are designed mainly for cognitive tutors, so they are not appropriate for designing ITS for virtual laboratories and in particular, they do not consider explicitly the uncertainty in the student model.

The development of an authoring tool offers several interesting challenges, because the tool must be able to capture the didactical intentions of the teacher, to represent the student knowledge in a meaningful way and to build a tutor able to interact with a virtual lab or some other kind of software able to modify and assess the knowledge acquired by the student. In this article, we present a tutoring tool, called HEDEA (*HERRamienta DE Autoría*) able to build an ITS that will interact with a virtual lab previously developed, and that does not requires the teacher any expertise on ITS. The

ITS generated by HEDEA guides a student performing experiments in a virtual lab, evaluates the performance in each experiment, gives the student personalized help, and sets up the next experiments (Figure 1). HEDEA is based on a probabilistic relational student model defined by Noguez [5], which represents the student's knowledge at different levels, and guides the tutoring process. Based on a curricula definition given by the teacher and associated to the virtual lab, the tutor generates a Bayesian network for the student model, as well as its initial probabilities. The Bayesian student model is the core of the ITS, which is complemented by a set of production rules that constitute the tutor module, and which are also generated automatically; and a set of lessons that are given by the instructor and form the knowledge base. Once the tutor is generated, it interacts with the virtual lab through a set of common variables defined by the instructor, and which are used to evaluate the performance of the student. Using this information, the student model is updated via probabilistic inference (using toolbox ELVIRA [6]), and based on the model the tutor gives lessons to the student and sets up the next experiments. To test the usability of HEDEA, a tutor to support a neural networks virtual lab was built and the results are reported here.

The paper is organized as follows: section 2 presents a brief introduction to intelligent tutor systems, section 3 describes the main components of HEDEA, section 4 evaluates its usability and section 5 discusses some conclusions and future directions of this research.

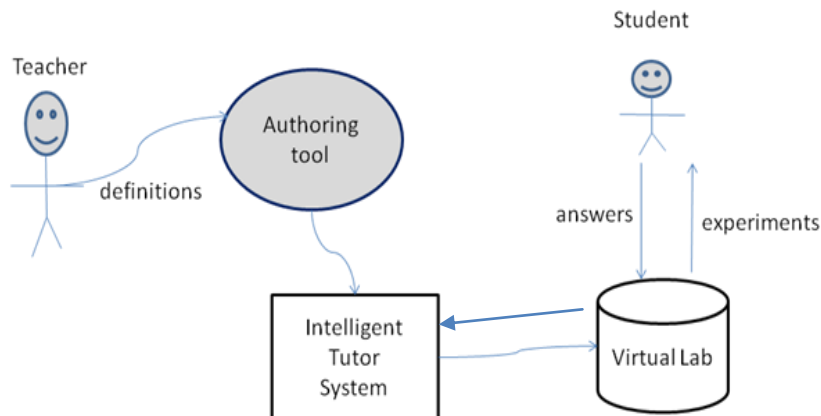


Figure 1. A context diagram of HEDEA. A teacher interacts with the authoring tool to define the syllabus and variables for evaluating the experiments; the ITS is generated and connected to the virtual lab which provides personalized assistance to a student.

2. Intelligent tutor system

An ITS is composed of the three main modules [7] described next: student model, tutor and domain knowledge.

Student Model. This represents the knowledge that a student has at any time. Given that an ITS has to deduce the student's cognitive state based on his/her interaction

with the tutor, there is inherent uncertainty in this process, so in general probabilistic representation are considered, such as Bayesian networks [8] and probabilistic relational models (PRM) [5]. PRM are an object-oriented formalization of Bayesian networks, integrating the advantages of probabilistic models and relational logic [9]. In a PRM, each class is an object with Bayesian network nodes as components, which allows representing several Bayesian networks in the same model. HEDEA bases its student model on the PRM representation proposed by Noguez [5] (see Figure 2). To define the model a hierarchical scheme is used; from the student general category (top node), to his/her knowledge of the different knowledge items, until the variables that summarize the results of each experiment, at the bottom. A PRM is first defined at a class level, including a set of attributes for each class, each corresponding to a node. The dependence model is specified at class level and the inference mechanisms are similar to the ones used by Bayesian networks. In this way a PRM can be seen as representing a set of Bayesian networks, one for each student and for each level. After a student performs an experiment, the values of the results variables are updated, and his/her current knowledge of the different topics is updated via probability propagation in the network. The student module communicates with the tutor module so that the tutor may be able to take the best decisions about what experiments should be presented to the student in order to improve his/her knowledge, and if necessary, give a lesson according to the knowledge items that present a low probability.

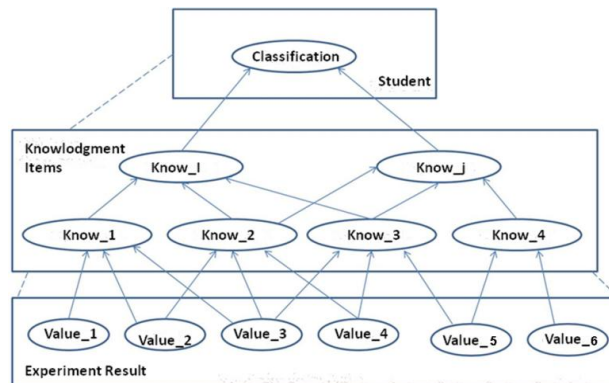


Figure 2. A PRM to represent student knowledge (adapted from [5]). It is a hierarchical model with 3 main layers: top, student category; middle, knowledge items; bottom, variables that summarize the results of an experiment.

Domain Module. Also known as knowledge base, it contains all material to be taught and experiments to be executed by students. In the case of HEDEA, this module contains material defined by the instructor, organized in lessons at different levels of abstraction. The lessons are presented to the student according to the results of the experiments, based on the student model.

Tutor Module. This contains information about the teaching process. The decision of this process is made based on the student model. HEDEA uses a simple production

rule mechanism, so that each rule relates the information in the student model to the lessons and experiments to be selected for the next phase in the tutoring process.

3. The Authoring Tool

The objective of HEDEA is to generate an ITS based on the information provided by the instructor and supported by an existing virtual lab. The tool automatically generates a Bayesian representation, using the structure of the syllabus input by the teacher. The teacher also inputs information related to the experiments that the student will execute in the virtual lab. Such information includes names and correct values of variables that the tool links with the Bayesian model. Each time that an experiment is executed, the Bayesian model is updated. The representation of the model allows a dynamical update of values in the network nodes, done in a way that knowledge acquired by previously executed experiments is not lost. Once that syllabus and experiments have been designed, the instructor registers the lab in the authoring tool. Using this information HEDEA generates an ITS, and the student may interact with the virtual lab by an interface built also by HEDEA. Such interaction is implemented using text files, to simplify communication among the ITS and the virtual lab. Next we described the generation of each component of the ITS.

Student Model. This is the core of the ITS and its automatic generation constitutes the main contribution of this work. As mentioned before, the model represents the knowledge of the student of the different topics related to virtual lab at different levels of abstraction, using a PRM. The instructor has to define a syllabus in a hierarchical way, from themes to sub-themes to concepts, which constitutes the structure of the knowledge items using 3 levels of abstraction. For instance, part of the knowledge structure for a neural networks laboratory is depicted in figure 3.

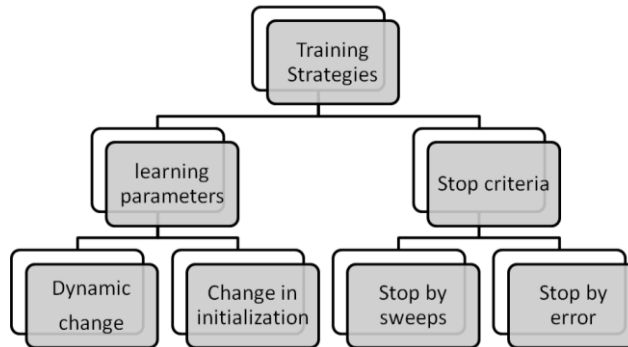


Figure 3. Hierarchical knowledge structure for a neural networks tutor.

The instructor also defines the importance of each knowledge item with respect to the upper level in the hierarchy. That is specified as a weight in the range 0 to 100%, such that all the items related to an upper level must sum 100%. These weights are used to build the conditional probability tables (CPTs) of the Bayesian net, as detailed below. Finally the instructor specifies the relations between the variables that are monitored

in the lab, to the concepts in the knowledge hierarchy. By default it is assumed that all lab variables related to certain concept have the same importance, but the instructor can specify an importance for each variable. Once the professor has specified this information, the student model is generated. The structure of the Bayesian network is deduced from the knowledge hierarchy and the related laboratory variables, such that each item constitutes a node in the network, and relations correspond to the arcs (dependencies). A probabilistic model obtained from the neural network example in figure 3 is shown in figure 4. Each node determines values of “know” or “do not know” associated to its corresponding theme, sub-theme or concept. Leaves represent experiment variables, and they get a value according to a grade assigned by the virtual lab, which ranges from 0 to 100.

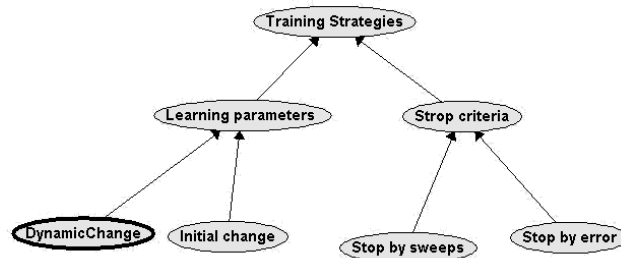


Figure 4. Bayesian network generated from the hierarchy in figure 3.

The CPTs of the Bayesian model are obtained from the importance weights given by the teacher. Each node in the Bayesian network is binary, representing if the student knows (1) or does not know (0) the corresponding item. The probability table is obtained from the weights based on a NOISY-OR model [10], such that if the result is correct for all the variables, the concept is known; if all are wrong, the concept is not-known, and otherwise it has intermediate values based on the weights. Figure 5 shows an example of one of the CPTs for the Bayesian network of figure 4.

DynamicChange\Initial change	Know	Not known	Know	Not known
Know	1.0	0.7	0.3	0.0
Not known	0.0	0.3	0.7	1.0

Figure 5. An example of a CPT generated for the Bayesian network model in figure 4.

From the base structure defined above, a dynamic Bayesian network is generated in order to merge student knowledge acquired from past experiments with knowledge acquired during the last experiment. The dynamic network is analogous to the base network; it has two additional nodes for each node in the base network: a node with the previous value before the experiment is realized, and a node with next value after the inference process is executed. Inference is done using the toolbox ELVIRA [6].

Domain Knowledge. In order to generate the knowledge base to be used by the ITS, all the instructional materials are defined and registered in HEDEA, at different granularities according to the knowledge hierarchy. Each lesson is linked to the appropriate level of the syllabus, so the tutor can select and display a lesson according to the student model.

Tutor Module. The tutor module performs two basic operations based on the student model: (i) if the model detects that the student is deficient in certain area (concept, sub-theme, theme), it gives his/her a lesson at a certain level of granularity; (ii) based on the overall evaluation of the student, it selects the next experiment. In order to define if a student knows or not a subject, a threshold is defined by the instructor (for example, he/she may decide that a grade above 60% means knowledge acquired). The tutor is represented as a set of automatically-generated production rules, where the antecedent corresponds to each node (knowledge item) in the student model, and the action is to display the corresponding lesson.

Once an ITS is generated, it has to be linked to the virtual lab to be used by a group of students interacting with the virtual lab. Based on the concept of PRMs, a copy of the Bayesian network that represents the student model is generated for each student, so there are as many instantiations as students.

4. Evaluation

A small virtual lab [11] was built to test the functionality of HEDEA. This lab has the aim to help in teaching the main concepts associated to back-propagation neural networks. The lab was programmed using Matlab 7.0 and its Neural Network Toolbox 4.0.1. A syllabus containing 2 basic themes, 8 sub-themes and 10 concepts was defined by an instructor (part of this syllabus was depicted in figure 3). Three experiments were developed, one for each knowledge level (basic, intermediate and advanced). Each experiment has 4 performance variables that were associated to the ITS. Depending on the experiment, the right values or range of values for each variable in each experiment is defined. A professor, expert in neural networks but not on ITS, used HEDEA to generate an ITS for the neural nets lab. HEDEA may be evaluated from three different perspectives:

1. Assessing the *functionality of the tool* and the process required to generate an ITS.
2. Assessing the *functionality of the ITS* generated by the tool, with respect to its ability to correctly represent the knowledge acquired by the student. Part of this ability depends on the performance of the virtual lab connected to the ITS and the ability of the instructor to define an appropriate set of experiment variables and their weights.
3. Assessing the *impact of the tutor* in learning. This requires a controlled experiment with a group of students interacting with the ITS.

Our evaluation is centered on the first aspect, functionality, measured in terms of usability, that is, “*the extent to which a product can be used by specified users to achieve specified goals with satisfaction in a specified context of use*” [12]. Usability may be evaluated mainly by *learnability, efficiency, user retention over time, error rate* and *satisfaction*. Given the fact that at the time of publishing this results HEDEA is still a prototype, only learnability (easiness to learn the main system functionality) and satisfaction (overall and subjective impression of the system given by the user) were considered. These two concepts were evaluated using the questionnaire shown in table 1, applied to the instructor who designed the syllabus and experiments of the neural network virtual lab. The answers given by the instructor are shown in table 1.

Table 1. Evaluation questionnaire to assess user’s learnability and satisfaction.

Question Number	Question description	Importance	Attribute	Grade (0 – 3)	Sub-total
1	To construct a syllabus with weighted concepts is easy.	0.2	Learnability	2	0.4
2	I was able to learn to use HEDEA very quickly.	0.2	Learnability	2	0.4
3	It was easy to follow the tool instructions.	0.2	Learnability	1	0.2
4	The steps to register experiments were easy to follow.	0.1	Satisfaction	2	0.2
5	I would recommend HEDEA to my colleges.	0.2	Satisfaction	2	0.4
6	HEDEA allowed me to correct or change my inputs easily.	0.1	Satisfaction	1	0.1
TOTAL GRADE (maximum value 3):					1.7

From the evaluation can be noticed that, at this point, three aspects need to be improved: the human-computer interface and user manual (related to question 3) and editing facilities (related to question 6).

Regarding the functionality of the generated tutor we have performed some preliminary tests by setting the values on the experimental results and analyzing the probabilities of the different knowledge items in the student model. According to the instructor, the inferred probabilities seem coherent to the results of the experiments (for more details see [14]).

To assess the impact of the tutor on learning, a control study will be done in near future using a group of students interacting with the neural network ITS generated. We expect a positive impact given previous results obtained by Noguez and Sucar [13] with a manually-generated tutor in the robotics domain. They found that students interacting with the virtual robotics lab have a significantly higher learning gain than those that used the virtual lab without a tutor.

9. Conclusions

We have described HEDEA, a novel authoring tool for building ITS. HEDEA receives as inputs a syllabus, a description of experiments to be executed in a virtual lab, and the relations between them via a set of variables; and generates an ITS that is able to interact with the virtual lab and provide personalized help to the students using the lab. The core of the ITS is a probabilistic student model that is generated automatically from the syllabus and weights given by the instructor, so that the internal representation is transparent to the user. Based on this model, the generated ITS can provide lessons to the students according their performance in the previous

experiments. The tool was able to generate a basic ITS for a small virtual lab to teach back propagation neural network. The usability of the tool was assessed by an instructor with promising results. HEDEA is a prototype and we are currently improving the human-computer interface and the communications schemes with the virtual lab, in order to assess its usability for more complex cases. In the future we plan to conduct a control user study using an ITS generated with HEDEA.

6. References

1. Freedman, R., Ali, S.S., McRoy, S.: What is an Intelligent Tutoring System? *Intelligence (Intelligence)* #3, 15-16 (2000)
2. Alevan, Vicent, McLaren, Bruce M. , Sewall, Jonathan, Koedinger, Kenneth R.: The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. *Proceedings of the 8th International Conference on intelligent Tutoring Systems*. Jhongli, Taiwan, 26-30 (2006)
3. Murray, Tom.: *Authoring Intelligent Tutoring Systems: An Analysis of the State of Art*. *International Journal of Artificial Intelligence in Education*, 98-129 (1999)
4. Ferrero, B., Arruate, A., Fernandez-Castro, I., Urretavizcaya, M.: Herramientas de Autor para Enseñanza y Diagnóstico: Iris-D. *Revista Iberoamericana de Inteligencia Artificial* No. 12, 13-28 (2001)
5. Noguez, Julieta.: *Modelo probabilista relacional del estudiante para laboratorios Virtuales*, Tesis Doctoral. Cuernavaca, Morelos: ITESM, Campus Cuernavaca (2005)
6. Elvira, Elvira. 01 01, 2001. <http://leo.ugr.es/elvira/> (accessed 01 18, 2009)
7. Salgueiro, Fernando, Cataldi, Zulma, Lage, Fernando, García-Martínez, Ramon: *Sistemas Tutores Inteligentes: Redes Neuronales para Selección del Protocolo Pedagógico*. IV Workshop de Tecnología Informática Aplicada en Educación. Buenos Aires, Argentina, 255-266 (2005)
8. Conati, C., Gertner, A., Vanlehn, K., Druzdzel, M.: On line student modeling of coached problem solving using Bayesian networks. *Using Modeling: Proceedings of Sixth International Conference*. New York, USA: Springer Verlag, 231-242(1997).
9. Getoor, Lise, Friedman, Nir, Koller, Daphne, Pfeffer, Avi, Taskar, Benjamin: *Probabilistic Relational Models*. In *An Introduction to Statistical Relational Learning*, by L. Getoor and B. Taskar, 129-174. MIT Press, (2007)
10. Díez, F.J., Druzdzel, M.: *Fundamentals of canonical models*. In *Proc. IX Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA)*, Oviedo, Spain (2001)
11. Gómez-Gil, P.: *Diseño de un Laboratorio Virtual Básico de Redes Neuronales de Retropropagación* (Design of a basic virtual lab for backpropagation neural networks). Technical report. Neural Net Research Lab. National Institute of Astrophysics, Optics and Electronics. Department of Computer Science, (2009)
12. Ferré, X., Juristo, N., Windl, H., Constantine, L.: *Usability Basics for Software Developers*. *IEEE Software* 18,1, 22-28 (2001)
13. Noguez, Julieta, Sucar, L. Enrique A *Probabilistic Relational Student Model for Virtual Laboratories*, *enc*, pp.2-9, *Sixth Mexican International Conference on Computer Science (ENC'05)* (2005)
14. Romero-Inzunza, M.A.: *Herramienta de Autoría para Tutores Inteligentes Basados en Modelos Probabilistas Relacionales*. Annual Report. Maestría en Ciencias en el Área de Ciencias Computacionales. National Institute of Astrophysics, Optics and Electronics. Department of Computer Science (2009)