

# A Learner Model in a Distributed Environment

Cristina Carmona and Ricardo Conejo

Departamento de Lenguajes y Ciencias de la Computación  
Universidad de Málaga  
{cristina, conejo}@lcc.uma.es

**Abstract.** A learner model must store all the relevant information about a student, including knowledge and attitude. This paper proposes a domain independent learner model based in the classical overlay approach that can be used in a distributed environment. The model has two sub-models: the learner attitude model, where the static information about the user is stored (user's personal and technical characteristics, user's preferences, etc.) and the learner knowledge model, where the user's knowledge and performance is stored. The knowledge model has four layers: estimated, assessed, inferred by prerequisite and inferred by granularity. The learner model is used as a part of the MEDEA system, so the first and second layers are updated directly by the components of MEDEA and the third and fourth are updated by Bayesian inference.

## 1 Introduction

In an educational environment, a good learner model must include all features of the learner's knowledge and preferences that concern to his learning and performance [1]. This information is used to adapt the system to the user. However, to build this model is a very difficult task. In practice, a partial model is used. It is necessary to take into account (1) what information is included in the model; (2) how to obtain it; (3) how the model will represent the information; and finally, (4) how the model will process and update the information.

Actually, there are two kind of information stored in a learner model: the first one includes information that does not change over the learning process, like user's particular characteristics (name, age, gender...), learner's capabilities (degree, background knowledge...), learner's technical characteristics (computer expertise, connection speed...), learner's preferences (learning style, screen options...), etc. This information is usually collected at the beginning of the learning process using forms and tests. The second includes those that change over the learning process, like the learner's knowledge level for each knowledge unit, skills, goals, etc. This information can be obtained directly from the learner (goals), from tests (tests marks shows how much the user knows about the subject) or from the learner interaction with the system (number of pages visited, links selected, time spent on each page...).

One method widely used to represent the learner knowledge model is the overlay model where the learner knowledge is represented like a subset of expert knowledge [4-7]. A strict overlay model contains only a subset of the expert knowledge. It is called a perturbation model if it also includes information about incorrect knowledge. Another popular method is the use of stereotypes. Each stereotype represents some

common characteristics in learning style, knowledge, etc. and the instruction is adapted to those features. Then the learner is classified in one or more stereotypes and inherits those features and also the adapted instruction [2, 3]. Stereotypes are often used in combination with other methods.

This paper proposes a learner model to be used in MEDEA [8]. MEDEA is an open system to develop Intelligent Tutorial Systems (ITS). It is composed of independent educative modules coordinated by a core that controls the instruction of the learner. MEDEA can decide which is the best module to explain a concept in each moment. The model has two sub-models: *the learner attitude model*, where the static information about the user is stored (user's personal and technical characteristics, user's preferences, etc.) and *the learner knowledge model*, where the user's knowledge and performance is stored. The knowledge model is based on the classical overlay approach with four layers: *estimated*, *assessed*, *inferred by prerequisite* and *inferred by granularity*.

This paper explains briefly the architecture of MEDEA and then it centers in the learner model proposed.

## 2 MEDEA

The elements that compose MEDEA architecture can be classified in three main groups: those that contain knowledge (*knowledge modules*), those that use this knowledge for making decisions along the instruction (*functional modules*) and those that serve to access and configure the system (*tools*). The base of MEDEA architecture is a core that plans the instruction based on a set of external tutorial components that are connected to the system. The domain and pedagogical knowledge is distributed between the core of MEDEA that serves as a master index, and these components. Fig. 1 shows the structure of MEDEA modules:

### Knowledge Modules:

- *Domain Model*. This module contains knowledge about the subject to be taught. Domain concepts and relationships among them are represented.
- *Learner model*. This is the goal of this paper and will be described in next section.

### Functional Modules:

- *Instructional planner*. This module will provide students with the necessary guidance during the learning process. It will design and compose the tutorial sessions, that is, it will decide in each moment the correct task to be performed by the student.
- *Learner Model Manager*. The function of this module is to create and update the learner model. The learner model is updated every time the learner interacts with the system.
- *The library of tutorial components*. A tutorial component is an external educational tool that is able to complete a tutorial task as make tests, present theory contents in hypertext, play a game, etc. MEDEA classifies tutorial components as *assessment components* [10] or *information components* [9]. The difference is that components of the first type are able to assess the student knowledge level about a concept.

### Tools:

- *Configuration and definition tools.* Domain experts, teachers and designers will use this module, they will be able to introduce the contents, define and configure the data and knowledge modules using specific interfaces.
- *Administration tools.* Teachers will use this module to monitor the evolution of their students. It will show the progress of each student, the statistics about the course use, and average student performances, and other secretariat and administrative tasks.
- *Navigation tools.* Student uses this module to support their navigation and interaction with the whole system; it can be conceptualized as an advisor during the learning process.

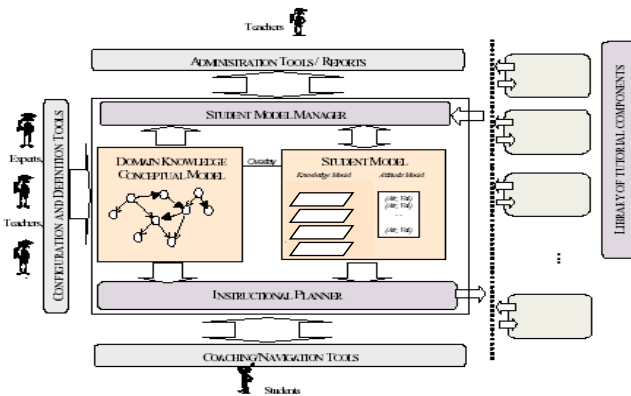


Fig. 1. MEDEA architecture

### 3 The Learner Model in MEDEA

The learner model in MEDEA is divided in two main sub-models: the *Attitude Model*, which contains those features that describe the student profile but are not related to his current state of knowledge and the *Knowledge Model*, which contains information about the learner's state of knowledge.

The *Knowledge Model* proposed is implemented by a multi-layer overlay model. The main characteristic of the overlay model is that for each domain concept an estimation of the user knowledge level on this concept is stored. If the model has multiple layers, it can have information from different sources. This information can be updated independently and without overwriting. The proposed model has four layers: (1) *estimated layer* that collects the indirect information and inferences of the student knowledge level based on the student behavior during the instruction. This value is given by the *informative components*; (2) *assessed layer*, it contains the marks obtained by a student using the systems *assessment components*; (3) *inferred by prerequisite layer*, this layer contains the values inferred for each concept via a Bayesian network that represents the *prerequisite* relationship between concepts; and (4) *inferred by granularity layer*, that contains the values inferred for each concept via a Bayesian network that represents the *belongs-to* relationship between concepts. No-

tice that the data of the first two layers are updated directly by the components, (they are considered as rough data) while the two later are updated by the learner model manager, as described in the next section.

The *instructional planner* can use all these layers to select the best concept to show in each moment. It is up to the planner strategy to decide what information to use. The role of the learner model manager is just to provide this information. For instance, a learning strategy that can be implemented by the planner would be to trigger an assessment component whenever there is a significance difference between values in the estimated layer and assessed layer. Another strategy can be to fully complete the study of a concept before entering a new one. Then the planner will look into the *inferred by granularity layer* to select the concept that is not completely known from a group of concepts already known, etc.

### 3.1 The Attitude Model

The Attitude Model contains static information of the student. It includes user's particular characteristics, user's technical characteristics and user's preferences. This information is obtained directly from the learner the first time he uses the system and each time he registers in a course. These data can be updated by user demand. The main features for a learning process are explained in more detail in Fig. 2 that shows the features and all its possible values. Course designers use the learner attitude model to establish relations between a concrete learner profile and some instruction parameters. For example, they can specify in the course definition that when a learner with low motivation level does a test, it is better to show him the right answer each time he makes a question, rather than show all the right answers at the end.

FEATURE	VALUES
<b>Level:</b> Cognitive development (formalization and abstract concepts understanding skills)	Beginner, Medium, Advanced
<b>Motivation</b>	High, Medium, Low
<b>Learning style</b>	Practice, Theory
<b>Progress:</b> The student learning speed	Poor, Regular, Good
<b>Computer expertise:</b> The student experience with computers	High, Medium, Low
<b>Connection speed:</b> Internet connection speed	High, Medium, Low

Fig. 2. Main features and values of the student attitude model

### 3.2 The Knowledge Model

Each layer of the Knowledge model is a list of elements that mainly contains the concept and the mark obtained by the student for that concept.

During the execution of MEDEA tutoring session, the learner is conducted through different informative and assessment components. The result of the interaction with each component is returned to MEDEA that calls the *learner model manager* to update the estimated and assessed layer respectively. This call also triggers the inference process over the prerequisite layer and granularity layer. The assessed values are used as initial values of certain nodes of a Bayesian networks that is explained in more detail in the next section.

### 3.2.1 Bayesian Networks for the Prerequisite and Granularity Layers

Domain model in MEDEA is structured by a set of nodes (concepts) and different relationships between them. Currently this learner model is just interested in *belongs-to* and *prerequisite* relationships. This approach is similar to the classical domain representation in ITS. For instance Fig. 3 shows an example of the domain model of a course divided in topics, subtopics and atomic concepts [11]. Each node in this graph is considered a concept in MEDEA. The light arrows represents the *belongs-to* relationship, (*granularity layer*), while the bold arrows represents the *prerequisite* relationship. (*prerequisite layer*). A Bayesian network is obtained from each type of relationship. This graph is divided in two, to assure that the Bayesian conditional probabilities comply with the independence conditional hypotheses. Each node of the network  $C_i$  can take a value  $x_{ik}$  among a finite number of possible values, representing the knowledge level of the learner for the corresponding concept  $i$  (i.e.:  $C_i$  can take values {low, medium, high}).

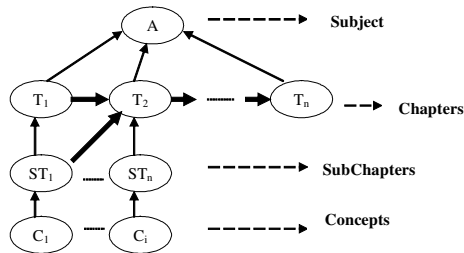


Fig. 3. Concepts' network

In order to initialize each network, the conditional probabilities must be estimated. Theoretically these probabilities can be obtained from real data of system usage (calibration), but if the course is constructed from scratch, they have to be empirically estimated. The problem of estimating these conditional probabilities is that they are meaningless for the course designer. This proposal includes a way of estimating these conditional probabilities from other data that are directly requested to the course designer.

The course designer is asked to provide the following information:

- A difficulty  $d_i$  of each concept.
- A weight of each relation ( $w_{ij}^b$ ,  $w_{ij}^p$ ), that is, the strength of each relation between the two concepts  $i$  and  $j$ . In the case of *belongs-to*, this weight  $w_{ij}^b$  shows how important the sub-concept  $i$  is inside the concept  $j$ . The sum of all  $w_{ij}^b$  of all the sub-concepts  $i$  of a concept  $j$  must be 1. In the case of *prerequisite*, this weight  $w_{ij}^p$  shows how strong the prerequisite is. The course designer may say if a concepts  $i$  is a strong ( $w_{ij}^p=1$ ) or weak ( $0 \leq w_{ij}^p < 1$ ) prerequisite of another concept  $j$ . If a concept  $A$  is a strong prerequisite of  $B$ , then if the learner does not know  $A$ , then he neither knows  $B$ ; but if  $A$  is a weak prerequisite of  $B$ , and the learner does not know  $A$ , he might have some knowledge of  $B$ .

From the information provided by the course designer, an estimation of the conditional probabilities of each layer network is done using an empirical formula.

In the case of the *prerequisite network*, the conditional probabilities of having a certain knowledge level  $y$  for a concept  $C_i$ , given the knowledge level  $x_1 \dots x_k$  of its prerequisite concepts  $C_1 \dots C_k$  is obtained by the formula (1):

$$P(C_i = y / C_1 = x_1, \dots, C_k = x_k) = \begin{cases} \frac{W}{\min(x_1, \dots, x_k)} + \frac{1 - W}{N} & \text{if } y \leq \min(x_1, \dots, x_k) \\ \frac{1 - W}{N} & \text{if } y > \min(x_1, \dots, x_k) \end{cases} \quad (1)$$

where:

- $W = \max(w_{1i}^p \dots w_{ki}^p)$  is the maximum of all the weights of the relations between  $C_1 \dots C_k$ , and  $C_i$
- $N$  is the number of all the possible values a concept can have.

*Example:* Let  $C$  be a concept with two prerequisites  $C_1, C_2$ ;  $w_{1i}^p=0.8$  and  $w_{2i}^p=0.5$  the weights of the relations, so  $W=\max(0.5,0.8)=0.8$ ; and  $N=3$  (1=low, 2=medium, 3=high), then the probability distribution is:

$C_1, C_2$	$P(C=\text{low}/C_1, C_2)$	$P(C=\text{medium}/C_1, C_2)$	$P(C=\text{high}/C_1, C_2)$
$C_1=\text{low}, C_2=\text{low}$	0.86	0.07	0.07
$C_1=\text{low}, C_2=\text{medium}$	0.86	0.07	0.07
$C_1=\text{low}, C_2=\text{high}$	0.86	0.07	0.07
$C_1=\text{medium}, C_2=\text{low}$	0.86	0.07	0.07
$C_1=\text{medium}, C_2=\text{medium}$	0.46	0.46	0.07
$C_1=\text{medium}, C_2=\text{high}$	0.46	0.46	0.07
$C_1=\text{high}, C_2=\text{low}$	0.86	0.07	0.07
$C_1=\text{high}, C_2=\text{medium}$	0.46	0.46	0.07
$C_1=\text{high}, C_2=\text{high}$	0.33	0.33	0.33

This formula is based on the assumption that there is a low probability to know a concept better than its prerequisites. If a learner knows a prerequisite with level  $x$ , the probability to know the concept with level less than  $x$  will be high and homogeneous for all the levels but the probability to know it with a level higher than  $x$  will be low.

In the case of the *granularity network*, the conditional probabilities of having a certain knowledge level  $y$  for a concept  $C_i$ , given the knowledge level  $x_1 \dots x_k$  of its sub-concepts  $C_1 \dots C_k$  is obtained by the formula (2):

$$P(C_i = y / C_1 = x_1, \dots, C_k = x_k) = \begin{cases} 1 & \text{if } y = \text{round}(w_{1i}^b x_1 + \dots + w_{ki}^b x_k) \\ 0 & \text{in other case} \end{cases} \quad (2)$$

where:

- $w_{1i}^b \dots w_{ki}^b$  are all the weights of the relations between  $C_i$  and  $C_1 \dots C_k$ , respectively
- $N$  is the number of all the possible values a concept can have.
- $\text{round}(x)$  is the function takes the integer value of  $x$ .

*Example:* Let  $C$  be a concept with two sub-concepts  $C_1, C_2$ ;  $w^b_1=0.8$  and  $w^b_2=0.2$  the weights of the relations ( $C_1$  is a very important sub-concept of  $C$ , but  $C_2$  is less important); and  $N=3$  (1=low, 2=medium, 3=high), then the probability distribution is:

$C_1, C_2$	$P(C=low/ C_1, C_2)$	$P(C=medium/ C_1, C_2)$	$P(C=high/ C_1, C_2)$
$C_1=low, C_2=low$	1	0	0
$C_1=low, C_2=medium$	1	0	0
$C_1=low, C_2=high$	1	0	0
$C_1=medium, C_2=low$	1	0	0
$C_1=medium, C_2=medium$	0	1	0
$C_1=medium, C_2=high$	0	1	0
$C_1=high, C_2=low$	0	1	0
$C_1=high, C_2=medium$	0	1	0
$C_1=high, C_2=high$	0	0	1

This empirical function has been obtained following the reasoning that the knowledge level of a concept depends of the weighted knowledge of its sub-concepts.

## References

1. A.Kavcic: The Role of User Models in Adaptive Hypermedia Systems, *Proceedings of the 10th Mediterranean Electrotechnical Conference MELeCon 2000*, Lemosos, Cyprus (2000).
2. D.N.Chin: KNOME: Modeling what the user knows in UC, *User Models in Dialog Systems* (1989) 74-107.
3. J.Kay: Stereotypes, Student Models and Scrutability, *Proceedings of the 5th International Conference on Intelligent Tutoring Systems ITS2000*, Montreal, Canada (2000) 19-30.
4. P.De Bra, J.P.Ruiter: AHA! Adaptive Hypermedia for a All, *Proceedings of the World Conference of the WWW and Internet WebNet2001*, Orlando, Florida, USA (2001) 207-213.
5. M.Grigoriadou, K.Papanikolaou, H.Kornilakis, G.Magoulas: INSPIRE: An Intelligent System for Personalized Instruction in a Remote Environment, *Proceedings of Third workshop on Adaptive Hypertext and Hypermedia UM2001*, Sonthofen, Germany (2001) 13-24.
6. N.Henze, W.Nejdl: Student Modeling for the KBS Hyperbook System using Bayesian Networks, Technical Report, University of Hannover, available online at <http://www.kbs.uni-hannover.de/paper/99/adaptivity.html> (1999).
7. G.Weber, P.Brusilovsky: ELM-ART: An adaptive Versatile System for Web-based Instruction, *International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems*, (2001) 351-384.
8. M.Trella, R.Conejo, D.Bueno, E.Guzman: An autonomous component architecture to develop WWW-ITS, *Workshop Adaptive Systems for Web-Based Education of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems AH2002*, Malaga, Spain (2002) 69-80.
9. C.Carmona, D.Bueno, E.Guzman, R.Conejo: SIGUE: Making Web Courses Adaptive, *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems AH2002*, Malaga, Spain (2002) 376-379.
10. E.Guzman, R.Conejo: An adaptive assessment tool integrable into Internet based learning systems, *International Conference on ICT in Education*, Badajoz, Spain (2002) 139-143.
11. E.Millán: Sistema Bayesiano para modelado del alumno. Phd Thesis (2000).