Alejandro Peña-Ayala
Editor

Intelligent and Adaptive Educational-Learning Systems

Achievements and Trends

SMART INNOVATION, SYSTEMS AND TECHNOLOGIES 17

Springer

Smart Innovation, Systems and Technologies 17

For further volumes:
http://www.springer.com/series/8767

Alejandro Peña-Ayala (Ed.)

Intelligent and Adaptive Educational-Learning Systems

Achievements and Trends

Springer

Editor
Alejandro Peña-Ayala
WOLNM
Leyes Reforma
Mexico

ISSN 2190-3018 e-ISSN 2190-3026
ISBN 978-3-642-30170-4 e-ISBN 978-3-642-30171-1
DOI 10.1007/978-3-642-30171-1
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012937668

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Educational learning systems (ELS) represent computer-based approaches devoted to spread educational services for teaching and learning mainly through the Internet. When the development of ELS takes into account artificial intelligence techniques (e.g., acquiring and representing knowledge, make inferences and automatic learning) they become intelligent. ELS are adaptive, once they pursue to adapt themselves to satisfy users' needs, such as: navigation, interaction, content authoring and delivering, sequencing, assessment, evaluation, assistance, supervision and collaboration. Hence, ELS that include some kind of intelligent and adaptive functionality are called: intelligent and adaptive ELS (IALES).

This book reveals a sample of current work in the IALES, where researchers and practitioners of fields such as pedagogy, education, computer sciences, artificial intelligence, and graphic design join efforts to outcome frameworks, models, methods, systems and approaches for innovate the provision of education and enhance the learning of students. According to the nature of the contributions accepted for this volume, four kinds of topics are presented as follows:

- Modeling: An essential component of any IAELS is the user model. It depicts relevant cognitive and personality traits of the student, the assessment of her/his performance, the acquired domain knowledge and other useful attributes in order the IALES to behave adaptive to tailor user's learning needs.
- Content: Content represents the raw material and the main source of stimuli for students in order they to acquire knowledge, develop skills and gain experiences to accomplish some level of competence in a given educational domain.
- Virtuallity: Modern user-system interfaces and technologies engage students to work in virtual environments that catch their senses and challenge their cognitive faculties in such a way they represent a new educational paradigm.
- Applications: Several sorts of approaches compose the scope of IAELS such as: metacognition, educational system architectures, collaborative learning, educational data mining and case studies.

This volume is the result of one year of effort, where more than forty chapters were rigorous peer reviewed by a set of ninety reviewers. After several cycles of chapter submission, revision and tuning based on the KES International quality principles, twenty works were approved, edited as chapters and organized according to the prior topics. So the first part corresponds to *modeling* and includes chapters 1 to 5; the second part represents *content* and embraces chapters 6 to 10; the third part concerns to *virtuallity* and holds chapters 11 to 14; the fourth part is related to *applications* and contains chapters 15 to 20. A profile of the chapters is given next:

1. Chapter one introduces an affective behavior model to point out a student's affect state by means of a dynamic Bayesian network and a cognitive model of emotions.
2. Chapter two presents an adaptive learning environment model composed by four models (e.g., domain, learner, course structuring, adaptation) in order to set adaptive learning curriculum.
3. Chapter three proposes a proactive sequencing based on a fuzzy-causal student model to estimate learning outcomes that different content about a given concept of the domain knowledge produce on the apprenticeship of the student for choosing the most profitable option.
4. Chapter four aims at applying mining process to learner models for finding out rules from event logs. The approach combines learning styles with process mining procedures.
5. Chapter five aims at using a learning style index to find out effective ways to learn. Moreover, the work advices tutor to adopt suitable content for efficient teaching.
6. Chapter six reports the experience gained with the use of the GRAPPLE, an environment that holds a common user model framework, where structured content is authored and adaptation is set as guidance and personalized material.
7. Chapter seven aims at adaptive content selection by means of an adaptation model, which uses a decision-based approach to adaptively choose learning objects in educational hypermedia systems.
8. Chapter eight outlines a collaborative adaptive learning tool, which is able to produce several instances of a learning object by the parameterization of some features through metadata.
9. Chapter nine shares a case study about the use of an adaptive learning management system and authoring tool to support the design of adaptive and reusable courses.
10. Chapter ten pursues the reuse of intelligent tutoring systems; thereby it implements them as if they were learning objects by means of the Sharable Content Object Reference Model.
11. Chapter eleven details how three-dimensional virtual worlds are suitable environments to be collaboratively used by a group of peers aimed to accomplish a common goal, such as writing.
12. Chapter twelve engages students to develop skills and gain knowledge within a smart home domain, which is intended to anticipate and meet inhabitant's needs as they adapt to changing goals and preferences.

13. Chapter thirteen relates a naval training experience, where conscripts were trained by means of a closed-loop adaptive training system that delivers tactical air controllers instruction and provides additional practice lectures.
14. Chapter fourteen implements a cognitive tutoring agent that holds episodic, emotional, procedural and causal learning capabilities, which are used during its interactions with users to enhance the support it provides.
15. Chapter fifteen outlines an approach to adapt agent prompts as scaffolding of reflection at two levels, generic and specific, that is implemented to support students' learning-by-teaching activities.
16. The chapter sixteen aims at triggering self-regulation to encourage users of an educational learning system to acquire higher order knowledge by means of using a dynamic modeling environment.
17. Chapter seventeen outlines a seamless Web-mediated training courseware design model that encourages novice courseware authors to deliver their own adaptive educational-learning systems.
18. Chapter eighteen examines whether the provision of illusionary sense of control, implicit in collaborative learning, is perceived as current control and cause intrinsic motivation towards better work.
19. Chapter nineteen points out an intelligent system for modeling and supporting academic educational processes, which aims at evaluating and refining university curricula in terms of best possible accumulative grade point average.
20. Chapter twenty evaluates three areas of the e-learning process (e.g., technological, business, educational) and presents a case study about how motivation is a key component to encourage students to get complete e-learning courses.

I wish to express my great attitude to all authors, all reviewers, the Springer editorial team, and the editors Prof. Thomas Ditzinger and Prof. Lakhmi C. Jain for their respective collaboration to accomplish this work.

Moreover, I acknowledge the support provided by the National Council of Science and Technology (CONACYT) and the National Polytechnic Institute (IPN) of Mexico by means of the grants: CONACYT 118862, CONACYT-SNI-36453, CONACYT 118962-162727, IPN-SeAca/COTEPABE/144/11, IPN-COFAA-SIBE, IPN-SIP-20120266, IPN-SIP-EDI: SIP/DI/DOPI/EDI-0505/11.

The last but not least, I appreciate the strength given by my Father, Brother Jesus and Helper, as part of the research projects of World Outreach Light to the Nations Ministries (WOLNM).

March 2012 Alejandro Peña-Ayala

Contents

Part I: Modeling

Part II: Content

Part I
Modeling

Chapter 1
Affective Modeling for an Intelligent Educational Environment

Yasmín Hernández[1], L. Enrique Sucar[2], and Gustavo Arroyo-Figueroa[1]

[1] Instituto de Investigaciones Eléctricas, Reforma 113, Cuernavaca, México
{myhp,garroyo}@iie.org.mx
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro 1, Tonantzintla, México
esucar@inaoep.mx

Abstract. Emotions have a ubiquitous role in education and play a key role in learning and motivation. A motivated student learns in a better way than an indifferent student. There is evidence that tutors look at and react to the emotional state of students to motivate them and improve their learning. As regards computers, they have made a contribution in education. There are programs to teach almost any subject matter, but the real challenge consists in providing personalized support to human learning in view of previous knowledge and affective states to achieve an adaptive and intelligent educational-learning system. We have developed an affective behavior model that considers the affect and the knowledge state to provide students with an adaptive and intelligent instruction. The affective behavior model has been integrated into an environment to learn robotics. The instruction is presented by an animated intelligent agent. The affective behavior model maintains an intelligent representation of the student's affect state to adapt the instruction by means of a dynamic Bayesian network (DBN). The affect diagnosis is based on the Cognitive Model of Emotions (CME) and on the five-factor model of personality. The model was evaluated and the results show a high precision in the affective student model and on students learning. We present the model to endow educational environments with affective behavior wherein students' affect is reflected on the user-system interactions. Our affective student model sets an intelligent representation of the student. We present results from the model evaluations.

1.1 Introduction

Emotions have been recognized as an important component in motivation and learning. There is evidence that experienced human tutors monitor and react to the emotional state of the students in order to motivate them and to improve their learning process (Johnson et al. 2000, Qu et al. 2005).

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 3–24.
springerlink.com

Recently, there has been extensive work on modeling student emotions in intelligent tutoring systems; an example of this kind of research can be found in (Conati and Mclaren 2009). However, there have been only limited attempts to integrate information on student affect into the tutorial decisions (Zacharov et al. 2008, Faivre et al. 2003, Murray and VanLehn 2000).

If we want to consider the student's affective state in the tutorial actions, an important problem is to identify the best tutorial action, given both the students' knowledge and affective state. In this chapter, we describe an approach to tackle this problem. We have developed an affective behavior model (ABM) that considers both the knowledge and the affective state of the student to provide students with adaptive and intelligent instruction. We have designed the ABM based on interviews with qualified teachers with the purpose of understanding the reason teachers carry out their actions according to the state of affect and the knowledge of the students. This work is one of the first attempts to build an affective tutor, in particular, based on an extensive study with teachers. In the literature there are very few studies reported with as many teachers participating (Alexander et al. 2005).

The affective behavior model maintains an affective student model by means of a DBN, and it is used to adapt the instruction. The affect prediction is based on contextual information as proposed by the well-known CME (Ortony et al. 1988). The affective student model also takes into account the theory stated by the Five-Factor model of personality (Costa and McCrae 1992).

Although sometimes emotion and mood are used interchangeably, we are making a distinction between them. We consider *mood* as representing an emotional state with longer duration time, whereas we consider *emotion* as a state with shorter time duration. These two states have impact on one another and interact in several ways. In this work, we include only emotions but we are planning on including mood in our student model. Herein, we will use affective state to mean emotional state. We decided to use *affect* instead of *emotion* as stated by (Picard 2000), who affirms that *emotion* has a negative connotation, whereas *affect* does not.

For testing the affective student model, the ABM is being integrated into an environment to learn mobile robotics (Noguez and Sucar 2005); the results are encouraging since they show a high precision of the affective student model. In this chapter, we present the affective student model.

1.2 Trends and Related Work

In the complex task of endowing computers with affective behavior there are several issues and proposals. Some approaches focus on providing computer programs with moods, temperaments, etc., while other proposals try to understand the users' affective state and react accordingly. All of these proposals try to make an adaptive and convincing user-computer interaction. In the educational field the final aim is endowing educational programs with emotional abilities to help students learn.

In order to understand the students' affect there are several proposals. Some proposals are based on corporal and biological signals, such as skin conductivity, blood volume pressure, muscle tension activity and other proposals are based on facial expression. In the literature, we can find works with the latter approach, for example, in (Abbasi et al. 2007) a relationship between facial expressions and affective states is established. They conducted a study videotaping students and asking them about their affective state.

In (Dragon et al. 2008) a proposal is described which includes technology to collect information about emotional states with real-time and multimodal sensors. They use a pressure mouse to detect the increase of pressure related to the increase of levels of frustration. A posture analysis seat which works with pattern recognition algorithms to identify interest and boredom is also used. In addition, they detect the skin conductivity by means of a sensor in a kind of glove; the detected signal is related to attention-getting events. Finally, a facial expression camera is integrated. This proposal looks to integrate emotion detection within an intelligent tutor as part of learning in a natural classroom setting.

A wearable camera system is presented in (Teeters et al. 2006). The camera analyses, in real-time, the facial expressions and head gestures of its wearer and infers six affective-cognitive states. These are: agreeing, disagreeing, interested, confused, concentrating and thinking.

Conversely, other approaches based their investigations on theoretical models of emotions with strong support in psychology. These models establish emotions given certain circumstances. For example, (Kort et al. 2001) proposes a pedagogical model of emotions. They state: "Emotions and learning are closely related, and that through the learning process, the students cross over several steps related to emotion dimensions". Another theoretical model is the CME (Ortony et al. 1988) which states that emotions emerge as a matching process between goals, principles and preferences with the current situation.

The CME was used in the design of the animated agent PAT, pedagogical and affective tutor (Jaques and Viccari 2005). PAT interacts with students by means of emotional behavior. The agent recognizes students' affective state given their actions and tries to motivate them with facial, corporal and textual communication.

However, this task is extremely difficult, and therefore there are many investigations attempting to explain the relationship between learning and affect. For example, in (D´Mello et al. 2008) a survey is presented comparing students' self-reports and teachers' judgments during several tutoring sessions, trying to establish a relationship between situation and affective states. In (Lehman et al. 2008) the relationship between affective state and tutor's actions is investigated. There is also some research trying to endow the tutor with personality, such as in (Kim et al. 2007) wherein the impact of different agents' personalities on students is analyzed.

Despite the importance of emotions in learning has been stated long time ago, the affective computing and particularly its application to learning environment is recent. In view of the related works, we can see how many issues are immersed in the affect processing and how much has to be done in order to have a model to respond with a suitable action and at the appropriate pedagogical time.

The affect research focuses on relationships between emotion, cognition and learning. The current research involve physical sensors o theoretical models to observe the emotions that occur during learning, investigating relationships between emotions and learning gains, modeling the temporal dynamics of the emotions, identifying cognitive, bodily and linguistic indicators of emotional expressions.

We are interested in developing a comprehensive model to detect students' emotions and to act accordingly; but a difference was reported by (Dragon et al. 2008) and (Teeters et al. 2006), who are interested in physical signs of emotions, our first step is to understand the cognitive basis of emotions and its relationship with learning as reported by (Kort et al. 2001) and (Ortony et al. 1988).

Our work reacts before students' emotions more than to show emotions as is investigated by (Jaques and Viccari 2005) and (Kim et al. 2007). We present our proposal to model affective behavior in learning environments in the next section.

1.3 Modeling Affective Behavior

Traditionally, an intelligent educational system decides what and how to teach based on a representation of the student's knowledge. However, there is evidence that experienced human tutors manage the affective state of students to motivate them and improve their learning process (Johnson et al. 2000, Qu et al. 2005). Thus, the student representation structure needs to be augmented to include knowledge about the affective state. An affective model which makes decisions with base on the students' affective state is also needed. In that way, the students can be provided with a tutorial action which fulfills knowledge requirements, and at the same time is appropriate with the student's affective state. Fig. 1.1 shows the general architecture of an intelligent educational system with affective modeling.

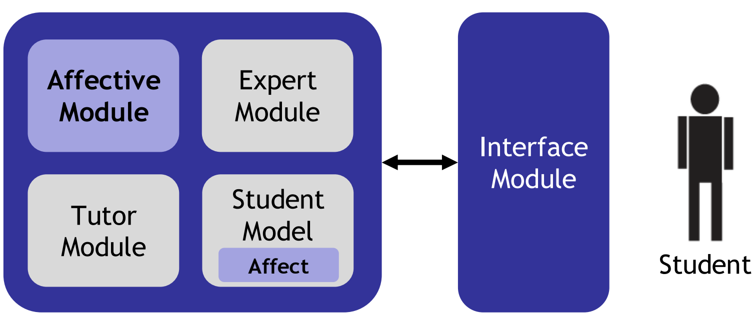

Fig. 1.1 Architecture of an intelligent educational system with affective modeling. The affective model extends the basic architecture of an intelligent educational system as it integrates information about the students' affect and includes an affective module to reason with this affective state, and in this way provides students with an affective and pedagogically suitable response

This architecture includes information about the student's affect in the student model; and it also includes an affective module. This new module contains knowledge to permit reasoning with the student affect. This architecture is based on the one proposed for intelligent tutoring systems (Burns and Capps 1988).

In the context of this work, the process of endowing educational systems with affective behavior includes two aspects: 1) understanding the affective state of the student; 2) deciding the tutorial action to be presented to the student in view of the student's affective and knowledge state. With these two aspects as the goal, we have developed an ABM for intelligent educational systems. The ABM is composed of two main components: the *affective student model* and the *affective tutor model*. A flow diagram of the ABM is presented in Fig. 1.2.

The ABM is set to enable intelligent educational systems to include affective responses in their pedagogical actions. The ABM relies on three elements for selecting the tutorial action to be presented to students: a model of the student's current knowledge (*pedagogical student model* in Fig. 1.2), a model of the student's current affect (*affective student model* in Fig. 1.2), and the tutorial situation.

The tutor module receives these three elements and produces an affective action and a pedagogical action rooted in pedagogical and affective models. The pedagogical action supports the students' learning and the affective action boosts students' morale in the current situation. The two actions are then integrated into the actual tutorial action delivered to the student through the interface module.

The affective action helps the pedagogical model to establish the next pedagogical action, and it also helps the interface module to establish the physical presentation of the pedagogical action. The decision of selecting the affective action first and using it to guide the selection of the pedagogical actions is based on feedback from the teachers in our investigations. Twenty teachers participated in our studies; they stated that they first observe the affect and motivation of students and then subsequently decide on the pedagogical strategies (Hernández et al. 2009, Hernández et al. 2009b).

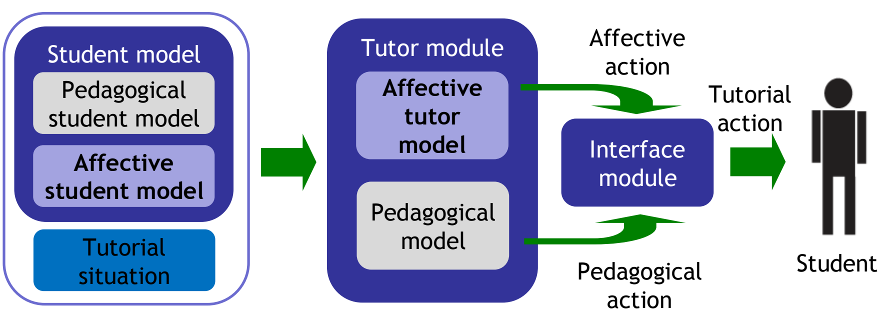

Fig. 1.2 General diagram for the affective behavior model. The model is composed of an affective student model and an affective tutor model. The tutor model produces an affective action, considering the affective and pedagogical student models and the tutorial situation. The affective action is a component of the tutorial action to be presented to the student

The ABM allows intelligent educational systems to make a mapping from the student's affective and pedagogical states to tutorial actions by means of the student model. In the next section, the affective student modeling is discussed.

1.4 The Affective Student Model

There are several proposals to predict or diagnose the individual's affect. These include facial expressions, and even, direct inquiry to the students as to their affective state. However, the latter is not a reliable means to ascertain affect, as asking them disrupts their concentration. People tend to be affable and to give a favorable answer even when the questioner is a computer (Reeves and Nass 1996).

Our affective student model uses the CME (Ortony et al. 1988) to provide a causal assessment of student's emotions based on contextual information. CME defines emotion as the end result of a cognitive appraisal of the current situation with respect to one's goals, principles and preferences. In this way, emotions represent a positive or a negative reaction, with respect to consequences of events, actions of agents and aspects of objects. Thus, an individual's emotions are related to the elements in the current situation: events, objects and agents, including him. Fig. 1.3 aims to show the fundamentals of the CME.

CME proposes 22 emotions and the emotions are classified according to the causes which elicit them: the consequences of events, the actions of agents and the aspects of objects. The elicited emotion also depends on the relevance of the event, agent or object to the individual; therefore, the model establishes parameters which represent the intensity of emotion.

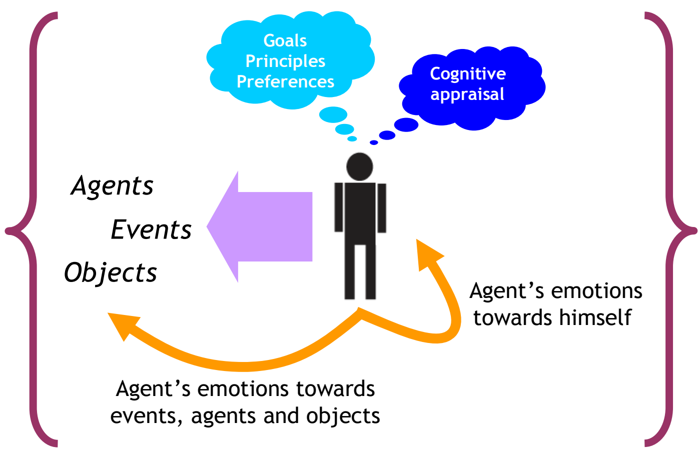

Fig. 1.3 Cognitive Model of Emotion basic diagram. The CME defines emotion as an end result of a cognitive appraisal of the current situation with respect to one's goals, principles and preferences. Emotions are elicited by elements included in the actual situation; they can be events, agents and objects

With regard to the consequences of events, in a tutorial session there are events that are pertinent to learning, such as an explanation (tutor's event) or the completion of an exercise (student's event). These events produce results that have an influence on the well-being of the student; therefore, these results trigger in the students the states we want to evaluate, such as *joy* and *distress*.

With reference to actions of agents, the tutorial situation contains two agents that are relevant for learning: the student and the tutor. These agents fulfill actions and the results of these actions cause emotions in the students, such as *pride* or *shame* if the student carried out the action; or *admiration* or *reproach* if the tutor performed the action. Thus, the results are attributable to the agent who carried out the action and consequently the student's emotions are focused on that agent.

We do not include emotions which emerge as a reaction to the aspect of objects, such as *love* and *hate*. Thus, from the set of emotions proposed by CME, the affective student model includes six emotions: *joy*, *distress*, *pride*, *shame, admiration* and *reproach*. The emotions *joy* and *distress* are reactions by the individual to an event in the tutorial session. The emotions *pride* and *shame* emerge as a consequence of the student's action. The emotions *admiration* and *reproach* emerge as a consequence of the tutor's action. Fig. 1.4 depicts how these emotions emerge in our model consistent with CME. The agent (student or tutor) performs an action and the student observes the result; he compares the results with his goal, causing emotions in keeping with the fulfillment of the student's goal.

Based on our affective student model on a comparison between the current situation and the individual's expectations, we make a prediction about the affective state. In that way, we do not need physical indicators such as facial expression, blood pressure, etc., or evidence of the individual's behavior for the affective state. Nevertheless, having additional indicators allows disambiguation of certain states; an approach to a student model with several indicators is given in (Conati and Maclaren 2009).

According to CME, goals are essential to determine the affective state. As in the case of understanding the student's affect, we believe goals cannot be explicitly asked of the student during the interaction; because in order the student to provide a reliable answer, he would need to understand the question and be introspective, and errors can occur. Consequently, the goals in our model are inferred from indirect sources of evidence; we use personality traits and student's knowledge as a predictor of the student's goals. We based the personality traits on the Five-Factor Model (Costa and McCrae 1992, Boeree 1998), which considers five dimensions of personality: *openness, conscientiousness, extraversion, agreeableness and neuroticism.* The Five-Factor Model describes each of these dimensions of personality and establishes their characteristics of behavior. For example, a person who has a high score in the *openness* dimension is a person willing to experience new things, is always disposed to dialogue, and has a high capacity for invention. Whereas, if he has a low score in *openness*; then he is a person with little disposition toward new experiences.

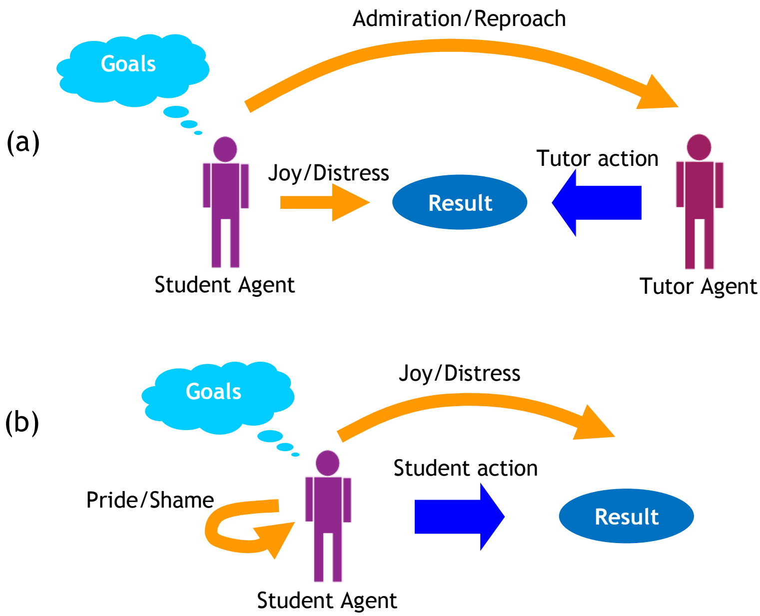

Fig. 1.4 Emotions represented in the affective student model. (a) Emotions from the student agent toward the tutorial session (joy/distress) and toward the tutor agent (admiration/reproach). These emotions surface when the student sees the result of the actions of the tutor. (b) Emotions of the student toward the tutorial session (joy/distress) and toward himself (pride/shame). These emotions are generated when the student observes the results of his actions. In accordance with the CME, the student compares the current situation against his goals

In a survey devoted to understand the relationship between personality and learning (Heinström 2000), it is stated that personality dimensions in terms of learning styles are reflected in learning strategies, and finally the personality produces the results in learning. This study also pointed out that the characteristics of personality act as guides for motivation and for learning strategies. The conclusions of the above study establish that the prominent dimensions for learning are *neuroticism* and *conscientiousness*. It also sets a relationship of learning with *openness*; but this relationship has not as yet been fully proven (Heinström 2000).

According to (Heinström 2000), the most important relationship of learning is with the *conscientiousness* dimension, since this personality dimension is related to discipline in one's work. Interest in the subject matter, concentration and the concept of study is easy. Students with this personality have intrinsic motivation and a positive attitude toward study. The *neuroticism* dimension is related to the lack of concentration, the fear of failure and the experience of studying as stressful. The neuroticism dimension is coupled with the lack of critical ability and difficulty in understanding the relationship between things. Students with this personality concentrate on memorizing, without interest in understanding and finding meaning in the material. In these cases the motivation of the students is extrinsic.

The *openness* dimension is related to questioning and analyzing arguments, critical evaluation, searches in the literature and building relationships. The students with this personality are analytical, logical, and they relate what they learn to their prior knowledge; their motivation is intrinsic and they seek personal and independent comprehension.

As a complementary way to understand students' goals, we use the student's current knowledge about the subject matter. We think knowledge has an effect on the absence or the presence of certain goals, especially on those goals related to learning, which is our main concern in this context.

As we mentioned, CME states that emotions emerge as a consequence of a cognitive appraisal of the current situation and goals; in our context we ascertain the current situation from the tutorial situation, i.e., the results of student actions (exercises, tests, etc.).

In that way, we can make a prediction about the affective state of the student based on contextual information; i.e., the current state of the student's knowledge, his personality traits and the tutorial situation. In the next section, we present the structure of the affective student model and we describe how it is built.

1.5 Building the Affective Student Model

As the process of establishing the affective state of students involves uncertainty, we rely on DBN for that task, due to their strong mechanisms for managing uncertainty. We use a DBN that probabilistically relates the student's personality, goals and interaction events with the student's affective states, based on the theory defined by CME. The dynamic network allows for the modeling of the changing nature of the affective state and representing the impact of the previous state in the current affective state. In our model, reaching the goals is the main factor influencing the affective state which in turn is influenced by the tutorial situation (the results of student's actions) and the student's goals. Consequently, the goals change during the tutorial session as the student learns. Fig. 1.5 shows a high level representation of the affective student model.

The affective state is not static but it changes over time as a result of the changing environment and the particular interpretation of the situation of each individual. The DBN models this dynamic nature of the affective state and its influence on the next state. In our model, the affective state changes after the student carries out an action. The dynamic network includes two time slots at any given time. A time slot is added and a time slot is discarded after each student's action. To infer the affective state at t_n we use the knowledge state of the student, the tutorial situation and the personality traits of the student; this is used to predict the affect at t_{n+1}.

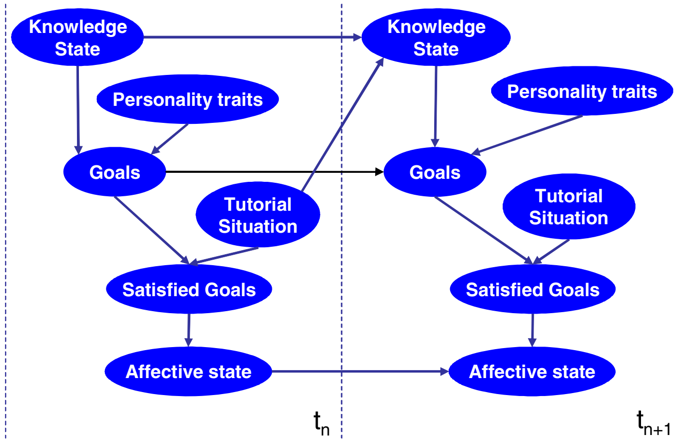

Fig. 1.5 High level DBN for the affective student model. We include two time slots to represent the dynamic behavior of affect and its impact in the next state

The student's appraisal of the current situation given his goal is represented by the relationship between the *goals* and the *tutorial situation* nodes via the *satisfied goals* node. The influence of the appraisal process on the student's affect is represented by the link between the *satisfied goals* node and the *affective state* node. This is our approach to the implementation of CME.

The goals change when the student interacts with the learning environment, that is, when the student is acquiring new knowledge. Another indicator for the goals is personality; however, this component does not change during the tutorial session. The influencing factors of goals are represented by knowledge state and personality traits nodes.

We call the network in Fig. 1.5 a high level representation of the model because of each node in that network is actually a set of nodes in the detailed model. Further, we describe comprehensively the DBN through a couple of subjects: how we build the affective student model and how we obtain the values of the nodes.

Fig. 1.6 shows the detailed DBN in a test case for robotics in which the students learn by carrying out experiments, such as setting up and guiding a mobile robot. One specific moment in time is depicted in this network. The dependency relationships in the DBN have been set based on the literature (Costa and McCrae 1992, Boeree 1998, Heinström 2000) and on insights from teachers and intuition.

The first node in the network is the knowledge node. The evidence for this node comes from the student action results (experiments, tests, etc.) by means of a pedagogical student model. The pedagogical student model is also a DBN that represents the current experiment and contains a node for each topic in the experiment. The probability of knowing each of these topics influences the probability of knowledge for the entire experiment. The knowledge node has two values: *knows* and *does not know*. This process is presented in Fig. 1.7.

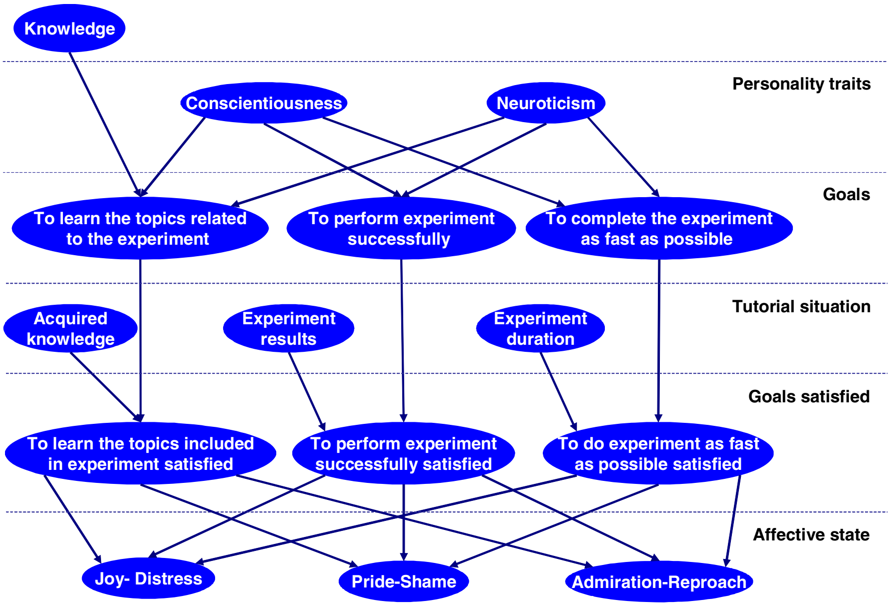

Fig. 1.6 Detailed affective student model represented by a DBN. Each set of nodes is a detailed representation of the DBN at a specific time

This DBN for the pedagogical student model is specific for a particular experiment. Each experiment in the learning environment has a different structure based on the main topic of the experiment. In this case, we show an experiment with four topics.

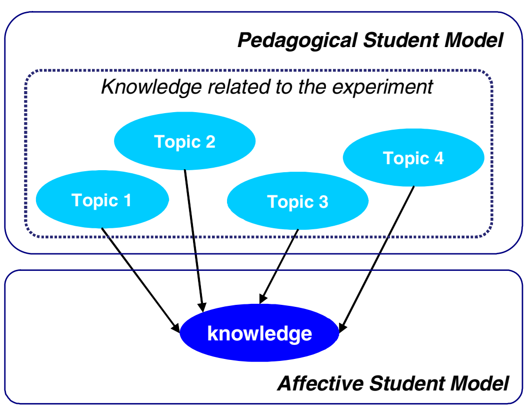

Fig. 1.7 Knowledge node. Evidence for this node is taken from the pedagogical student model, from the probability of knowing each topic in the experiment

The next set of nodes is the *personality* nodes. As previously indicated, for the personality traits of the student we use the neuroticism and conscientiousness dimensions. To obtain priors for these personality nodes, we conducted a study with 58 students. This group of students is a representative sample of the population who will use the learning environment given that they are graduate or undergraduate students, they are attending the same courses, and they are in the same age group. The complete survey can be found in (Hernández 2008). In this study, the students answered a personality test based on the Five-Factor model (Boeree 2005). The test consists of 50 adjectives, and the individuals have to rate how much the adjectives applies to them. The grade of the test indicates if the student is at a low, average or high level for each personality dimension. According to the survey, most of the students (78%) are at average level and a smaller student group (22%) is at low level for both personality dimensions. In this sample no one student is at high level for no one personality dimension. Coincidentally, both personality traits have the same percentages for the three personality levels. Based on this study, we establish the priors for the personality nodes presented in Table 1.1. In our DBN, the personality nodes have three values: high, average and low.

To have a more precise estimation of students' personality, they can answer the same personality test. To establish the dependency relationships in the DBN between personality nodes and goal nodes we considered the personality dimension description as stated by the Five-factor model (Costa and McCrae 1992, Heinström 2000, Boeree 1998). For example, if the student has a conscientious personality and limited understanding of the subject matter, the probability of having the goal *to learn the topics related to the experiment* is high, because he is a responsible person who cares about his performance. On the other hand, if the student is a neurotic person, there is a higher probability of having the goal *to perform the experiment successfully* rather than to learn, because a neurotic person wants to have immediate and tangible success.

The student's knowledge about the topics and the student's personality are accounted to infer the students' goals. We included three goals in the affective model: 1) to learn the topics related to the experiment; 2) to perform the experiment successfully; 3) to complete the experiment as fast as possible.

The reasons for establishing these goals are based on the nature of the task. That is, to perform an experiment to learn mobile robotics. The first goal can be present due to the main objective of the task: to complete an experiment for learning. The second goal can be present because of the student can wish to have success in reaching a target.

Table 1.1 Priors for *conscientiousness* and *neuroticism* personality nodes

Values	Conscientiousness	Neuroticism
1) High	0.01	0.01
2) Average	0.77	0.77
3) Low	0.22	0.22

The third goal can be present because generally students want a quick reward. In Table 1.2, we present the conditional probabilities table (CPT) for the goal *to perform experiment successfully* node. This node has two values: *present* and *absent*, and the influencing nodes, conscientiousness and neuroticism, have three values: *high*, *average* and *low*.

The CPT for the other two goals are similar to CPT in Table 1.2: the goal node has two influencing nodes, the personality nodes and its probabilities of having (present value) or not having the goal (absent value) are based on the personality traits. Additionally, the probabilities of the goal *to learn the topics related to the experiment* are based on the student's current knowledge.

The next set of nodes is the *tutorial situation* nodes (Fig. 1.6). The information for the *tutorial situation* nodes comes from the results of the student action by means of the pedagogical student model. We use the knowledge about the topics included in the experiment, and based on the specific experiment, data such as: how many times the student made a correction to the robot's track, if he reached or did not reach the target, and how long it took to reach to the target. This process is shown in Fig. 1.8.

Consequently, the student's appraisal of the current situation given his goal (CME) is represented by the relationship between the *goals* and *tutorial situation* nodes via the *satisfied goals* nodes.

Table 1.2 CPT for the to perform experiment successfully goal node

Goal 2: *to perform experiment successfully*									
Conscientiousness	High			Average			Low		
Neuroticism	High	Avg	Lw	High	Avg	Lw	High	Avg	Lw
Present	0.9	0.8	0.7	0.8	0.7	0.6	0.7	0.6	0.4
Absent	0.1	0.2	0.3	0.2	0.3	0.4	0.3	0.4	0.6

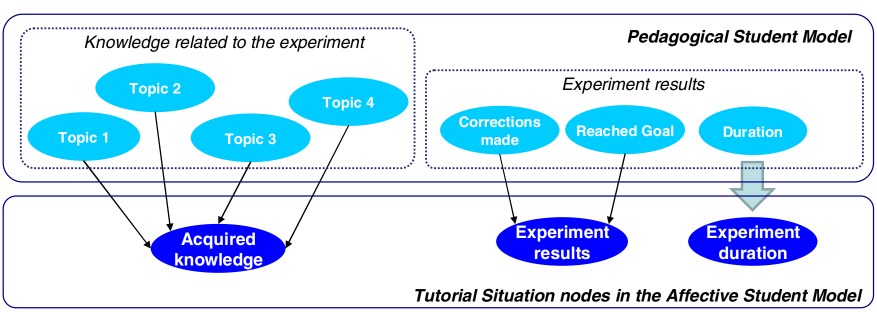

Fig. 1.8 Tutorial situation nodes. The tutorial situation nodes consider the pedagogical student model and the experiment's results

The probability that the goal has been satisfied depends on the presence or the absence of such a goal and the evidence of the results of the students' actions (tutorial nodes). However, it is more important to have the goal present in order to satisfy it. These nodes have two values: *satisfied* and *not satisfied*.

Finally, as stated by CME the emotions emerge as a comparison between goals and situation. In our model, the influence of the appraisal process on the student's affect is represented by the link between the *satisfied goals* nodes and the *affective state* nodes. We include in the model six emotions: joy, distress, pride, shame, admiration and reproach. These are represented as three pairs of mutually exclusive emotions: *joy-distress*, *pride-shame* and *admiration-reproach*. Each pair is represented by a binary node in the network. We used each pair of emotions as a dimension (see Fig. 1.9). We considered that for the same event/situation the student cannot have both emotions in the dimension; that is, the student cannot be happy and sad at the same time about the result of an experiment. It could be possible for several events; but it is not our case, we are evaluating the emotion toward one event only.

The *joy-distress* affective state node represents the emotions the student can have regarding the situation. That is: he is happy because of he learned, or due to he got the target, or because of he completed the experiment quickly. The *pride-shame* affective state node represents the emotions from the student towards himself. It means, he is proud of himself because of he learned the topics in the experiment, or due to he completed the experiment successfully, or because he achieved the goal quickly. The *admiration-reproach* affective state node represents the emotion from the student towards the tutor depicted by an animated agent (as a part of our study we included in the learning environment an animated agent to present the instruction and to be the face of the tutor). The student can feel admiration for the tutor as a result of the tutor taught him and therefore he reached his goals. In Table 1.3, we show the conditional probabilities for the *joy-distress* node.

Fig. 1.9 Emotion dimensions. The affective student model includes three pairs of mutually exclusive emotion. This consideration applies only to a same event

Table 1.3 Conditional probabilities table for the joy-distress node

Joy-distress								
Goal to learn the topics related to the experiment satisfied	Yes				No			
Goal to perform experiment successfully satisfied	Yes		No		Yes		No	
Goal to complete the experiment as fast as possible satisfied	Yes	No	Yes	No	Yes	No	Yes	No
Joy	0.9	0.8	0.7	0.8	0.7	0.6	0.7	0.6
Distress	0.1	0.2	0.3	0.2	0.3	0.4	0.3	0.4

In the following section, we describe how we evaluated the affective student model we just presented.

1.6 Evaluation of the Model

In order to evaluate the affective student model we integrate the ABM into an intelligent learning environment to learn mobile robotics (Noguez and Sucar 2005). In this environment the students learn by carrying out experiments about to set up and guide a mobile robot to reach the target. Once they completed the experiments, they learn a lesson based on their performance. The instruction is based on a probabilistic representation of the students' knowledge state. The pedagogical actions are explanations about the topics in the current experiment. This learning environment presents the instruction by means of a textual explanation without an agent or face for the tutor. However, in our affective student model we assess the student's emotion toward the tutor. Therefore, we need a face for the tutor so that when we evaluate the model we can ask for the emotion toward the tutor without causing confusion in the student. For that reason, we integrate an animated agent into the learning environment.

To include a suitable agent, we conducted a survey in which we asked nine teachers to select an animated character and appropriate animations to be integrated into the intelligent environment. Teachers were presented with the possible animations displayed by four characters of Microsoft Agent® (Microsoft 2005), so they can see the potential of the every animated character. In Fig.1.10, the characters of Microsoft Agent® presented to teachers are shown, left to right: Robbie, Genie, Peedy and Merlin.

Fig. 1.10 Characters of Microsoft Agent. Teachers could see the potential of these characters and select the one they considered suitable for the intelligent environment audience

Teachers could select a character and see all of its animations as many times as they wanted. The selected animations were included in the ABM as affective actions. The character Robbie was selected by seven teachers and the character Merlin was selected by two teachers. Even the teachers that selected Robbie acknowledged Merlin is much more expressive than Robbie; they thought Robbie was more suitable for the domain (Robotics) and for the students' ages (college students).

To evaluate the performance of the affective student model, we conducted a Wizard of Oz study with a group of 20 students. This sample is small but it is representative of the type of students who will be using the learning environment. The aim of a Wizard of Oz study is to obtain information for designing and evaluating prototypes or systems which have not yet been finished (Dow and MacIntyre 2007, Dow et al. 2005). This type of study has been used since human-computer interaction began and it has been widely used to emulate technologies in interactive systems (Dow and MacIntyre 2007, Anderson et al. 2002). It consists of employing operators or mechanisms hidden from the user temporarily to emulate unfinished components of a computer system during its development (Dow et al. 2005). In our case we did not have the Microsoft Agent® completely integrated into the intelligent environment for learning robotics. Therefore, we videotaped several tutorial scenarios, and for every scenario we showed the animation (affective action) according to the affective behavior model.

Aside from having personality priors, we requested the participating students answer the same personality test before using the system in order to have a more precise evaluation of personality. As the first point in the survey, the students answered a personality test based on the Five-Factor model (Boeree 2005). This test is the same test used to obtain the priors for the personality nodes. It is composed of 50 adjectives such as talkative, sympathetic, envious, deep, careless, relaxed, average, bold, kind and moody. The students have to rate how each adjective applies to them.

The survey consisted of, presenting to the students, three different tutorial scenarios. The tutorial scenario included the development of an experiment and a tutorial action presented by the animated character Robbie. The tutorial action was selected considering the student's affective state and the tutorial situation presented in the scenario. After presenting each tutorial scenario, the students were asked about their affective state given the tutorial situation with the purpose of comparison to the affective state established by the affective student model. In Fig. 1.11, we present a block diagram of the Wizard of Oz study.

1.7 Results

We compared the affective state reported by the students with the affective state established by the affective student model. The results are summarized in Table 1.4. We found that the model estimated the affective state correctly: for the emotion *joy-distress* in 72% of the cases, for the emotion *pride-shame* in 70% of the cases and for the emotion *admiration-reproach* in 55% of the cases. As we can see, the model reached a high precision for the emotions *joy-distress* and *pride-shame*. However for the emotion *admiration-reproach* the precision of the model is not so high.

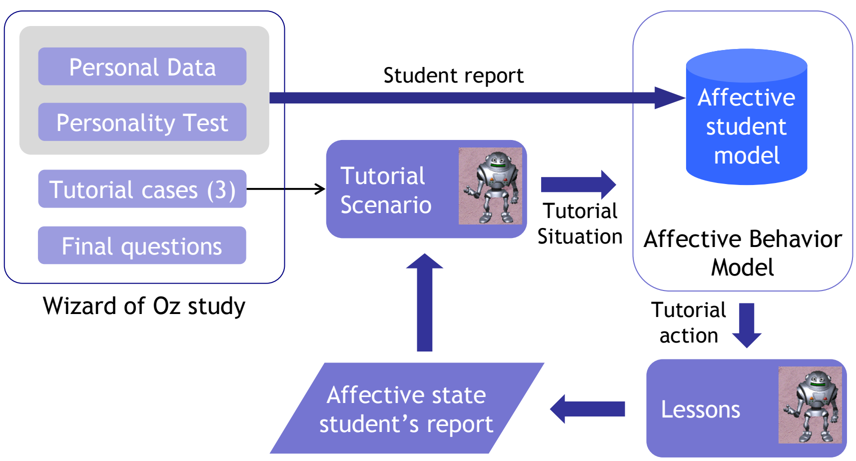

Fig. 1.11 Block diagram for Wizard of Oz study to evaluate the affective student model. The students were presented with three tutorial scenarios and they were asked about their affect after each tutorial action; their responses were compared to the affective state established by the affective student model

Table 1.4 Percentage of agreement between the affective state established by the affective student model and the affective state reported by the students

	Joy-distress	Admiration-reproach	Pride-shame
1) Agreement	43	33	42
2) Disagreement	17	27	18
Percentage of agreement	72%	55%	70%

We suppose: the emotions from students toward teachers evolve slowly. We think: students believe they are learning on their own and, in general, they do not think that teachers are instructing them well. This concurs with comments of teachers who rated the emotions from the students toward them as being mostly negative (the survey is reported in detail in Hernández 2008). It is also possible that the students did not have enough information to evaluate this emotion because of they did not receive a knowledge test. We have to conduct further investigations to validate this hypothesis and refine the affective student model and the ABM.

In this chapter, we have presented an evaluation of the affective student model, a component of the ABM. Moreover, we have tested and evaluated the complete affective behavior model in another domain. We have also integrated it into an educational game to learn number factorization (Manske and Conati 2005). We carried out a controlled user study with 82 actual students. The trial held a control group and an experimental group. The control group used the system without the ABM and the experimental group interacted with the system with the ABM. The students were given a pre-test and a post-test in order to establish learning gains by using the system. The learning gains of the groups were compared using the statistical *t*-student test. The results of the studies show positive impact on students' learning when the affective behavior model is incorporated, as shown in Table 1.5. The complete survey is presented in (Hernández et al. 2009b).

Table 1.5 Statistical analysis of the learning gains in each group, control and experimental, and between both groups

Grade	Control Group Pre-test/post-test		Exp. Group Pre-test/post-test		Learning gains Cntl grp/Exp grp	
	t	P (1-tailed)	t	P (1-tailed)	t	P (1-tailed)
6° gr.	2.55	0.09	6.95	0.000036	8.10	0.04
7° gr.	0.29	0.80	0.70	0.210000	0.36	0.69
8° gr.	0.63	0.58	0.53	0.370000	0.09	0.93
9° gr.	1.10	0.08	0.19	0.800000	0.97	0.28

1.8 Discussion

The results of our investigations about developing an intelligent representation of the student affect are encouraging as they show high agreement between the reports of the students and the results of our affective student model. Additionally, we also get positive results in the evaluation of the complete ABM due to high learning gains when we used the affective student model to instruct the student.

We decided to base our investigation on theoretical models of emotions and on indirect sources of evidence, such as personality, goals and results. This is because of we have tried to build an approach which does not interfere with the students' main task. Nevertheless, we pursue to deal with the lack of direct sources of evidence, such as biological signals, through the use of DBN. In addition, we assert: Bayesian reasoning provides strong mechanisms to work with any minimal evidence in order to manage the inherent uncertainty in the assessment of both the current relevant student affective states and the effects of the tutor's actions on them.

The ABM allows intelligent educational systems to make a mapping of a student's affective and pedagogical states to tutorial actions. Having two test domains with positive results suggest us that the ABM can be integrated into any intelligent learning environment. However, these results are not conclusive, as we need to include knowledge tests in order to prove whether our model helps students to learn. Nevertheless, the results that we have obtained so far will allow us to refine our models and to design other studies, and to progressively achieve a comprehensive approach to affective behavior.

1.9 Conclusions and Future Work

We have developed an affective behavior model for intelligent learning environments. The affective behavior model integrates an affective student model and an affective tutor model. In this chapter, we presented the first component, the affective student model, which was built considering the theory stated by the CME. We also presented the affective student model's evaluation in the mobile robotics domain via a Wizard of Oz experiment.

The results are encouraging, as they show strong agreement between the affective states given by the model with those of the students. The next step is to complete the integration of the affective student model and the complete ABM with the intelligent learning environment for mobile robotics. Subsequently, we aim to conduct a controlled user study and in this way, try to confirm our hypothesis: the learning process is improved when the affective state is considered.

Acknowledgments. We would like to thank Cristina Conati from the University of British Columbia, for many useful discussions on the definition of the affective model. We would also like to thank Amelia Hakim and Marisa Velasco Feld and the anonymous reviewers whose comments helped to improve this manuscript.

References

Abbasi, A.R., Uno, T., Dailey, M.N., Afzulpurkar, N.V.: Towards Knowledge-Based Affective Interaction: Situational Interpretation of Affect. In: Paiva, A., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 452–463. Springer, Heidelberg (2007)

Alexander, S., Hill, S., Sarrafzadeh, A.: How do human tutors adapt to affective state? In: Proceedings of Adapting the Interaction Style to Affective Factors Workshop of UM (2005)

Anderson, G., Hook, K., Mourao, D., Paiva, A., Costa, M.: Using a Wizard of Oz study to inform the design of SenToy. In: Mackay, A., Amowitz, J., Gaver, W., Verplank, B., Sutcliffe, A. (eds.) Proceedings of DIS, pp. 349–355. ACM Press, London (2002)

Boeree, G.: Big Five Minitest, http://www.ship.edu/~cgboeree/ (accessed August 15, 2005)

Boeree, G.: Teorías de la personalidad. E-book. Psychology Department of Shippensburg University (1998), http://www.ship.edu/~gboeree/personalidad.html (accessed August 15, 2005)

Burns, H.L., Capps, C.G.: Foundations of intelligent tutoring systems: An introduction. In: Polson, M.C., Richardson, J.J. (eds.) Foundations of Intelligent Tutoring Systems, pp. 1–20. Lawrence Erlbaum Associates Publishers, Hillsdale (1988)

Conati, C., Maclaren, H.: Empirically building and evaluating a probabilistic model of user affect. User Modeling and User-Adapted Interaction 19(3), 267–303 (2009)

Costa, P.T., McCrae, R.R.: Four ways five factors are basic. Personality and Individual Differences 13(1), 653–665 (1992)

Dow, S., MacIntyre, B.: New media collaboration through Wizard-of-Oz simulations. In: Proceedings of HCI and New Media Arts: Methodology and Evaluation Workshop of CHI (2007)

Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J.D., Gandy, M.: Wizard of Oz support throughout an iterative design process. IEEE Pervasive Computing: Special Issue on Rapid Prototyping 4(1), 18–26 (2005)

D'Mello, S., Taylor, R., Davidson, K., Graesser, A.: Self Versus Teacher Judgments of Learner Emotions During a Tutoring Session with AutoTutor. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 9–18. Springer, Heidelberg (2008)

Dragon, T., Arroyo, I., Woolf, B.P., Burleson, W., el Kaliouby, R., Eydgahi, H.: Viewing Student Affect and Learning through Classroom Observation and Physical Sensors. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 29–39. Springer, Heidelberg (2008)

Faivre, J., Nkambou, R., Frasson, C.: Towards empathetic agents in tutoring systems. In: Russell, I., Haller, S. (eds.) Proceedings of FLAIRS, pp. 161–165. AAAI Press, Florida (2003)

Heinström, J.: The impact of personality and approaches to learning on information behavior. Information Research: an International Electronic Journal 5(3) (2000), http://informationr.net/ir/5-3/paper78.html (accessed August 15, 2005)

Hernández, Y.: Modelo de comportamiento afectivo para sistemas tutores inteligentes. PhD dissertation, Tecnológico de Monterrey (2008)

Hernández, Y., Sucar, L.E., Arroyo, G.: Obtaining teachers' expertise to refine an affective model in an intelligent tutor for learning robotics. In: Hernández, A., Monroy, R., Reyes, C.A. (eds.) Proceedings of MICAI, pp. 122–127. IEEE Press, Los Alamitos (2009)

Hernández, Y., Sucar, L.E., Conati, C.: Incorporating an affective behavior model into an educational game. In: Lane, C.H., Guesgen, H.W. (eds.) Proceedings of FLAIRS, pp. 448–453. AAAI Press, Florida (2009b)

Jaques, P.A., Viccari, R.M.: Considering student's emotions in computer-mediated learning environments. In: Zongmin, M. (ed.) Web-Based Intelligent E-Learning Systems: Technologies and Applications, pp. 122–138. Information Science Publishing, Hershey (2005)

Johnson, W.L., Rickel, J.W., Lester, J.C.: Animated pedagogical agents: Face-to-face interaction in interactive learning environments. International Journal of Artificial Intelligence in Education 11(1), 47–78 (2000)

Kim, Y., Xu, B., Wei, Q.: Virtual peers scaffold motivation to learn. In: Proceedings of Modeling and Scaffolding Affective Experiences to Impact Learning Workshop of AIED (2007)

Kort, B., Reilly, R., Picard, R.W.: An affective model of interplay between emotions and learning: Reengineering educational pedagogy – Building a learning companion. In: Okamoto, T., Hartley, R., Kinshuk, Klus, J. (eds.) Proceedings of ICALT, pp. 43–46. IEEE Press, Madison (2001)

Lehman, B., Matthews, M., D'Mello, S., Person, N.: All alone with your emotions: an analysis of student emotions during effortful problem solving activities. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.), Proceedings of ITS. LNCS, vol. 5091, pp. 9-18. Heidelberg: Springer (2008)

Manske, M., Conati, C.: Modelling learning in educational games. In: Looi, C., McCalla, G.I., Bredeweg, B., Breuker, J. (eds.) Proceedings of AIED, pp. 411–418. IOS Press, The Netherlands (2005)

Microsoft Corporation. Microsoft Agent (2005), `http://www.microsoft.com/msagent/default.asp` (accessed November 1, 2005)

Murray, R.C., VanLehn, K.: DT Tutor: A Decision-Theoretic, Dynamic Approach for Optimal Selection of Tutorial Actions. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 153–162. Springer, Heidelberg (2000)

Noguez, J., Sucar, L.E.: A Semi-open Learning Environment for Virtual Laboratories. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 1185–1194. Springer, Heidelberg (2005)

Ortony, A., Clore, G.L., Collins, A.: The cognitive structure of emotions. Cambridge University Press, New York (1988)

Picard, R.W.: Affective computing. MIT Press, Cambridge (2000)

Qu, L., Wang, N., Johnson, W.L.: Using Learner Focus of Attention to Detect Learner Motivation Factors. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 70–73. Springer, Heidelberg (2005)

Reeves, B., Nass, C.: The media equation: How people treat computers, television and new media like real people and places. Cambridge University Press, New York (1996)

Teeters, A., El Kaliouby, R., Picard, R.W.: Self-Cam: Feedback from what would be your social partner. In: Proceedings of SIGGRAPH, p. 138. ACM Press, Boston (2006)

Zakharov, K., Mitrovic, A., Johnston, L.: Towards Emotionally-Intelligent Pedagogical Agents. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 19–28. Springer, Heidelberg (2008)

Abbreviations

ABM	Affective Behavior Model
AVG	Average
CME	Cognitive Model of Emotions
CPT	Conditional Probabilities Table
DBN	Decision Bayesian Network
EXP	Experimental
GR	Grade
GRP	Group
LW	Low

Chapter 2
ALEM: A Reference Model for Educational Adaptive Web Applications

Mohammed Tadlaoui[1], Azeddine Chikh[2], and Karim Bouamrane[3]

[1] Ecole Preparatoire Sciences et Techniques, BP165 RP Bel Horizon, Tlemcen, Algeria
mtadlaoui@hotmail.com
[2] College of Computer and Information Sciences, King Saud University, P.O. BOX 51178, Riyadh 11543, Kingdom of Saudi Arabia
az_chikh@ksu.edu.sa
[3] Departement d'informatique, Oran Senia University, BP 1524 EL Mnaouer, Oran, Algeria
bouamrane.karim@univ-oran.dz

Abstract. The adaptive hypermedia systems or adaptive web applications is a research area between hypermedia and user modeling. It can customize hyperspace to different users. The existing reference models are generic and are not dedicated to educational systems. This chapter presents in the first part, a reference model that is specific to adaptive educational hypermedia systems. This model is called ALEM (Adaptive Learning Environment Model). It consists of a domain model, a learner model, a course structuring model and an adaptation model. The main contribution of this model is modeling the adaptive learning curriculum. Furthermore, we develop the UML Tutor application which is an educational adaptive hypermedia system based on our reference model.

2.1 Introduction

Adaptive hypermedia systems (AHS) is an area of research that tries to provide the user with content adapted to his needs. AHS are used in several application domains such as educational systems, information systems, online help systems and online systems for information retrieval.

AHS is a set of nodes and links that allow a user to navigate in the structure of the hyperspace and dynamically customize the various visual aspects of hypermedia to the user's needs. Two types of adaptation exist (De Bra 2008): content adaptation and link adaptation. The first type is used to display and adjust the content of the pages to the characteristics and needs of the user. The second type allows to customize and limit the possibilities of navigation in hypermedia.

There are several methods to implement these two types of adaptation such as the comparative explanation or the additional explanations for the content adaptation and annotation, sorting or link hiding for link adaptation.

A. Peña-Ayala (Ed.): Intelligent and Adaptive ELS, SIST 17, pp. 25–48.
springerlink.com

A more recent taxonomy of adaptation methods and techniques can be found in (Knutov 2009). It distinguishes between content adaptation and presentation adaptation. Three models are used to adapt the hypermedia to the user's needs. These models are: 1) the domain model which is a representation of the subject of hypermedia through concepts and links between them; 2) the user model that represents user characteristics and needs; 3) the adaptation model that contains the rules for adaptation. In AHS, the user model is named learner model.

(Brusilovsky and Millan 2007) distinguish two types of user models: Models that represent the characteristics of the user as the knowledge, interests or goals and models that represent the work context of the user such as location or platform of the user. The first models are important to all adaptive web systems, while the latter are mainly used for adaptive mobile systems.

The user model may be a part of the AHS or it may be shared with multiple systems. In the latter case, we speak about user modeling servers. This type of servers is used in distributed environments where multiple adaptive systems access to this server to query or update user information.

This paper is particularly interested in the adaptation of the course plan to the learner specificities. A course plan is a path that a learner may take to meet a goal of a given training. In other words, a course plan is an ordered set of training resources that a learner must perform to reach his goal.

This chapter is organized as follows: Sect. 2.2 gives an overview of the most known reference models describing the AHS. Sect. 2.3 presents the reference model ALEM that we propose to model the educational AHS. Sect. 2.4 describes a prototype of the ALEM reference model which is called UML Tutor. The aim is to describe the various application modules that interact to satisfy the requirements of the ALEM reference model. This chapter will conclude with a presentation of possible extensions of our model and future work.

2.2 Adaptive Hypermedia Reference Models

The reference models describing the classic hypermedia systems (not adaptive) have begun to appear before the existence of the web, including the model Hypertext Abstract Machine (HAM) (Campbell and Goodman 1988). After the appearance of this model, others have followed and the best known of them is the Dexter Hypertext Reference Model (Halasz and Schwartz 1994).

The theoretical framework of the AHS explained in the previous section has served to define some reference models. The objectives of these models are:

- Model the existing AHS,
- Provide a platform for describing existing systems and specify future systems,
- Provide a platform to compare the different existing systems,
- Describe the basic concepts of AHS and the relationship between them,
- Separate the content, structure and presentation aspects of hypermedia systems.

Among the existing reference models of adaptive hypermedia systems, we found the seminal work about INSPIRE system (Papanikolaou et al. 2002), AHAM model (Wu 2002), Munich model (Koch and Wirsing 2002) and Social LAOS (Ghali and Cristea 2009). The third reference model is explained below.

2.2.1 The Munich Model

The Munich reference model is a model based on the Dexter model. It was developed independently of the AHAM model. The main contribution of this model is that it uses a graphical language for describing the different components of AHS. The layered architecture of Dexter model has been replaced by a Unified Modeling Language (UML) (OMG 2010) package diagram and the description of the user model, domain model and adaptation model has been illustrated by UML class diagrams. These diagrams are also used to describe different functions that are offered by the three models. Other than the graphical modeling, the Munich model makes the following extensions from its two predecessors:

- The components of the domain model are not only connected by navigational relationships (links), but also by other conceptual relations such as "part of", "prerequisite" and "variant of",
- The user model includes a user manager and a model for each user of the system composed of attributes and values,
- Two types of user attributes are taken into account: the attributes that are domain dependent and domain independent,
- The rules are classified into construction rules, acquisition rules and adaptation rules (content, link and presentation adaptation),
- The adaptation model also models the user behavior (browsing, input and user inactivity).

As AHAM, the Munich reference model is also a model which is not only used for educational adaptive hypermedia systems, but also for other types of AHS.

2.3 The ALEM Model

The weaknesses of some existing reference models and the limitation of others demonstrate the contribution that may exist in the proposed reference model. The ALEM model (Adaptive Learning Environment Model) (Tadlaoui et al. 2010) is an extension of the Munich Reference Model.

The greatest contribution of our model over existing models is the modeling of the course and the learning curriculum. The main objectives that have guided us for the development of this model are:

1. Describe existing and future adaptive educational hypermedia systems,
2. Include the concept of educational activity and the concept of educational curriculum,
3. Take into account all types of representation of the user model (overlay model, perturbation model, stereotyped model, etc.),
4. Model the goals of the learner and distinguish them from the goals of domain concepts.

The ALEM model is described using UML notation. This language has allowed us to perform a visual, rich and intuitive description of our model. It was also useful to show the concepts of our model and the various relationships between them. Items added to our model over the Munich model are marked on next figures with an asterisk (*).

The architecture of the ALEM model contains the same three layers existing in the Munich model, but it extends their functionality to better take into account the modeling of educational systems. In addition to these three layers, we added an educational layer.

The different layers of the model are shown in Fig. 2.1:

- The Within Component Layer contains the content and the structure of the hypermedia nodes and it also serves to separate the other layers from detail specific to media,
- The Storage Layer stores information on the structure of the hypermedia. This layer is composed of three models:
  - The domain model describing the scope of the hypermedia,
  - The learner model describing the learner characteristics useful for customizing the hyperspace,
  - The adaptation model describing the adaptation strategies and adaptation rules.
- The Educational Layer is an abstract representation of the course. This layer contains the structural model of the course,
- The Run-Time Layer is the description of how the nodes are presented in the front-end. This layer is responsible for interaction with the learner, acquisition of the learner data and management of sessions.

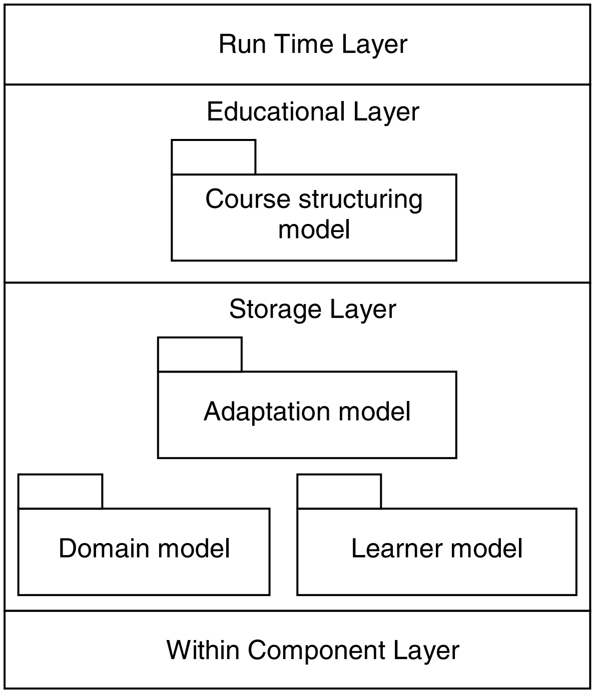

Fig. 2.1 The structure of the ALEM model

2.3.1 Domain Model

The domain model describes the structure of hypermedia as a set of components. Fig. 2.2 shows the domain model and its repartition in the storage layer and within component layer.

The application domain of the hypermedia is modeled by the class "Domain". It is described by attributes which permit to expose the definition of the application domain to other adaptive hypermedia for interoperability purposes. A domain is composed of a finite set of components.

The "Component" class is used to represent abstractly all components of the application domain: concepts, pages, fragments, goals and relationships between components. A component can be described by several descriptors using the Learning Object Metadata (LOM) formalism (IEE LTCS 2010) which is a standard that provides a set of attributes for describing learning objects.

The domain model can also describe, through the class "Presentation specification", how to present a component or relationship to the end user.

A concept is an abstract representation of the application domain information. It is defined by one or more pages.

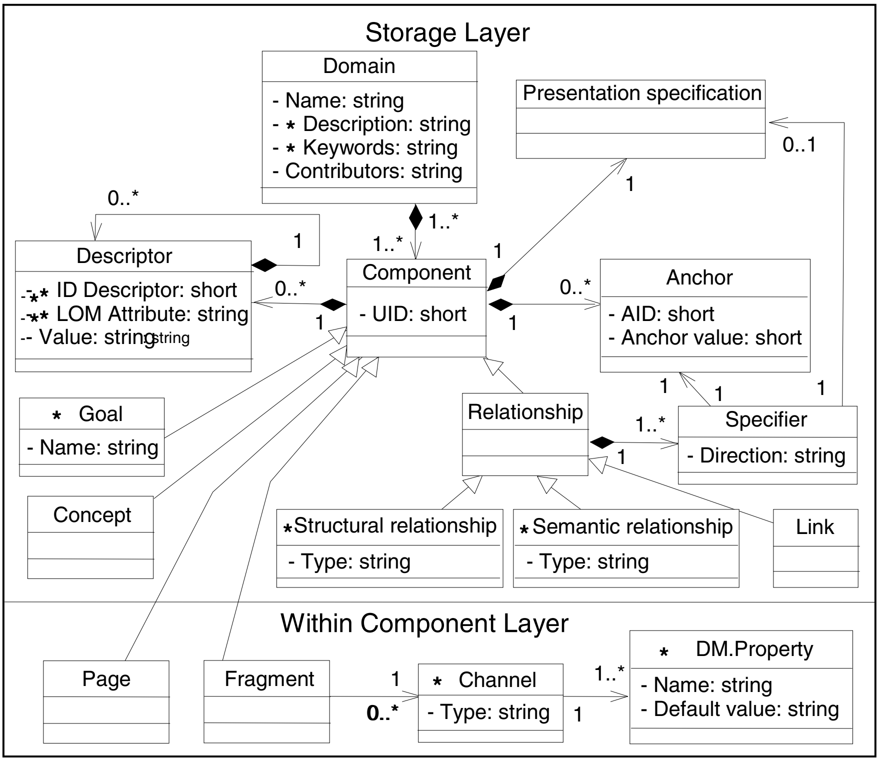

Fig. 2.2 Domain model

The page and fragment components are contained in the within component layer because they represent the content part of the hypermedia. A page consists of one or more fragments. A fragment belongs to a media channel (audio, video, Image, text, etc.). Each channel is described by properties (audio volume, text style, brightness of images, etc.) which are used to personalize the presentation.

The class "Goal" represents the objectives for which the component is created. A goal can be achieved by itself or by other goals. The reference model provides the possibility to define a hierarchy of goals associated with *n*-ary relationship of type “and” and “or”.

The component relationship is a link mechanism between various components of the domain. As shown in the model, a relationship can be either:

- A navigation link: It is the link that allows the user to move from one page to another to browse the hypermedia. This type of relationship connects the pages and fragments,
- A semantic relationship: It is used to express any kind of semantic links (prerequisite of, is similar to, is a version of, and, or, before, after, is important in, etc.) to connect all types of components,

- A structural relationship: It is used to express a composition relationship between concepts, pages and fragments. Among the possible structural relationships there is: is a, is part of and defines the concept.

A relationship may contain one or more specifiers to allow the description of the reflexive, binary or *n*-ary relationship. Each specifier is pointing to an anchor of a component. For example, in the case where the relationship is an hyperlink, the relationship must be composed of two specifiers. One point to the anchor of the source page and the other to the anchor of the destination page. In the source specifier, the value of the direction attribute is set to "From", and the other to "To". The value of the destination anchor is the URL of the destination page. By this mechanism of anchors, the most complex relationships can be modeled.

2.3.2 Learner Model

The learner model describes the learner by an identifier (LID) and a set of attributes. With these attributes the adaptive hypermedia system can represent the characteristics that are relevant for the application. We can distinguish several types of information contained in a learner model: name, background, experience, goals… classified in seven categories. The values assigned to attributes represent what the system believes about the learner. The learner characteristics are given below:

- Personal information: It is about information regarding the learner, such as: name, age, language, educational level, diplomas, certificates, etc.,
- Domain dependent knowledge: It is the knowledge that the user has acquired about a concept of the application domain of the hypermedia. It can be an exact value that a learner has about concept or a probability that a learner knows a particular concept,
- Domain independent knowledge: It is the knowledge in domains related to the domain of hypermedia, which are relevant for adaptation,
- Purpose: It represents the goal to be achieved by the learner. The class "Purpose" holds a time attribute to represent the time to achieve the purpose,
- Physical preference: It is related to the channel of media (audio volume, font, video speed, etc.),
- Cognitive characteristics: They are:
  - Cognitive capacity for example the speed of learning,
  - Cognitive preference, such as the type of interactivity with the system (active or passive), the density of content, the degree of difficulty, the resource type (formal, graphical, simulation, etc.).

The ALEM model allows taking into account the notion of stereotypes. These stereotypes have features with default values that are used principally in the initialization values of the characteristics of the learner. This model and its relationship between classes "Component" and "Property" is illustrated in Fig. 2.3.

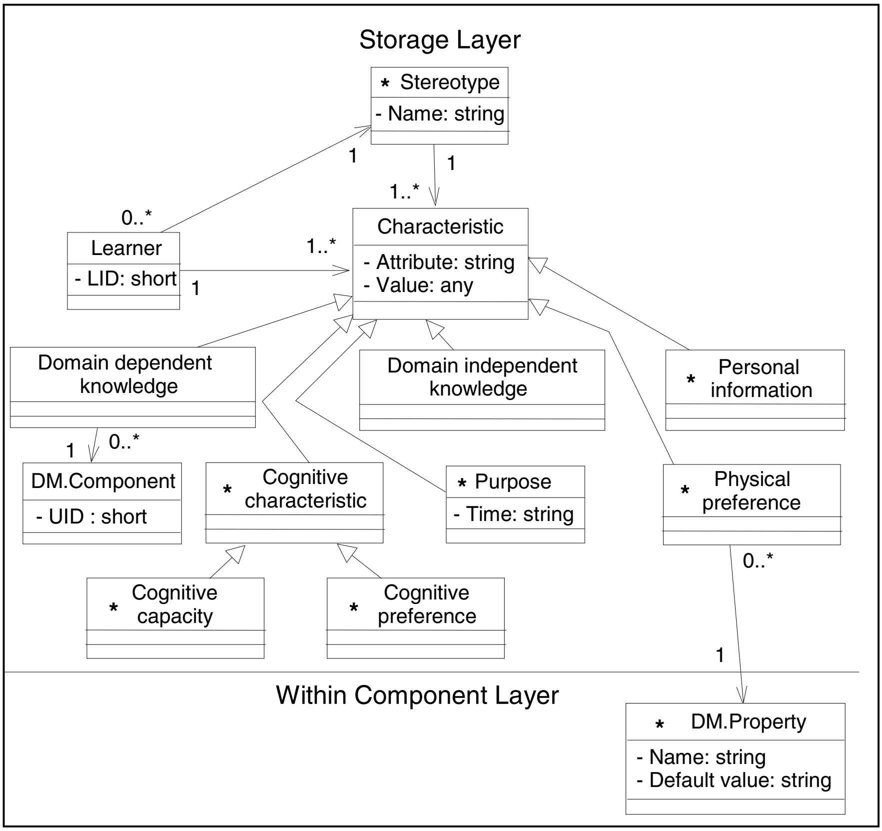

Fig. 2.3 Learner model

2.3.3 Adaptation Model

The adaptation model of ALEM, shown in Fig. 2.4, describes how the adaptation of link and content is made and how the learner model is updated. Adaptation is done using information from the domain model, the learner model and the learner interaction. The adaptation operation is performed by the adaptation engine. The basic element used for adaptation is the rule that determines how the pages are constructed and how they are presented to the learner.

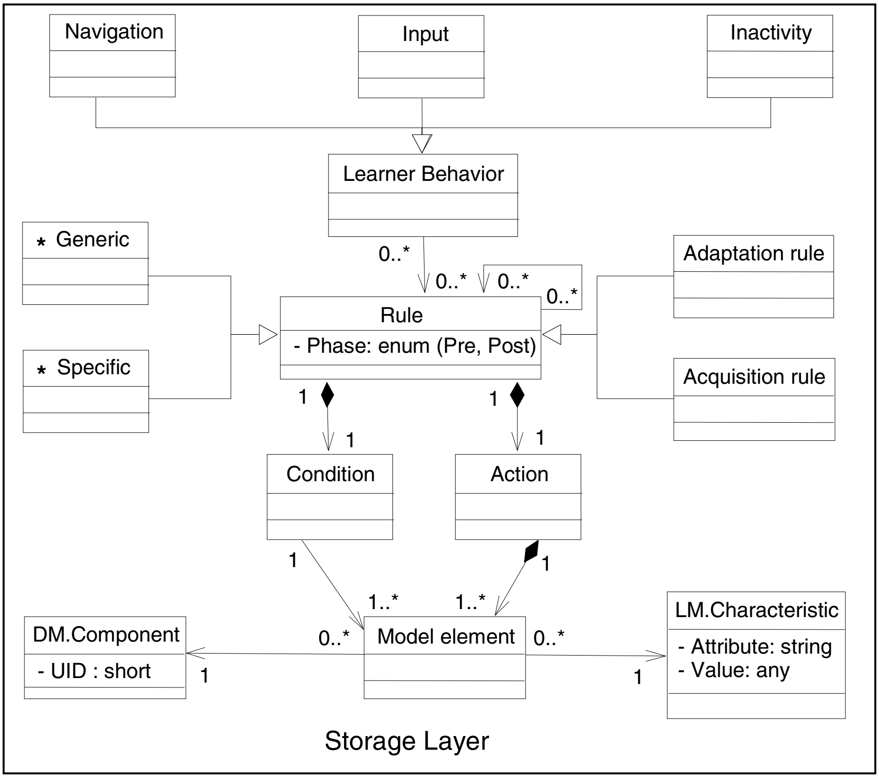

Fig. 2.4 Adaptation model

A rule consists of two parts:

- A condition necessary for the application of a rule,
- An action resulting from a rule. It can be the update of the learner model or the adaptation of the content and presentation.

The two parts of a rule contain expressions which are composed of elements and logical operators. These elements are mainly related to a characteristic of a learner or a component.

A rule can be applied before or after the page generation, following to the attribute phase of the class "Rule" which can take as value "post" or "pre".

A rule can belong to one of the following classes:

- Adaptation rule: It is used to adapt the content and links to the application,
- Acquisition rule: It is used to update the learner model.

An AHS may have predefined adaptation rules (class "Generic"). If these rules are not enough, other new rules (class "Specific") can be defined. The creation of a generic adaptation rule is made by the system designer, while a specific rule is created by the authors of the hypermedia system.

The behavior of the learner, embedded in the adaptation model, is classified according to the actions of the learner: navigation, input and inactivity. A rule is triggered either by the behavior of the learner or by another rule. Our model represents the rules in general and it is not a representation formalism of rules. The syntax of permitted rules depends on the hypermedia system.

2.3.4 Course Structuring Model

A course is the set of educational activities chosen to represent a specific material to meet a very specific purpose. In the ALEM model, the structure of a course is modeled as an "and/or" tree, as it is shown in Fig. 2.5.

The course structuring model is composed of four types of nodes:

- Purpose: This is the final objective that a learner must reach at the end of the course. For example: revising UML within 10 days. A purpose is decomposed into several goals,
- Goal: This is part of the "and/or" tree that defines the intermediate goals between the purpose and activities. A goal can be decomposed into other goals and is realized by one or more activities,
- Activity: This is an operational goal. It defines a task that the learner must perform, such as acquiring a concept, solving a problem, listening to an audio clip. It must be performed to meet a goal. An activity is connected with components defined in the domain model,
- Component: This is the element on which an activity is executed. It represents the educational resources. It can be a concept, a page or a fragment.

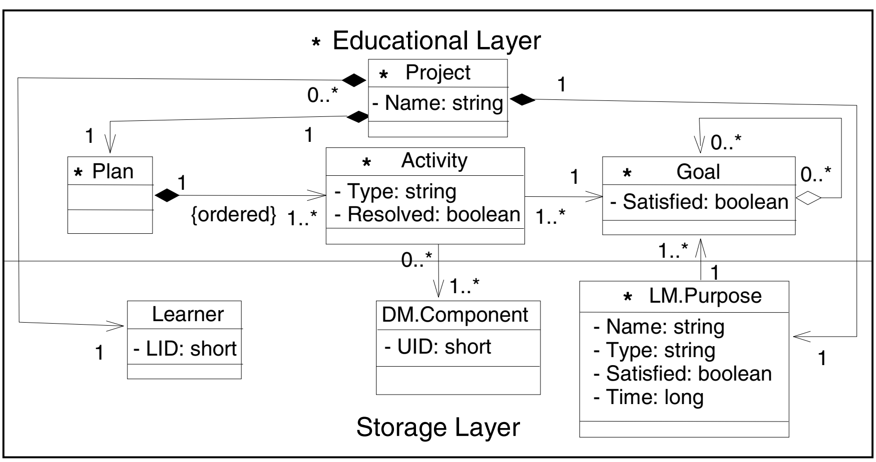

Fig. 2.5 Course structuring model

The course structuring model can represent the course structuring model. From a course “and/or” tree, the system generates a sequence of activities that the learner must follow to achieve the purpose. This sequence is called a plan (learning curriculum).

In this model, we define an educational project as a package that contains a purpose and a plan and which is assigned to a specific learner. A learner may have several projects in progress.

2.3.5 The Process of Generating the Appropriate Curriculum

After that the learner chooses a purpose, the system must perform the following steps to generate the appropriate learning curriculum:

- Update the model of the learner with the value chosen for the purpose,
- Select a subnetwork from the domain model that satisfies this purpose,
- Eliminate from the resulting network the components that are already acquired by the learner,
- Build the tree course model:
  - Position the components (concepts, page and fragments) in the bottom of the tree (leaves of the tree),
  - Set their strictly higher goals in level n-1 of the tree and name them as activities,
  - Add in the lowest levels of the tree (level $< n$-1) the higher level goals,
  - Add in level one (root of the tree) the purpose.
- Choose a path among the different possible paths to reach the purpose based on adaptation rules,
- Update the learner model by the values of project, activities, goals and plan.

2.4 UML Tutor System

To validate our reference model described previously, we defined an architecture and developed a prototype called UML Tutor which is used to teach UML. The prototype that we present in this section aims to propose an adaptive learning environment respecting the different concepts and processes of the ALEM reference model.

The prototype allows the possibility to integrate existing learning resources (documents) in the educational system. These resources are described using meta data respecting the LOM formalism. The system allows creating concepts, to annotate them and to link them to documents. The goal is to add semantic information to manipulate these resources to adapt them to the demands of the learner. The various entities of hypermedia are represented in a visual manner using graphs. We remember that all these information are stored in the domain model, the learner model, the adaptation model and the course structuring model.

In what follows, we will first define the different actors of the system and actions performed by these actors. We will then explain the application architecture. Finally, we will present the various modules of our prototype (UML Tutor) with the application GUI (Graphical User Interface).

2.4.1 UML Tutor Architecture

According to the ALEM reference model, there are two types of actors (system users): Teachers providing functionality for managing learning resources and learners who use the system to self train. To enable interoperability between UML Tutor system and other educational adaptive hypermedia systems we added a secondary actor named "other system".

UML Tutor system provides different application modules for that the three types of actors can perform their actions. Each module is related to a type of action. The system allows the learner to initialize his or her profile and perform self-training. It also allows the teacher to create and manage the hypermedia domain, courses structures and adaptation and acquisition rules. Application modules are grouped by type of user. Thus, the architecture consists of two applications offered to users. One used by the teacher to build adaptive hypermedia and another used by the learner to perform learning activities. Both applications are web applications made using the PHP language. UML Tutor contains a third application that is destined for other systems. It is the interoperability interface. It opens access to the models base to enrich it and to extract information. This application is made with web services.

These three applications are front end modules. They invoke internal modules that can perform basic functions for manipulating information stored in the four models of UML Tutor. The internal application modules are offered as web services. This will enable the modularity and ease of interoperability of the system.

For reasons ease of use of the application, we preferred to split the update of the domain model into two stages:

1. The construction and update of the objects "Concept", "Page" and "Fragment" via the Domain Editor,
2. The construction and update of the hierarchy of the domain goals (building adaptive courses) through the Course Editor.

Fig. 2.6 illustrates the software architecture of UML Tutor system showing the various modules of UML Tutor and interactions between them. This architecture also shows the two databases that the system uses which are the educational documents base and the models base.

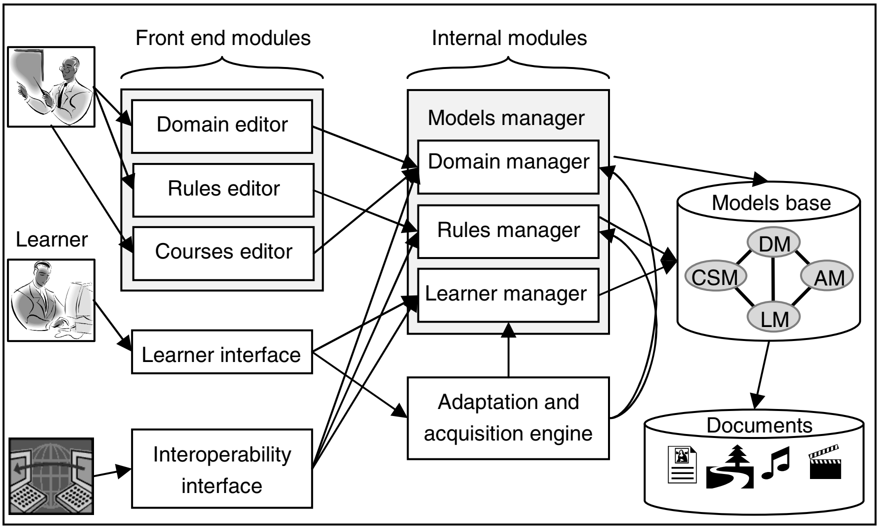

Fig. 2.6 UML Tutor architecture

2.4.2 *Learning Process in the UML Tutor system*

The following steps explain the different tasks performed by the learning environment, the learner and the teacher:

1. The teacher creates the domain model by filling the various concepts, pages, fragments and relations between them,
2. Creation of an adaptive course plan by the teacher describing a course that satisfies a goal (and/or tree),
3. The learner initiates his model by filling a quiz that focuses on his characteristics,
4. The learner selects a purpose,
5. The system generates a course plan adapted to the characteristics and the selected learner purpose,
6. The UML Tutor system presents to the learner the activities in the order of the adapted courses.

2.4.3 *Modules of UML Tutor*

In this subsection, we explain the role of various modules of the architecture of UML Tutor.

2.4.3.1 Domain Editor

This module allows teachers to manage the domain model via an intuitive web GUI (Fig. 2.7). This GUI allows building a graph composed of concepts, pages, fragments and relations between them. The composition of the model is done by drag and drop from the "Object" palette.

Each object in the palette is defined using several properties as follows: Concept (color, name and LOM description); Page (color, name, URL and LOM description); Fragment (color, name, URL and LOM description); Relationship (color and type). The description of these properties is presented as follows:

- Color: The display color of objects. It is used to make the model more readable,
- Name: The name of the object instance. It is unique throughout the system,
- URL: The path to the document which may be a local file (ex: "c:\documents...") or a remote file (ex: "http://www.UMLTu..."). The system offers the possibility to browse files on local and network directories,
- LOM Description: An XML clause which describes the object using the LOM formalism. The click on the button of this property displays a window which allows to describe the object using the 45 attributes of LOM grouped in 9 categories then transform them into XML format,
- Relationship type: The type of links that connect the various objects (composition, prerequisite, define, include, etc.).

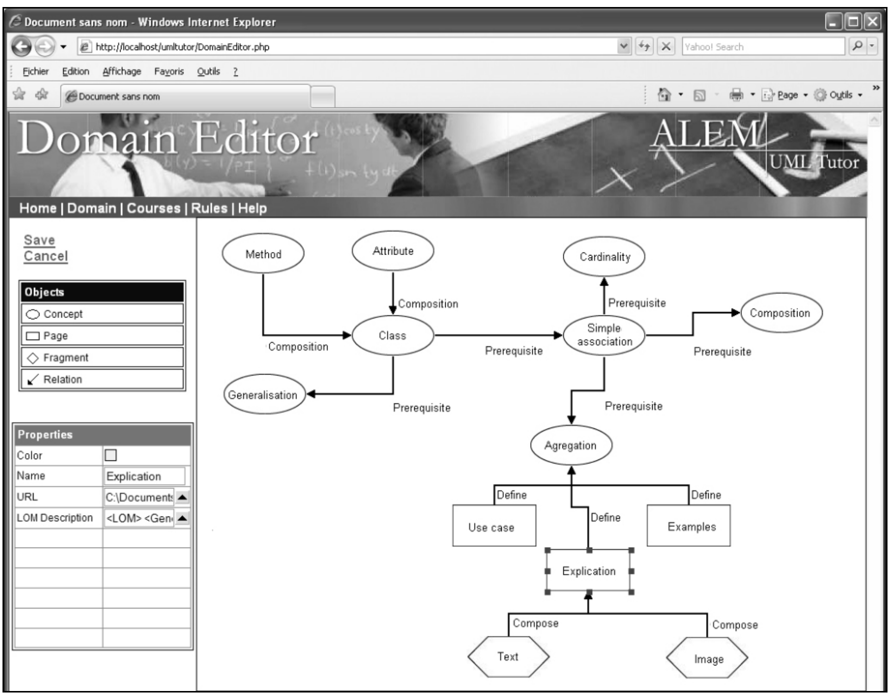

Fig. 2.7 Domain Editor

2.4.3.2 Courses Editor

The application module "Course Editor" allows creating and maintaining adaptive courses. As a reminder, this module does not interact with the structuring course model but with the domain model through the Domain Manager. The domain model consists of a forest of and/or trees (several adaptive courses). In this forest there are trees that can share the same nodes, which means that there may be activities, goals or resources that are common to several adaptive courses.

From the first page of this module the teacher can view the list of existing adaptive courses which are identified by their names and can also add, delete or modify an adaptive course.

Fig. 2.8 shows the detail page of an adaptive course represented in an “and/or” tree who is called "Class diagram". From this page it is possible to construct the tree using purpose, goal and activity objects by "drag and drop" items present in the objects palette. The purposes, goals and activities are transformed into goals when stored on the domain model.

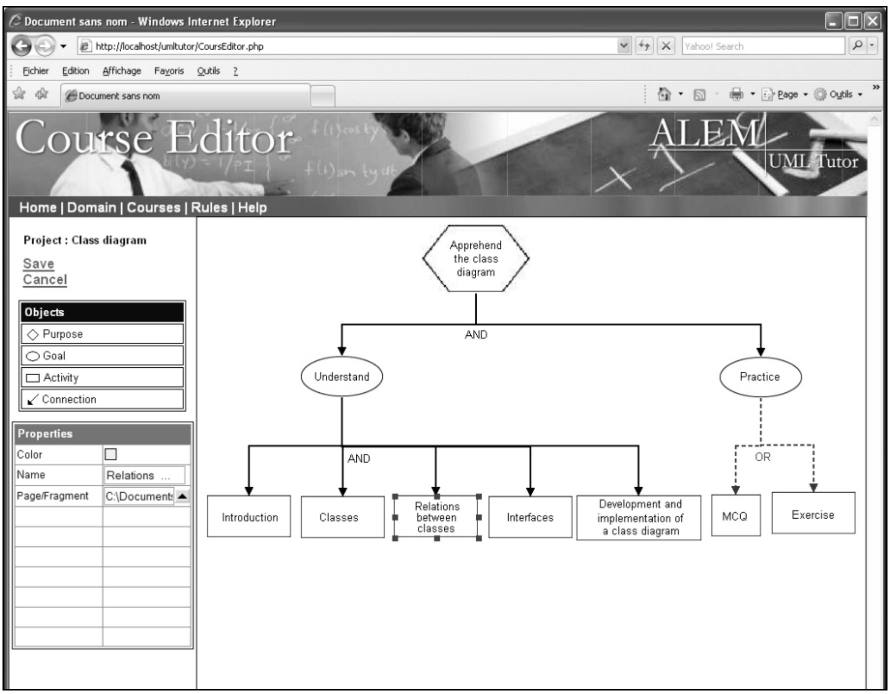

Fig. 2.8 Courses Editor

Each object is defined using several properties as follows: Goal (color and name); Activity (color, name and page/fragment); Relationship (color and type). The description of these properties is given below:

- Color: The display color of objects. It can be used to make the model more readable,
- Name: The name of the object instance. It must be unique throughout the system,
- Page/Fragment: The path to a page or a fragment which already exists in the domain model. To ease the teacher to identify the right page or fragment, the system offers the ability to search using the LOM attributes,
- Relationship type: The type of links that connect the various objects (AND, OR).

2.4.3.3 Rules Editor

This module is used by teachers to define the adaptation and acquisition rules. It interacts with the rule manager to maintain the adaptation model.

From the first web page of this module the teachers can view a list of available rules which are identified by their names and can also add, delete or modify a rule. Fig. 2.9 shows the detail page of a rule.

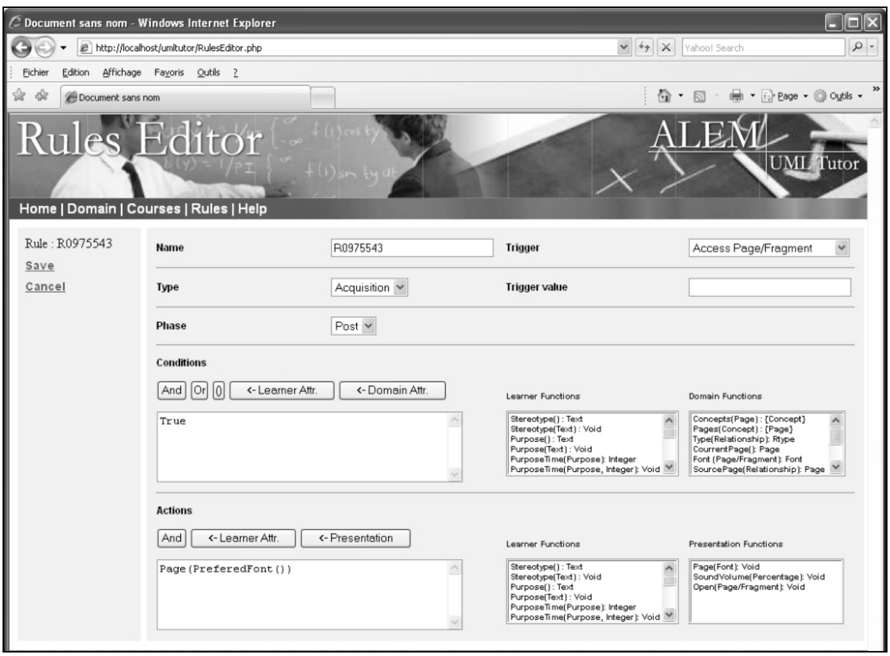

Fig. 2.9 *Rules Editor*

From this page it is possible to describe the properties of the rule, its conditions and actions:

- Name: The name of the object instance. It must be unique throughout the system,
- Type: The type of the rule, if it concerns an adaptation or acquisition rule. We remind that the first type is used to adapt the hypermedia to the requirements of the learner and the second is to update the learner model,
- Phase: The moment of the rule launching, before (Pre) or after (Post) the generation of the page and its display to the learner,
- Trigger: The event that triggers the rule. The UML Tutor system contains 5 possible triggers:
  - Access page/fragment: The rule is triggered when the learner accesses a page or a fragment,
  - Learner inactivity: The rule is triggered when the learner is idle for a specified duration. This duration is specified in the property "Trigger value",
  - Scroll page: The rule is triggered when the learner scrolls a page,
  - Learner input: The rule is triggered when the learner fill an input value. This value is specified in the property "Trigger value",
  - Executed rule: The rule is triggered after the execution of the rule specified in the property "Trigger value".
- Trigger value: The parameter for some types of triggers,
- Conditions: All the conditions for triggering the rule are separated by operators "and/or" and parentheses. The conditions can be domain functions or learner functions,
- Actions: The set of actions to be executed when a rule is triggered. These actions are separated by the operator "and" and might be either learner functions or presentation functions.

UML Tutor system provides to the teacher several functions for writing the conditions and actions of rules. The system can compose between different attributes and functions. Below are some attributes and functions grouped by type:
Domain functions: It holds the next attributes and functions:

- Concepts (Page): {Concept}, this function returns all the concepts related to the page specified as a parameter,
- Pages (Concept): {Page}, this function returns all the pages linked to the concept specified as a parameter,
- Relations (Concept/Page): {Relationship}, this function returns all the relationships related to the page or concept specified as a parameter,
- LOM (Concept/Page/Fragment, attribute): Text, this function returns the value of the specified LOM attribute about a concept, a page or a fragment,
- Font (Page/Fragment): Font, this function returns the font of the page or fragment specified as a parameter,
- SourcePage(Relationship): Page, this function returns the source page of the relationship specified as a parameter,

- DestinationPage(Relationship): Page, this function returns the destination page of the relation specified as a parameter,
- The above two functions also exist for the concepts: SourceConcept (Relationship): Concept and DestinationConcept(Relationship): Concept,
- Type(Relationship): Rtype, this function returns the type of the relationship specified as a parameter. The type can be prerequisite, variant, compose, etc.,
- CurrentPage(): Page, it returns the current page that the learner is browsing.

Attributes/characteristics of the learner: It contains the next attributes and functions:

- Stereotype(): Text and Stereotype(Text): Void, the first function returns the name of the stereotype of the current learner and the second updates it,
- Purpose(): Text and Purpose(Text): Void, the first function returns the name of the purpose that the learner has chosen and the second updates it,
- PurposeTime(Purpose): Integer and PurposeTime(Purpose, Integer): Void, the first function returns the time chosen by the learner to achieve the purpose specified as a parameter and the second update it,
- ReadPage(Page/Fragment): Boolean and ReadPage(Page/Fragment, Boolean): Void, the first function returns "True" if the learner have read the page or the fragment specified as a parameter otherwise the function returns "False" and the second updates the status of reading the page or fragment,
- ConceptAcquired(Concept): Boolean and ConceptAcquired(Concept, Boolean): Void, the first function returns "True" if the learner has acquired the concept specified by the parameter otherwise the function returns "False" and the second updates the status of acquisition of the concept,
- PreferedSoundVolume(): Percentage and PreferedSoundVolume(Percentage): Empty, the first function returns the sound volume preferred by the learner and the second updates it,
- PreferedFont(): Font and PreferedFont(Font): Void, the first function returns the value of the font preferred by the learner and the second updates it,
- CurrentActivity(): Activity and CurrentActivity(Activity): Void, the first function returns the current activity that the learner is achieving and the second set the activity specified as parameter as the next activity to execute,
- Page(Activity): Page, returns the page related to the activity specified as parameter.

Functions of the presentation: It embraces attributes and functions, such as:

- Page(Font): Void, this function is used to change the font of the page displayed to the learner,
- SoundVolume(Percentage): Void, this function changes the sound volume of the media that appears to the learner whether audio or video,
- Open(Page/Fragment): Void, this function opens the page set as a parameter.

2.4.3.4 Learner Interface

Learning interface is used by the learner to perform the necessary activities to achieve educational purposes. This module interacts with the "Learner manager" and the "adaptation and acquisition engine". The learner interface is divided into two parts (pages) as follows.

Edition of the learner profile: From this page the learner can set his profile (Fig. 2.10). The learner needs to update the greatest possible number of parameters for a better adaptation. Below are the 6 types of parameters of the learner profile in UML Tutor:

- Personal information: This page lists the general characteristics of the learner to set such as the first name, last name, school level and language,
- Knowledge of UML: This page lets to fill the level of understanding of domain dependent knowledge (UML),
- Knowledge on the Entity Relationship model: This is the page where the learner can fill the level of understanding of domain independent knowledge which can be useful for the adaptation,
- Presentation Preference: This page offers the ability to specify physical preferences of the learner such as the sound volume or the font,
- Cognitive Preference: It is the page where the learner defines his cognitive characteristics such as density of content, the preferred type of resources,
- Purpose/Project: On this page the student chooses one purpose from the goals of the domain model. UML Tutor offers multiple ongoing learning projects. Therefore, it is possible to switch from one project to another via a setting on this page.

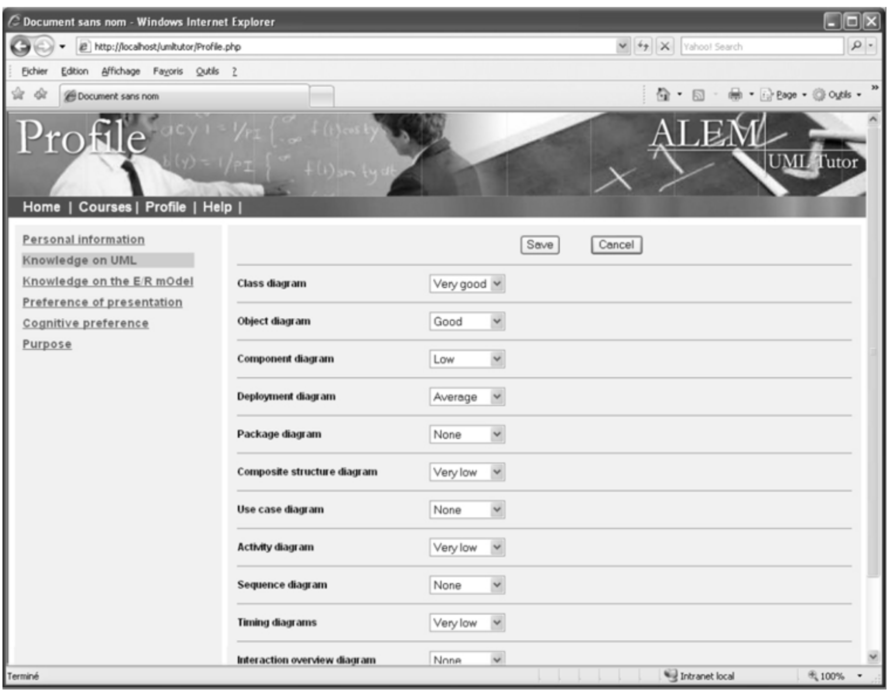

Fig. 2.10 Edition of the learner profile

Learning GUI: Once the learner chooses a purpose, the system generates a course plan adapted to his characteristics and his requirements. This course plan is displayed on the left page of the learning GUI (Fig. 2.11). Each line represents an activity such as the resolution of an exercise integrated into a web page, conducting a simulation, etc. The right frame shows pages related to the activities.

2.4.3.5 Interoperability Interface

This module is used by external systems to import or export the information stored in the four models of UML Tutor, i.e., the domain model, the learner model, the adaptation model and the course structuring model. To increase the accessibility of this interface and make it easily usable by external systems, we chose to develop it as a web service.

Fig. 2.11 Learning GUI

The interoperability interface conducts its extraction operations or updates via the domain manager, the rules manager and the learner manager.