An Agent-based Intelligent Tutoring System for Enhancing E-Learning/E-Teaching

José M. Gascueña, Antonio Fernández-Caballero

Abstract

Intelligent Tutoring Systems (ITS) have proven their worth in multiple ways and in multiple domains in Education. In this paper the application of an agent-based Intelligent Tutoring System for enhancing e-learning / e-teaching is introduced. The general architecture of the ITS proposed is formed by the three components that generally characterize an ITS – the Student Model, the Domain Model, and the Pedagogical Model. Also, an Educational Model has been added to the ITS, which provides the functionality that the teacher of the system needs. The Pedagogical Model provides the necessary mechanisms to efficiently present the matter to the student (pedagogical strategies). Precisely, in the Pedagogical Model four agents – the Preferences Agent, the Accounting Agent, the Exercises Agent and the Tests Agent - have been incorporated to monitor the progress of the students and to propose new tasks. The Educational Model is one of the most important contributions in our experience. It offers recommendations of how to enhance the layout of the presentation. Across this module the teacher changes his preferences, gives reinforcement to the students, obtains statistics and consults the matter.

This document is structured as follows. Firstly, a general introduction is provided. Afterwards, the state of the art of agent-based ITS in education is offered. Next, we explain the objectives of our ITS and the elements that form its architecture. Afterwards, the pedagogic strategy as implemented in the Pedagogic Module and a detailed description of the agents that monitor the progress of the students and that propose new tasks is described. Lastly, we show the capabilities offered by the Educational Model of the system to professors. Finally, we draw some conclusions.

Keywords: Intelligent Tutoring System, Agent system, E-learning, E-teaching, Distance learning

Introduction

ITS are programs that possess a wide knowledge on certain matter, and their intention is to transmit this knowledge to the students by means of an interactive individualized process, trying to emulate the form in which a tutor or human teacher would guide the student in his learning process. They are growing in acceptance and popularity for several reasons, including: (i) an increased student performance, (ii) a deepened cognitive development, and, (iii) a reduced time for the student to acquire skills and knowledge (Sykes & Franek, 2003). Basically, an ITS is characterized for incorporating three models corresponding to three knowledge levels. Firstly, there is a domain model where the domain knowledge is gathered, that is to say the knowledge of what has to be taught. A student model represents the knowledge of the student, that is to say knowledge of what the student knows. Finally, there is a pedagogical model where the knowledge of the instructing strategies, that is to say how to teach the domain knowledge, is described. The goal for every ITS is to communicate its embedded knowledge in an effective manner (Wenger, 1987).

One of the main problems in Intelligent Tutoring Systems (ITS) consists in adapting to the needs of the student who interacts at each moment. A way to provide user adaptation is by means of the so called pedagogical strategies, which specify how to sequence the contents, what kind of feedback has to be given during education, when and how the tutor’s contents (problems, definitions, examples, and so on) have to be shown or explained (Murray, 1999). There has been a great research effort in learning strategies to be incorporated into ITS (Boulay & Luckin, 2001). As an example, Meyer has used the analogy (Meyer, 2002) to teach a less known domain from a more familiar one. The case based reasoning paradigm has also been an inspiration to help in obtaining new incrementing knowledge (Martens, 2004). When various strategies are implemented together in an ITS, as for instance in (Hatzilygeroudis & Prentzas, 2004), the system selects the most appropriate one for the activity that the student is performing. In this paper we will focus on the pedagogic strategy of the pedagogical model of our ITS, thought to introduce the matter in a more efficient way to the students.

On the other hand, agent technology has been suggested by experts to be a promising approach to address the challenges of the modern computer based education (Aroyo, Stoyanov & Kommers, 1999). “*An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future*”

(Franklin & Graesser, 1996). Any agent, in accordance with this definition, satisfies four properties: autonomy, social ability, reactivity and pro-activeness. By using intelligent agents in an ITS architecture it is possible to obtain an individual tutoring system adapted to the needs and characteristics of every student (Cardoso et al., 2004). In this paper a detailed description of the agents which monitor the progress of the students and propose new tasks is also provided.

Related Work

Many learning/teaching computer-based environments framed in the form of ITS use agent technology. Capuano et al. (Capuano, Marsella & Salerno, 2000) have described ABITS. ABITS is suitable to several knowledge domains and is composed by different kinds of agents (evaluation, affective and pedagogical agents) which extend the traditional Course Management System with a set of "intelligent" functions allowing automatic curricula generation. Dorça et al. propose a similar approach (Dorça, Lopes & Fernández, 2003).

Also, the Baghera platform is a web-based multiagent learning environment for assisting students/teachers in learning/teaching of Geometry. Baghera is composed of two multiagent systems (MAS). The higher-level MAS is composed of cognitive agents which provide the main function of the educational system, while the lower-level MAS is responsible for diagnosing student's conceptions (Pesty & Webber, 2004). MyClass includes an agent tutor that can re-plan teaching strategies which best fit a student or group of students at each specific stage during the teaching/learning sessions in a virtual classroom (Mota, Oliveira & Mouta, 2004). Tang (Tang & Wu, 2000) carried out the implementation of a multi-agent intelligent tutoring system for the learning of programming languages. Electrotutor is an Electrodynamics distance teaching environment designed according to JADE - Java Agent for Distance Education framework (Silveira & Vicari, 2002). For collaborative work, I-Help allows the students to locate human peers and artificial resources available in the environment to get help during learning activities (Greer et al., 2001). An architecture of Intelligent Virtual Environments based on the agent paradigm is offered in (de Antonio, Ramírez, Imbert & Méndez, 2005) and for nurse training in (Hospers et al., 2003).

An ITS usually also incorporates pedagogical agents (animated characters) to do learning more attractive and effective (Johnson, Rickel & Lester, 2000). The architecture that we introduce in this paper does not incorporate at present any animated agent. Nowadays, the adaptive and intelligent Web-based educational systems (AIWBES) form a new approach to develop learning/teaching environments using adaptive hypermedia and intelligent tutoring technologies (Brusilovsky & Peylo, 2003). In the area of multiagent hypermedia systems, MASPLANG may be found (Peña, Marzo & de la Rosa, 2002).

Another ITS can be found in core databases, physics, language, mathematics, medicine, and other courses in many schools, but they don't use the agent technology. For instance, KERMIT (Suraweera & Mitrovic, 2004) teaches the conceptual modeling of databases using the entity - relation data model, ELMART (Weber & Brusilovsky, 2001) teaches programming in LIPS, and Design Pattern (Jeremic, Devedzic & Gasevic, 2004) is used to learn design patterns.

Objectives and architecture of the agent-based ITS

The ITS proposed in this paper creates an infrastructure for distance learning/teaching of a matter. In order to obtain good results, we propose to decompose the matter into theory, exercises and test questionnaires (see figure 1). The alumni study each topic of the matter reading theory first, then making exercises and finally answering to a test. The system will provide help to the students whenever it is necessary.

The first goal of the ITS proposed is that the alumni learn more and better, that is to say, to be able to structure learning matter in such a way to facilitate the learning facilities. One characteristic to take into account in learning is the rhythm the student is able to learn. Thus, the ITS has to adapt rhythm it introduces the concepts to the learning rhythm of each student (for instance, to show more or less exercises, to show more or less tests, etc.). Another aspect widely considered in learning theory is reinforcement by rewarding a correct answer and penalizing the errors (by means of messages, sounds, etc.).

Another goal in our environment is to enhance teaching as well as learning. One of the main problems a professor faces when teaching is that he does not know the skills of his alumni. Our proposal leads to conclusions that “teach how to teach”. Within this objective there is the need to make the matter more comprehensive for the overall alumni, but always keeping in mind the requisites given to the subject.

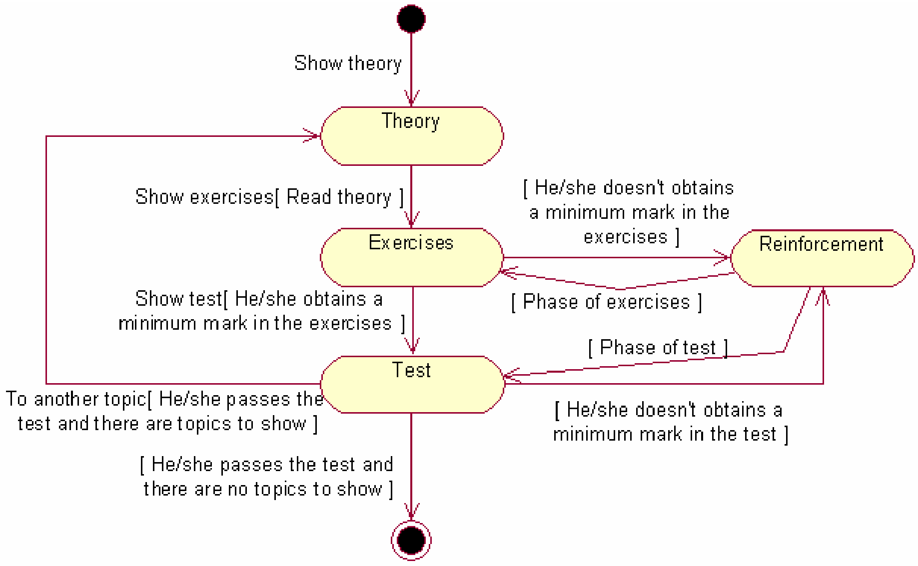

Figure 1: Decomposition of the matter

The general architecture of our ITS (see figure 2) is formed by the three components that characterize an ITS – the Student Model, the Domain Model, and the Pedagogic Module. Also, in the ITS an Educational Model has been added, which provides the functionality that the teacher of the system needs. In the Pedagogic Module four agents have been added. The ITS proposed is not tied to any course in particular, being the only requisite that the course has to be divided into theory, exercises and tests.

In the Student Model the knowledge the system has about the student (profile and interaction with the system) is represented. The model is composed of three knowledge databases (KDBs). (1) The Personal Information KDB stores the necessary personal information of the student to control his access to the system. (2) The Profiles KDB stores the level as well as the presentation styles of the students. The students are assigned different levels depending on their learning rhythm. (3) The Learning KDB stores parameters such as the exercises and tests proposed so far to the students, the time spent on answering the questionnaires and the exercises, the pages of theory visited and the scrolls performed on those pages, or the reinforcement material prepared by the Pedagogic Module.

In the Domain Model the knowledge about the contents to be taught is stored. This model consists of four KDBs: (1) the Theory KDB incorporates the pages of theory that have been prepared for teaching on the matter, (2) the Tests Questionnaire KDB stores the battery of test questions related to the matter, (3) the Exercises KDB stores the battery of exercises on the matter, and, (4) the Reinforcement KDB contains the information used by the Pedagogic Module to prepare the material to be shown when a student needs to be reinforced.

The Pedagogic Module provides the necessary mechanisms to efficiently present the matter to the student. This module is in charge of carrying out three tasks: (1) to provide the learning guidelines for the student (including any necessary reinforcement provided by the system), (2) to update statistics in the Domain Model of the exercises and tests presented, (3) to store into the Learning KDB important data as the material prepared to reinforce the student who needs it, the responses given by the student to the exercises and tests proposed, as well as the punctuation that the student has gotten and the time that he has spent in reaching the aims.

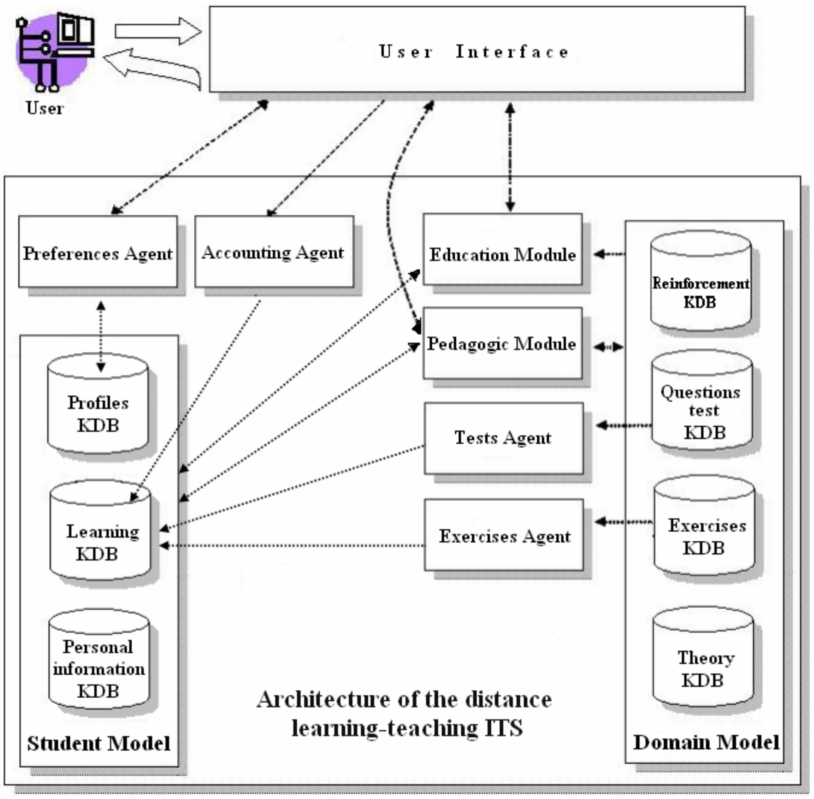

Figure 2: Architecture of the agent-based ITS system

The Preferences Agent supervises the user preferred style of presentation (type and size of letter, colors, margins, and so on). When the user changes his style of presentation the Preferences Agent creates a personalized sheet of styles for the user and updates the user's interface in accordance with his new pleasures. The information that this agent gathers is stored in the Profiles KDB. The Accounting Agent observes the student interaction with the interface when the pupil accesses a page of theory. When the student changes to another page of theory, the Accounting Agent stores in the Learning KDB some valuable information (the name of the visited page, the time that the student has spent on it and the scrolls performed on it). The Exercises Agent takes charge of choosing the exercises that will be proposed to the student in the topic that he is currently studying. This agent stores the chosen exercises in the Learning KDB as well. In the same way, the Tests Agent is in charge of choosing the test questions that will compose a test questionnaire proposed to the student in the topic that he is studying at this moment. The test questions selected are also stored in the Learning KDB. The Exercises Agent and the Tests Agent do the selection when the student finishes the first visit to the first page of theory of every topic. We may highlight that the Exercises Agent and the Tests Agent are proactive because they carry out their tasks in parallel with the activity that the student performs. Indeed, the student is reading theory without realizing the work of both agents. The agents have been implemented as applets.

Lastly, the Education Module provides the functionality that the teacher of the system needs. Across this module the teacher changes his preferences, gives reinforcement to the students, obtains statistics and consults the matter. This module is in fact devoted to help the teacher to change the contents of the matter on the basis of the information obtained from the Student Model and the Domain Model.

Strategy of the Pedagogic Module

Figure 3 shows the steps followed by the pupil when studying each topic of the course ("Matter learning").

(1) Firstly, the student has to read the whole theory for the current topic.
(2) Afterwards, the student has to solve the exercises proposed. If the student is a level-1 (low level) student, firstly he has to solve the basic exercises and then the complex ones. O the other side, if the

student is a level-2 (high level) student, he will only have to solve the complex exercises. The basic exercises are all shown in a sequential way, and then the ITS evaluates if the student has reached a minimum score associated to the topic. On the contrary, the complex exercises are shown in blocks (composed of a pre-determined quantity of exercises), and, after showing each block, there is an evaluation to ask for a minimum mark before composing the next block. After correctly fulfilling a number of complex exercises, the system goes on to the test questionnaires.

(3) Lastly, the student has to solve the test questionnaire offered.

(4) If there are more topics in the course, the system goes back to step (1). Otherwise, the student has finished studying the matter.

During steps (2) and (3), if the student does not obtain the minimum scores fixed for the topic, he gets reinforcement in order to reach the objectives for the course.

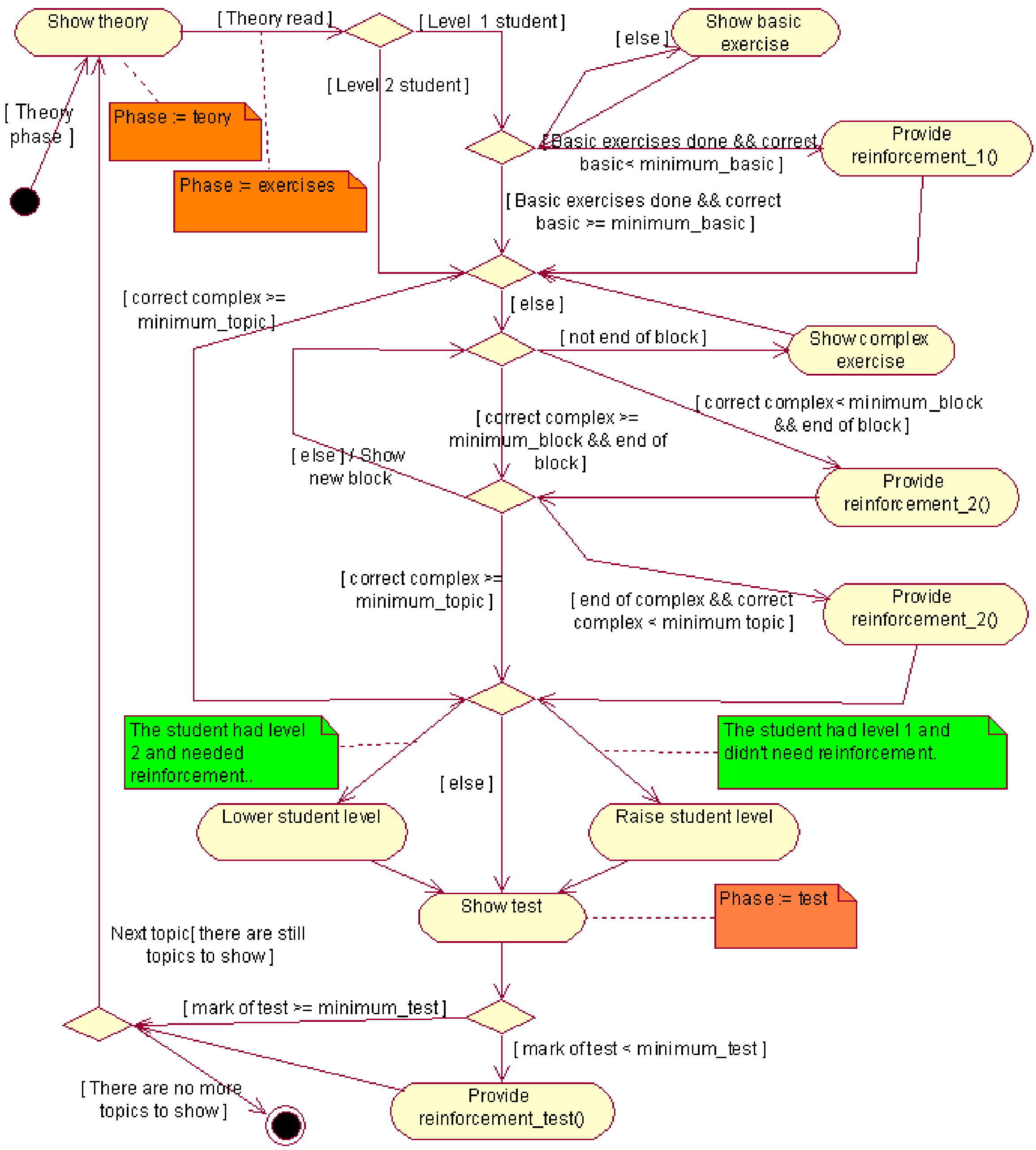

Figure 3: Activity diagram for "Matter learning"

In activity "Provide reinforcement_1" of figure 3, you may appreciate the process followed when the alumni require reinforcement in the basic exercises. The system selects one of the basic exercises previously proposed and not well solved from the set of basic exercises and gets the reinforcement material (based on previous topics studied). This way the system helps the student to correctly solve the basic exercise. After proposing the reinforcement material, and before the student has to solve again the basic exercise, the ITS shows the bad response that the student gave previously. When the student passes the minimum score, the system does not go on providing reinforcement. But, and this is the worst situation, if the system has provided reinforcement to all badly answered basic exercises, and even so the student has not been able to solve them, the ITS tells the student to consult the tutor personally. After having his meeting with the teacher, the student is permitted to advance in the study of the course.

Now, in activity "Provide reinforcement_2" the alumni are reinforced during their phase of solving complex exercises. The strategy for providing reinforcement to the student in complex exercises is very similar to the strategy followed to give reinforcement in basic exercises. The only difference is that the ITS firstly tries to reinforce with material of the current topic; and, if the student is still not able to solve the complex exercise, he is reinforced by material from previous topics of the course. If the student does not have seen all selected complex exercises, he will only get reinforcement for those exercises offered to the user in the last block of complex exercises. But, if he already has been offered all the complex exercises blocks, he will be reinforced for all complex exercises incorrectly solved and not yet reinforced previously.

Finally, activity "Provide reinforcement_test" show what happens if the student does not get a minimum mark in the test questionnaire proposed for the current topic. The ITS builds a new test questionnaire, offers it to the student, and, if the student does not perform well, the professor personally must reinforce in order to proceed with the learning activity.

Description of the Agents

The Preferences Agent

The Preferences Agent supervises the style of presentation that the user likes. It perceives the interaction of the student with the user interface and acts when he changes his tastes. The preference agent is continually running to know the student's preferences at any time. The process that follows when the user decides to change his visual preferences comes reflected in the figure 4. When the student decides to change preferences the Preferences Agent shows him a form with the preferences that the he has selected in this moment. This way the user can perform the changes that he considers to be opportune. After having completed the form, the new elected preferences are updated and an example page is shown to the student with all the features of the new elected style of presentation. If the student does not like the page, he may continue changing his preferences again.

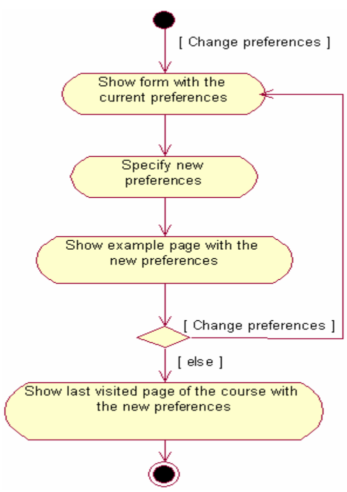

Figure 4: Activity diagram for change preferences

The Accounting Agent

The Accounting Agent perceives the interaction between the student and the user interface when the student accesses a page of theory. Concretely, the agent is in charge of watching the scroll that the student realizes on a page of theory as well as the time that he has remained in that page. When the student leaves studying a page of theory, the Accounting Agent stores all parameters gathered on a page of theory (scroll and time of permanence) in the learning database.

In figure 5 the algorithm implemented to detect the scroll that the student performs when he visits a page of theory is shown. Once the student has entered a theory page, he may advance in his reading or go back in the page. Whilst the student is advancing through the page, the value of "Greatest advance" is being updated. Now, when he steps back the value of "Greatest advance" is stored in the scroll history and the value of "Greatest backward" is being updated. When he decides to advance, it will produce the contrary process, that is to say, the value of "Greatest backward" is stored in the scroll history and the value of "Greatest advance" is being updated again, and the process is repeated. When the student leaves studying the page of theory, the Accounting Agent stores all parameters gathered on the scroll history in the learning database.

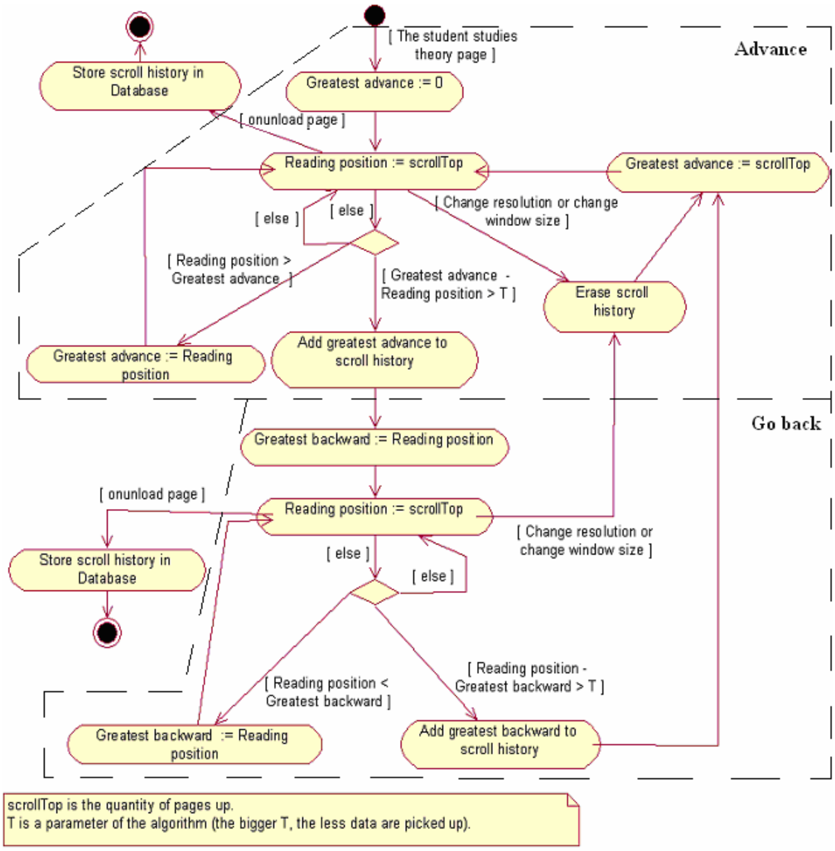

Figure 5: Activity diagram for "detection of scroll"

The Exercises Agent

The Exercises Agent takes charge of choosing the exercises that will be proposed to the student in the topic that he is studying. The Exercises Agent is autonomous as it controls its proper actions in some degree. The agent by its own means (pro-active) selects the set of exercises to be proposed in the subject studied by the student and adds to each exercise the links to the theory pages that explain the concepts related to the exercise. The Exercises agent stores the chosen exercises in the Learning KDB. As it may be observed in figure 6, when the student has just visited for the first time the page index of the topic for which he is studying the Exercises Agent it realizes the selection of exercises that will be proposed to the student in the above mentioned topic. If the student has level of that time 1 first selects the basic exercises (state to prepare basic exercises) and later the complex exercises (state prepare complex exercises). If the student has level of that time 2 alone selects complex exercises. Once it has selected the exercises it will remain inactive (Idle state) while the student I did not go on to the following topic.

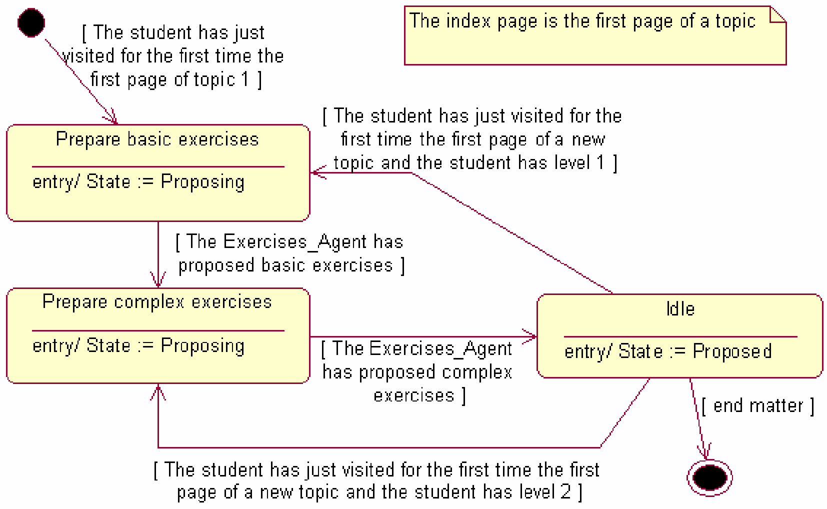

Figure 6: Exercises agent state diagram

The Tests Agent

The Tests Agent takes charge choosing the questions test that will compose the test that will be proposed to the student in the topic that he is studying. The agent by its own means (pro-active) goes on designing a set of tests for the subject the student is engaged in. These tests will be shown to the student in form of a questionnaire. The Tests Agent performs the selection of questions test at the same time that the Exercises Agent realizes the selection of exercises. Once it has selected the questions test it will remain inactive (Idle state) while the student I did not go on to the following topic.

The Education Module

The Education Module is one of the most important contributions to take into account the experience of the teacher. The use case diagram in figure 7 provides the functionality that the ITS offers to the teacher across the Education Module. Evidently, the teacher must be authenticated successfully to accede to the ITS functionality (use case "Authenticate"). Once the teacher has been authenticated, he is able to change his preferences (colors, margins, interlineate, size and type of source of the interface – use case "Change preferences"), to give reinforcement to the students (use case "Reinforce the student") if necessary, obtains statistics (use case "Obtain statistics") and to consult all the didactic material (theory –"Consult theory"-, exercises –"Consult exercises"-, and test questionnaires –"Consult test battery questions") of each of the topics of the matter across. The information provided to the teacher is obtained by the Student Model and the Domain Model; this information was picked up and gathered during the user interactions with the ITS.

The greatest benefit is that the teacher may consult statistics fruit of the interaction of the students with the system. For every topic of the matter, the teacher will be able to know the efficiency of the implemented mechanism by pedagogic module to reinforce the students – for each topic of the course, the teacher may consult the information on the number of times that the students have needed reinforcement and how many times the students have personally been reinforced by the professor.

Respect to the theory read by the students, the teacher may know the number of times that the students have accessed each theory page and the mean time the students have been on each visited page. The statistics are classified as (a) pages read during the theory phase, (b) pages consulted during the exercises phase, either solicited by the student or due to the reinforcement mechanism. The ITS also offers the possibility to know which students have performed scrolling when visiting theory pages and to reproduce the scroll movements as performed.

In the same sense, and in accordance with the exercises proposed to the students, the teacher may observe the possibilities in relation to the exercises statistics. The professor may consult to how many students an exercise has been shown, the mean time the students have spent to solve the exercise and the percentage of blank, correct and incorrect answers to the exercise. There is a classification in exercises presented as reinforcement and normal exercises. The information on how many times an exercise has been explained personally by the teacher is also provided. It is also possible to consult the percentage of pupils that have answered correctly, have answered badly or have not answered an exercise.

Lastly, in relation to test questionnaires, the teacher may look for the number of times that each test question has been presented to the students and the number of times that the students have left the test blank, have answered correctly and have answered incorrectly. It is possible to know if the test question was presented as reinforcement to an exercise, or if it was part of a test questionnaire. Furthermore, the teacher may know the number of times that he had to personally explain the test question.

Another important option available in the ITS is that the teacher can give reinforcement to any student who needs his help. This is because the student has not managed to advance in the study of the subject because the material that has provided to him the pedagogic module is not sufficient to overcome the goals of the topic that he is studying.

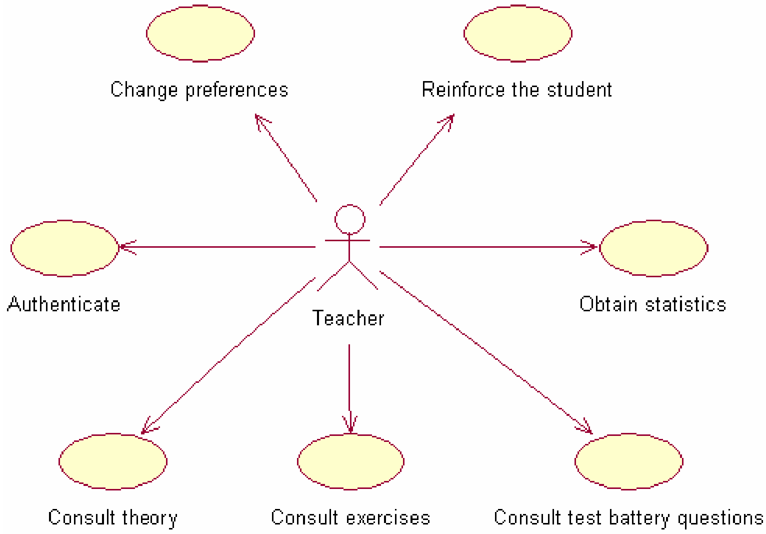

Figure 7: Functionality offered by the Education Module to the teacher

Functional Requisites for the Student

First of all let us focus on the functionality that the ITS offers to the student (see figure 8). Of course, the student must register in the course (use case "Register for the course"), by typing in his personal data; the system shows the login and the password assigned to enter the course. The registered students can change their passwords (through use case "Change password") each time they enter the course to begin a new study session.

Once a new study session has been started through use case "Enter the course", the student reads pages of theory ("Read theory"), answers exercises ("Solve exercise") or test questionnaires ("Solve test") depending on the task that a pedagogic module proposes through time. While completing an exercise, the student can consult the theory (use case "Consult theory") closely related to the exercise.

During a study session the student can also change the style of presentation of the matter (that is to say the visual preferences – "Change preferences"). The student may also consult at any time his state after performing any task.

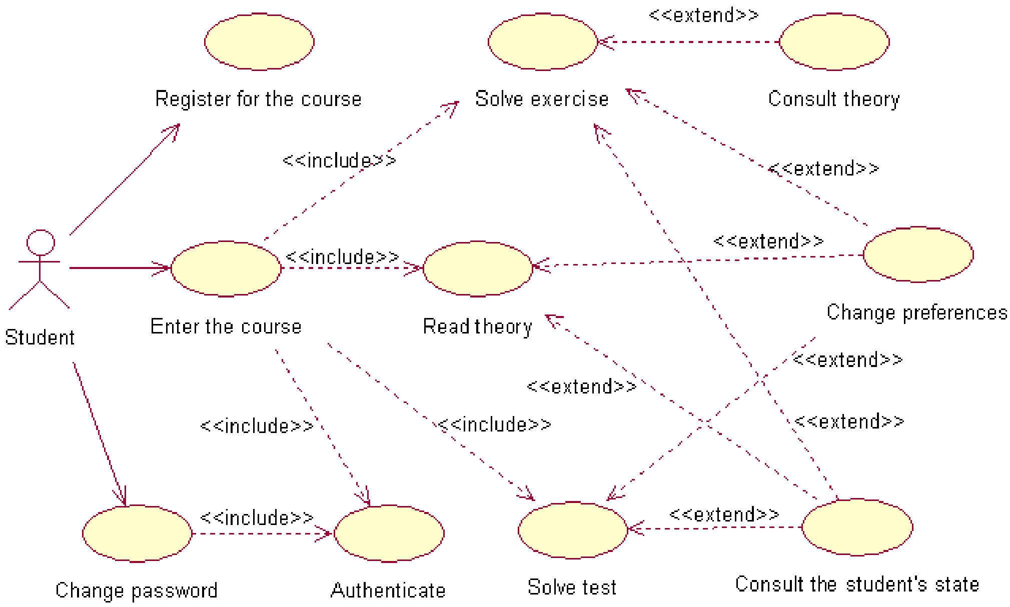

Figure 8: The student's requisites

Conclusions

The ITS have turned into a technology of increasing interest to complement traditional education so much from the perspective of the students as from that of the teachers. In our distance learning system we have introduced a Student Model, a Domain Model, a Pedagogical Model, and an Educational Model. In the pedagogical model four agents – the Preferences Agent, the Accounting Agent, the Exercises Agent and the Tests Agent - have been added.

Also, we have introduced an explanation of how the course adapts to the students as well as to the teachers. User adaptation is provided by means of the so called pedagogical strategies, which among others specify how to proceed in showing the contents of the matter for a better assimilation of the knowledge by the student. Precisely, the Pedagogical Module represents this knowledge.

To conclude, the ITS proposed gets all needed data, obtained fruit of the interaction of the students with the system, to adapt the rhythm of introducing the contents of the matter to the learning rhythm of each student. On the other hand, the Education Module obtains measures that permit to get recommendations to teacher to enhance the course. This way, jointly e-learning and e-teaching are greatly enhanced.

Acknowledgements

This work is supported in part by the Spanish Junta de Comunidades de Castilla-La Mancha PBC-03-003 and the Spanish CICYT TIN2004-08000-C03-01 grants.

References

de Antonio, A., Ramírez, J., Imbert, R., Méndez, G. (2005). Intelligent Virtual Environments for Training: An Agent-based Approach. *4th International Central and Eastern European Conference on Multi-Agent Systems* (CEEMAS'05). Budapest, Hungary. 15-17 September 2005.

Aroyo, L., Stoyanov, S. & Kommers, P. (1999). An Agent-oriented Approach for Ideational Support in Learning - Integration and Impact. *Special Issue on Intelligent Agents for Education and Training Systems, Journal of Interactive Learning Research (JIRL)*, 10 (3/4), 1999, 389-400.

Boulay, B., Luckin, R., 2001. Modelling human teaching tactics and strategies for tutoring systems. *International Journal of Artificial Intelligence in Education*, 12, 235-256.

Capuano, N., Marsella, M., Salerno, S. (2000). ABITS: An agent based Intelligent Tutoring System for distance learning. *Proceedings of the International Workshop on Adaptive and Intelligent Web-Based Education Systems*, ITS 2000.

Cardoso, J., Bittencourt, G., Frigo, L.B., Pozzebon, E., Postal, A. (2004). MathTutor: A Multi-Agent Intelligent Tutoring System. *AIAI 2004 - First IFIP Conference on Artificial Intelligence Applications and Innovations*, 231-242.

Dorça, F.A., Lopes, C.R., Fernández. M.A. (2003). A multiagent architecture for distance education systems. *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies*, ICALT'03, 368.

Franklin, S., Graesser, A. (1996). Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL),* LNCS, 1193, 21-35.

Greer J., McCalla G., Vassileva J., Deters R., Bull S., Kettel L. (2001) Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience. *Proceedings of AIED'2001*, San Antonio, 410-421.

Hatzilygeroudis, H., and Prentzas, J. (2004). Using a Hybrid Rule-Based Approach in Developing an Intelligent Tutoring System with Knowledge Acquisition and Update Capabilities. *Expert Systems with Applications*, 26(4) 2004 477-492

Hospers, M., Kroezen, E., Nijholt, A., op den Akker, H.J.A., Heylen, D. (2003). An agent-based intelligent tutoring system for nurse education. *Applications of Intelligent Agents in Health Care*, J. Nealon and A. Moreno (eds), 143-159.

Jeremic, Z., Devedzic, V., Gasevic, D., 2004. An Intelligent Tutoring System for learning design patterns. *International Workshop on Adaptative Hypermedia and Collaborative Web-based Systems,* AHCW' 04.

Johnson, W., Rickel, J., Lester, J. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, 11: 47-78, 2000.

Martens, A., 2004. Case-based training with Intelligent Tutoring Systems. *Proceedings of the The 4th IEEE International Conference on Advanced Learning Technologies.*

Meyer, C., 2002. Hypermedia environment for learning concepts based on inter-domain analogies as an educational strategy. *International Conference on Intelligent Tutoring System, ITS 2002. LNCS*, 2363, 281-290.

Mota, D., Oliveira, E., Mouta, F. (2004). MyClass: A Web-based system to support interactive learning in virtual environments. *Workshop on Modelling Human Teaching. Tactics and Strategies*.

Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10, 98-129.

Peña, C.I., Marzo, J.L., de la Rosa, J.L. (2002). Intelligent agents in a teaching and learning environment on the Web. *Proceedings of the 2nd IEEE International Conference on Advanced Learning Technologies*, ICALT2002.

Pesty, S., Webber, C. (2004). The Baghera multiagent learning environment: an educational community of artificial and human agents. Upgrade, *Journal of CEPIS* (Council of European Professional Informatics Societies), 4, 40-44.

Silveira, R.A., and Vicari, R.M. (2002). Developing Distributed Intelligent Learning Environment with JADE – Java Agents for Distance Education Framework. *International Conference on Intelligent Tutoring Systems*, ITS 2002. LNCS, 2363, 105-118.

Suraweera, P., Mitrovic, A. (2004) An Intelligent Tutoring System for Entity Relationship Modelling. *International Journal of Artificial Intelligence in Education*, 14, 375-417.

Sykes, E.R., and Franek, F. (2003). A prototype for an Intelligent Tutoring System for students learning to program in Java. *Proceedings of IASTED International Conference on Computers and Advanced Technology in Education*, CATE2003, 78-83.

Tang, T.Y., Wu, A. (2000). The implementation of a multi-agent intelligent tutoring system for the learning of computer programming. *Proceedings of 16th IFIP World Computer Congress-International Conference on Educational Uses of Communication and Information Technology*, ICEUT 2000.

Weber, G., Brusilovsky, P., 2001. ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education*, 12, 351-384.

Wenger, E. (1987). Artificial Intelligence and Tutoring Systems: Computational and Cognitive approaches to the communication of knowledge. Morgan Kaufman Ed.

About the authors:

José M. Gascueña Noheda received his M.Sc. in Computer Science from the University of Castilla-La Mancha at the Superior Polytechnic School of Albacete, in 2004. He is pursuing his Ph.D. in adaptive e-learning systems. His research interests are in Ontologies, Software Agents and Multi-agent Systems, and Learning Objects.

Laboratory of User Interaction and Software Engineering (LoUISE)
Computer Science Research Institute
University of Castilla-La Mancha, 02071 Albacete, SPAIN
Phone: +34 967-599-200 ext. 2406; Fax: +34 967-599-224
jmanuel@info-ab.uclm.es

Antonio Fernández-Caballero received his M.Sc. in Computer Science from the Polytechnic University of Madrid (Spain) in 1993 and his Ph.D. from the Department of Artificial Intelligence of the National University for Distance Education (Spain) in 2001. His research interests are mainly in Artificial Intelligence, and, mainly in Agents Technology. He is currently an Associate Professor in the Department of Computer Science of the University of Castilla-La Mancha (Spain).

Laboratory of User Interaction and Software Engineering (LoUISE)
Computer Science Research Institute
University of Castilla-La Mancha, 02071 Albacete, SPAIN
Phone: +34 967-599-200 ext. 2406; Fax: +34 967-599-224
caballer@info-ab.uclm.es