A Web-Based Multiagent Educational System

Fabiano A. Dorça, Márcia A. Fernandes, Carlos R. Lopes
Universidade Federal de Uberlândia
Faculdade de Computação
Av. João Naves de Ávila, 2160 – 38400-902 – Uberlândia/MG – Brasil
Phone: +55 (034) 3239-4144
E-mail: {fabianodor, crlopes, marcia}@ufu.br

ABSTRACT

This paper describes a web-based multiagent educational system, which presents characteristics of intelligence and adaptability. The system development is based on techniques of Distributed Artificial Intelligence, Instructional Planning and Intelligent Tutoring Systems. All the tasks, which this educational system is able to execute, are distributed into specialized agents that cooperate and collaborate in order to achieve an effective teaching/learning relation. An important contribution of this work is a system that allows adaptation of a given course to different students.

Keywords: ITS, web-based multiagent system, student modeling, knowledge representation.

1. INTRODUCTION

In this paper we describe a multiagent system for web-based distance education (WBDE) with characteristics of adaptativity and intelligence. The distance education is a research and development area in large expansion. The benefits of this type of education include platform and classroom independence. Moreover, a course proposed inside this context can be used to reach a large amount of students, contributing for the knowledge democratization.

The adaptativity mechanism is responsible for making the system capable of changing automatically its own characteristics in accordance with a particular student. These modifications are based on how the student interacts with the system and the degree of knowledge acquired. In this context, the process of monitoring the behavior and evaluating the knowledge of the student is very important.

The proposed system is a result of a long period of investigation and it brings up a lot of recent research techniques developed in the multiagent systems area. The system described here has shown satisfactory behavior during tests and simulations of real situations related to the distance learning activity. The system made possible the development and the implementation of an algorithm for automatic content generation taking into account the dynamic peculiar aspects of web based education systems.

Our approach is based on Intelligent Tutoring Systems (ITS) and Multi-Agent Systems (MAS) to reach the characteristics of adaptativity and intelligence. The development was divided into two main parts, the Course Management System (CMS) and the Multiagent Intelligent System (MIS), which are discussed in Section 2. In Section 3 we discuss about the knowledge representation and the student model implemented in the system. Section 4 presents the main characteristics of the MIS agents. In Section 5 we show up all the implementation issues. Section 6 comments about some related works. Section 7 concludes the article and present the future works.

2. THE PROPOSED SYSTEM

The implementation of the system has been divided in two main parts, the CMS and the MIS. The CMS was firstly implemented after which an array of tests were made with it. Then the MIS implementation was started. The CMS is like a platform where MIS can act adding some important characteristics like intelligence and adaptativity. Without a system such as CMS, the MIS has no use. This relation will be understood in the two following subsections, where we explain the CMS and MIS in details.

2.1 – The CMS Component

Based on the study done already [1], we propose CMS that is responsible for directly interacting with the student and capturing every action done. Its main functions include presenting the contents of a course and supplying the necessary infrastructure, as for example, some communication tools, providing the interaction among students and tutors. Besides, it supplies all the necessary tools for user and course management and implements an authoring tool for the content generation.

Two environments make up the Course Manager System: the Course Environment and the Administrative Environment.

The Course Environment interacts directly with the students and its main function is to present the course content and provide communication tools. The Administrative Environment has all the necessary procedures for user management, course management, tests and content generation. Below we described each one of these comp4nents of both environments.

Course Environment*:*

- **Content Presenter**: It is responsible for presenting the course contents to the student and providing easy access to the pedagogical material.
- **Communication Tools**: It contains tools for the interaction of the students such as forum, chat and internal e-mail, etc. These tools enable the students to work in groups, exchange information and solve problems with other students or with the tutors.
- **Bulletin Board**: This tool enables the tutor to make notes about the course, schedule chat meetings and so on.
- **Test Presenter**: It is responsible for dynamically presenting the evaluation forms with the questions proposed by the author and storing the student answers.
- **Notes Block**: This tool enables the students to make online notes about the course content. This way the students can easily remember important points about the course content and quickly access the notes.

Administrative Environment*:*

- **Test Management**: It enables the tutor to create the tests.
- **Course Management**: This component enables the tutor to create a course or share pedagogical material with other tutors.
- **Tutor Management**: It is responsible for managing the tutor data and his permissions in each course.
- **Student Management**: It is responsible for managing the students subscribed in the courses and generating reports about the student performance.
- **Authoring Tool**: It is directed to authors in order to create pedagogical material for a given course.

It is important to notice that CMS does not take into account the previous knowledge and experiences of the students. All the students in a course follow the same content sequence. However, the particular abilities, the preferences related to the content presentation and the intellectual capacities of the students must be considered in order to make the learning process more effective. In addition, in the CMS the student is not closely guided and evaluation results cannot be presented immediately. All of these problems are present in the most existent web-based education system. In order to solve these problems, we propose an approach based on the multiagent intelligent systems and ITS paradigms. The next section contains a detailed description of such system.

CMS is already implemented and working properly. Figure 1 shows a CMS screenshot from the the Administrative Environment.

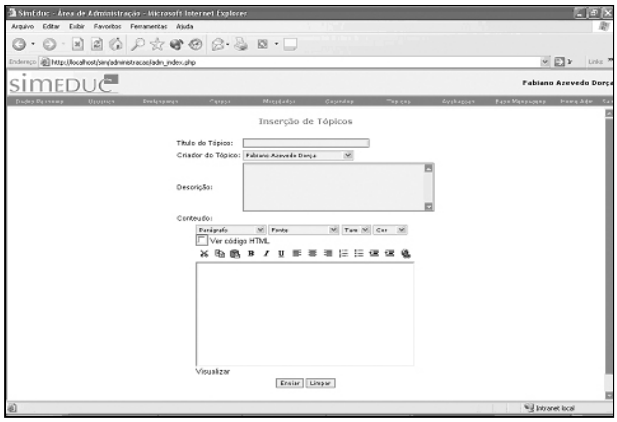

Figure 1 – CMS screenshot from the Administrative Environment

2.2 – The MIS Component

The MIS depicted in the Figure 2 provides adaptativity and intelligence to the CMS through the introduction of intelligent agents. This component is responsible for the whole pedagogical processing, curriculum sequencing, diagnosis of learning problems, knowledge evaluation, behavior evaluation and help the students. These are high complexity tasks and usually demand a certain degree of concurrency, continuous execution, decentralization of the information ownership, decentralization of tasks execution involved in the process, autonomy and cooperation between the agents. In this context the MAS application has great value [2].

As it can be seen in the Figure 2, the MIS implements all the modules that make up an ITS - the Knowledge Module (Knowledge Domain Database), the Pedagogical Module (Pedagogical Database, Pedagogical Agent and Evaluation Agent), Expert Module (Expert Agent), Communication Module (Assistant Agent and CMS) and the Student Model (Student Model Database).

According to Figure 2 the Assistant Agent receives information from the student (1) and sends it to the Evaluation Agent (2). In order to evaluate a problem solved by the student, the evaluation agent can request help from the Expert Agent (3). As a result of this interaction, the Student Model is updated (4) and this is communicated to the Pedagogical Agent (5). Based on the Student Model, the Pedagogical Agent can update the student curriculum (6) with the proper content (7). This process was accomplished by instructional planning techniques [3].

Our approach takes into account a high specialized division of the tasks involved in the teaching-learning process in order to improve the efficiency of the message passing, minimizing the traffic and taking the maximum advantage of the distributed processing provided by MAS.

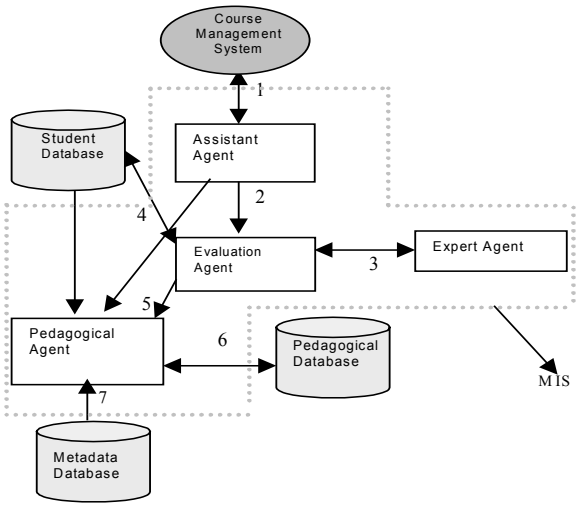

Figure 2. The Multiagent Intelligent System

3. THE KNOWLEDGE REPRESENTATION AND THE STUDENT MODELING

In order to implement the Pedagogical Agent, an efficient knowledge representation was necessary. Our proposal to the knowledge representation was inspired in ABITS [4]. In this proposal, the knowledge module is composed by two parts, the Knowledge Database and the Metadata Database.

The Knowledge Database contains all types of instructional material that can be presented to the student. This database may be centered in one server or spread in some servers over the Web. In this way, the Knowledge Database includes the course contents that are stored in formats that can be visualized with a web browser. In addition, the Metadata Database stores metadata information about the contents in the Knowledge Database.

The Metadata Database contains the information necessary to index the instructional material, attributing meanings to them and setting the dependences between them. These information will be used in the adaptation of the curriculum to be presented the a particular student.

The Pedagogical Agent needs pedagogical information such as pedagogical type, interactivity type, interactivity level, semantic desity, educational environment, duration, dificulty level, user age and learning time to each learning object to be used in the curriculum generation. So, we use the LTSC LOM (Learning Object Metadata) from IEEE [5].

In the Metadata Database the learning objects indexes instructional material that can be presented through the Web, as for example a lesson (page HTML), a simulation (Java applet),

a virtual world (vrml archive), a test (page HTML with an evaluation form) and other types of objects supported by the Web. A learning object must define the minimum set of properties necessary to allow the management, the localization, the evaluation and the correct use of the instrucional material represented by the object.

The adoption of the IEEE LTSC LOM was based on the necessity of an well accepted standard what brings the capacity of re-using ready materials available in the Internet. Another important characteristic of this standard is the capacity to store a large variety of pedagogical information, making possible the implementation of an efficient pedagogical processing.

It has been noticed that the Student Model is a very important ITS module because it provides information for the execution of all the functions previously mentioned (pedagogical processing, curriculum sequencing, diagnosis of learning problems, knowledge evaluation, behavior evaluation and help to the student).

The Student Model used by MIS is based on the Overlay Model [6] and uses Fuzzy Sets to represent the degree of knowledge acquired by the student in a certain time. The Fuzzy Sets are characterized mainly by flexibility, efficiency and relative simplicity of implementation when compared with other techniques. In addition to that, they achieve satisfactory efficiency [7]. Moreover, the Student Model stores the student preferences which allows the Pedagogical Agent to put the most appropriate learning objects into the curriculum.

The learning preferences store all the information about the student perceptive capacity, as for example, the kind of material the student is more receptive. The student learning preferences are stored as a pair Pref(type, value) where *type* indicates the preference type and *value* indicates the level of acceptance or the accept format. The student preferences are set before starting a course, through a form that is presented to the student in the course submission process, where another basic data are acquired, as name, profession, etc. During the course, the student learning preferences may be updated, according to the progress acquired by the student

The Metadata Database stores the pedagogical characteristics for each learning object. So, the appropriate learning object is chosen doing a comparison of the student preferences and these characteristics. The characteristics stored for each learning object are pedagogical type, interactivity type, interaction level, semantic density, educational environment, duration, difficulty level, user age, learning time.

4. THE MIS AGENTS

In this section we present a description of each one of the agents that constitutes MIS. Special attention is going to be given to the Pedagogical and Evaluation Agents. The role developed by the Assistant Agent is very simple. In fact it does the interface between the CMS and the other MIS agents. In the other hand, the Expert Agent is very complex and we propose it as future work.

The Assistant Agent is responsible for interacting with the student providing help and guiding through the sending of messages to the CMS. As a future work this agent can be incremented to autonomously execute tasks to the student and also stimulate the collaboration among the students, which brings great benefits to the learning process [8].

The main function of the Expert Agent is to provide some help to the Evaluation Agent during the evaluation of the student answers in practical exercises, as for example a physics problem, where the student must do some calculations using specific physics formulas. This kind of problem may need a more complex reasoning so the agent, if it finds a wrong answer, must be able to point where are the possible student mistakes so the Evaluation Agent could correctly evaluated which learning object was misunderstood.

4.1 – Pedagogical Agent

The Pedagogical Agent is responsible for the generation of a curriculum for the student and monitoring its presentation, interfering in the learning process when the student shows some difficulty in learning. To implement this functioning, the agent has been divided in two layers: the Reactive Layer and the Planning Layer. This division has been considered due necessity to divide tasks inside the agent.

The agent must be able to start the content presentation and monitor it during the process of update or generation of a new curriculum. In this way, the system integrates planning (responsible for the curriculum generation) and execution (responsible for presenting the curriculum and monitor its presentation), what provides efficiency gain and transparency to the user, that will not need to wait the complete generation of the curriculum. Moreover, the agent must react as fast as possible to any unexpected situation that may occur during a learning section. To implement this feature, the agent uses reactive planning techniques.

The curriculum presentation is not sufficient to guarantee that the student will have a satisfactory learn because the teaching environment is extremely dynamic. So, it is necessary the adoption of strategies to control the execution as the one implemented in the Reactive Layer. In this context, the presentation is monitored and can be modified when necessary.

The Pedagogical Agent proposed here extends ABITS [4] because it introduces the notion of external events. External events are related to the interaction between the student and the system. Examples of these events are violation of time restriction and evidence that the student is no longer concentrated. The Pedagogical Agent internal structure is showed in Figure 3. All the details about the used techniques and implementation of the agent can be found in [3].

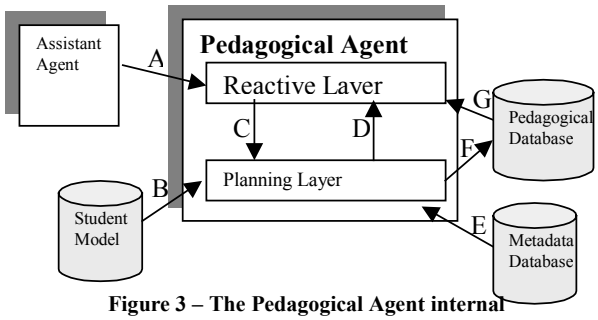

Figure 3 – The Pedagogical Agent internal architecture

In Figure 3 we have:

A – Information about the interaction between the student and the system.

B – Preferences and knowledge degree **C** – Asks for the plan

D – Plan **E** – IEEE Learning Objects **F** – Plan

G – Reaction plan

The curriculum sequencing, which is the main activity of the pedagogical agent, is achieved by using a content planner and a presentation planner. The content planner is responsible for defining and ordering the contents to be presented to the student. Our approach is based on hierarchical planning [3] and defines abstract and primitive operators. Also, an important feature in our work is the fact the external events are considered during planning time. After the generation of the content plan, a presentation plan is achieved. The presentation planner, based on the student model, decides how to present the content to the

student. More details about the curriculum sequencing can be found in [3].

4.2 - The Evaluation Agent

The Evaluation Agent is responsible for evaluating the progress of the student during the course. As a result of this process the Student Model is updated.

After carefully analyzing the aspects to be considered related to the performance of the student, some characteristics such as the time spent in content, time for resolution of each question and level of rightness in each question were considered very important to the evaluation process.

Moreover, it is necessary to analyze the student behavior during learning sections, as for example, the movement of the scroll bar, the participation in chats and forums, the number of questions posted to the teacher, etc. Based on these requirements, the evaluation process implemented by the Evaluation Agent is showed in the Figure 4.

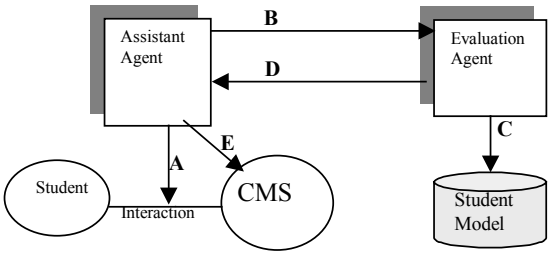

Figure 4 – The evaluation proccess

In Figure 4 we have:

A – Student monitoring **B** – Student information sending

C – Student model update **D** - Feedback to the student

E – Feedback to the student

The student monitoring process (A) is done by means of the CMS incoming messages analyses. The information acquired in (A) is sent to the evaluation agent (B) that updates the student model (C) and produces a feedback to the student (D), sending it to the Assistant Agent. Then, the Assistant Agent formats and sends the generated feedback to the CMS (E) that will effectively present it to the student. As explained before, the Assistant Agent may give immediate feedback in some cases.

The evaluation algorithm used in the Evaluation Agent is divided in two parts: the Knowledge Evaluation (KE) and the Behavior Evaluation (BE). Figure 5 shows the Evaluation Agent internal structure and other components related to it. Their description follows bellow.

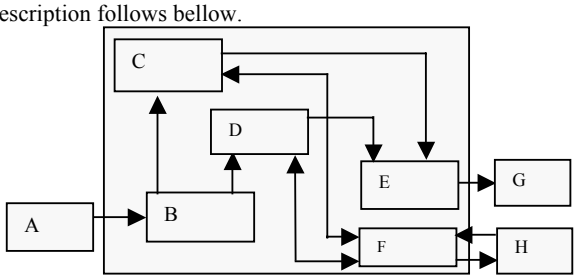

Figure 5 – The Evaluation

In Figure 5 we have:

A – The FIFO queue where the incoming messages are stored. It represents the agent *sensor* according to [9].

B – Perception Handler – It is responsible for reading the queue (A) and determines what to do with the incoming message.

C – BE evaluates the student behavior during the learning sections.

D – KE evaluates the student knowledge in a certain time.

E – Communication Module – It is responsible for sending messages to other agents. These messages include the feedback generated by the evaluation process that must be sent to the Assistant Agent.

F – Database Handler – It handles information related to the Student Model. It represents the agent *effectors* according to [9].

G – It represents the other SIM Agents.

H – Student Model Database.

Both BE and KE use Fuzzy Rules to evaluate the student. Initially, our system possesses three input linguistic variables, used by KE: **Answers**, **Time of Reply in Each Question** and **Behavior** (calculated by BE), and the resulting variable, **Performance**. Based on these linguistic variables, an example of one of these Fuzzy Rules is given by:

IF the **Answare** is *Corrrect*, the **Time of Reply** is *Reasonable* and the **Behaviour** is *Bad* THEN the **Performance** is *Satisfactory*.

The linguistic terms to **Answer** are *Wrong, More or less correct, or Correct*. **The Time of Reply** may be *low, reasonable or high*. The liguistic terms to **Behavior** are *Bad, Normal and Good*. Finally, the linguistic terms to the **Performance** are *Low, Regular, Satisfactory, Good, Excelent*. So far, the KE has 25 fuzzy rules.

In the student behavior evaluation, the following questions must be answered: What does characterizes the student behavior? What could modify the student behavior? Etc. Therefore, the input linguistic variables used by BE are **Time spent in a page**, **Scroll bar movimentation** and the **Participation in Chat and Forum.** The output variable is **Behavior**, as explained above .

The liguistic terms to **Time spend in a page** are *Fast, Normal, Slow.* **The Scroll bar movimentation** may have the values *Low, Normal High.* The **Participation in Chat and Forum** variable may be *Bad, Resonable or Good.* According to [10], the student participation in forums and chats must be evaluated because it allows the system to infer how much the student is involved in the course and with the other students. The frequency of the student in this kind of resource is quantified through the sending and receiving of questions. The BE has 27 Fuzzy Rules.

In order to transform these linguist variables in numeric values, some membership functions map the input variables in values inside the interval [0, 1]. These are the values to be stored in the Student Model for each Learning Object in the curriculum of the student. In addition, the Student Model stores the student learning preferences and personal characteristics so that the most adequate Learning Objects can be introduced into the student curriculum.

5.IMPLEMENTATION ISSUES

The CMS is a common web application which consists of a set of scripts that provides dynamic pages to the students. CMS was implemented using PHP. Therefore, the application is accessible by any Web browser in any computational platform that offers Internet connection. Moreover, the CMS implementation provides asynchronous interaction with the students, without determining the time for the educational activities. The system databases have been implemented using MySQL.

MIS have been developed using Jade Multiagent Platform [11]. Jade complies with FIPA standards for intelligent agents [12], which is the most complete and used specification for multiagent system development. It specifies all the protocols, resources and services that must be provided and how they must

be used in the MAS context (for example, name services, yellow pages and communication protocols).

The MIS agents are started when the system is initiated. Although MIS and CMS make up a unique application, they can run on different hosts. It is possible because MySQL allows remote access, as shown in Figure 6 where H1, H2, H3, H4, H5 e H6 represent different hosts.

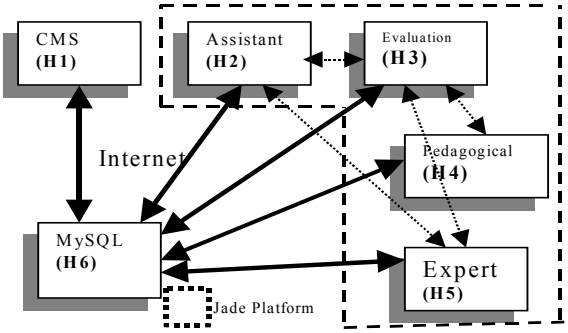

Figure 6 – MIS running on Jade platform

It is important to notice that each one of the agents could be run in a different host. This is possible because Jade is a distributed platform. Figure 6 shows how it would be. In this way, the agent platform could be split among several hosts and they could be connected through RMI (Request Method Invocation) provided by Java. Only one Java application, and consequently only one Java Virtual Machine is executed on each host.

The agents running on a platform are able to move to another host on the same platform. The agent code and state are serialized and taken to the new destination, where the processing is re-started from the same point. The MIS agents do not use mobility. In the future, as the system becomes more complex, it may be necessary.

Only one agent of each type is started in the platform. It is not necessary to have one MIS dedicated to each student because all the messages exchanged between CMS and MIS and among MIS agents requires the student and course identification numbers. This is sufficient to make the agents work well to many students. As the number of students grows up it may be necessary to start other agents so they could distribute the tasks and improve the time of processing. This possibility should take into account some load balance techniques and it is not considered in this work due to the complexity level of this problem. So, it is proposed as a future work.

The communication between CMS and MIS has been implemented by means of a Blackboard System [13], which is made up by two distinct tables inside the system database. One of them stores the messages containing all the information collected by the CMS about the user and the other stores the messages containing all the feedback to be presented to the student (generated by the MIS). The CMS and MIS read the messages sent through the Blackboard System in a FIFO fashion. After reading it, the message is immediately deleted.

The tables that store the messages have two attributes: the message identification and the message content. The message identification is an integer type that is auto incremented by the system. In addition, we will never get two messages with the same identification because this attribute is unique. It allows the MIS and the CMS to correctly identify a message and to delete it. The other attribute stores the message content in string type.

The content of the messages sent by CMS to MIS is composed by some attributes, as for example, the student identification, the course identification, the learning object identification where a problem occurred, the time spent in a learning object, the frequency of use of the communication tools, the answers in a test and others. In general, the attributes in a message vary depending on the circumstances. So, the attributes are separated by a " | ", where the first value is the type of the message. So, the Assistant Agent can identify what message is being received and what attributes it brings. All the agent has to do is to tokenize the string using the character " | " as parameter. Java provides efficient tools to do it easily. An example of this string may be: "1|152|231|99|100|5|2". The first number indicates that the message is about a learning section ending. In this case, the subsequent values indicates the student identification, the course identification, the learning section identification, the time spent in the learning section (seconds), the number of questions posted in the chat and the number of questions posted in the forum.

The content of the messages sent by MIS to CMS is composed by a unique attribute: the feedback to be present to the student. The feedback is a message with the intent to help, alert, guide, congratulate or call the student attention to some problem.

A good example of communication between MIS and CMS would be a situation where the Assistant Agent sends a message to the Evaluation Agent containing the student behavior in a learning object. The Evaluation Agent receives the message, Evaluates the student and send a message to the Assistant Agent with the evaluation result. So, the Assistant Agent search for the appropriate answer to the student and sends a message to the CMS containing the feedback to be shown to the student. The CMS receives it and shows the feedback message generated by the Assistant Agent through a pop-up window. The possible feedback messages to the student behavior are based on the linguistic variables describing the student behavior, as previously mentioned. So we can have 3 different feedback messages to the student behaviour, one for each linguist variable.

After the CMS-MIS integration we got the architecture shown in Figure 7. The Administrative Database contains necessary information about user identification (student, tutor or auxiliary tutor). Moreover, this database stores the responsible tutor for a course, the auxiliary tutor and the tools that can be used in the course (chat, forum, internal e-mail, student's note block, bulletin board).

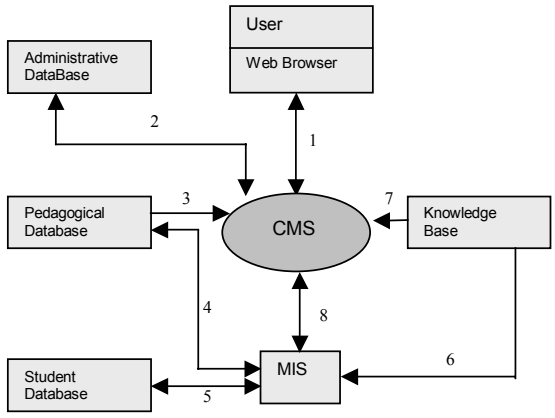

Figure 7 – System Architecture

As it can be noticed in Figure 7, the student interacts with the CMS receiving and sending information through the Internet protocol using a web browser (1). The CMS needs to access the information inside the administrative database in order to verify the users login and the tools available to be used in a course (2). In addition, the CMS uses the content sequences stored in the

pedagogical database in order to present the course content accord the student preferences (3).

The MIS component accesses the content sequence of a student in order to modify it and correct the learning problems that occurred during a course (4). After the modifications, the updated content sequence is stored in the pedagogical database. The student database needs to be accessed in order to detect learning problems in the student profile and update the content sequencing properly (5). It updates the student model regularly, as the student goes along with the course. The MIS uses the knowledge base to correctly evaluate the student (6).

The CMS component retrieves the content to be presented (7), according to the pedagogical database. We can see that MIS introduces the desirable characteristics for a web-based distance education system since it allows adaptability and intelligence.

6. RELATED WORKS

A reasonable number of adaptive Web-based tutoring systems are available. Some examples of such systems are CALAT [14], ELM-ART [15], PAT-OnLine [16], Tapejara [17] and ABITS [4].

Our work was particularly inspired on the ABITS architecture. ABITS is like MIS, a module that aggregates intelligence to a CMS. We both share the idea that is possible to extend the most used web-based educational systems, which are not considered intelligent systems, by incorporating into it an intelligent multi-agent component. However, different from ABITS, our Pedagogical Agent is more sophisticated in the sense that it uses techniques of artificial intelligence planning to generate curriculum sequencing and to support an interesting feedback to the user.

The ABITS and CMS are totally separated and the communication among them occurs in a unique way – from CMS to ABITS. The Spooler Agent implements this communication. When the CMS needs a service from ABITS it calls the service though calling a method from the agent. We preferred to use the Blackboard System to implement this communication because it is simpler and do not require any extra processing, what may improve the communication velocity. Moreover, the Blackboard System allows the communication in two ways, what is essential to CMS feedback receiving. ABITS was implemented using JAFMAS, Java-Based Framework for Multi-Agent Systems, which does not complies with the FIPA standards, what makes the system incompatible with other FIPA complying systems.

7. CONCLUSION

Our system presents innovative characteristics in relation to the adaptation of the course to the students and the evaluation of the teaching/learning process. Intelligent agents have been suggested as a promising application in the attempt of extending Web-based educational systems. The characteristics of this system make possible a large participation of the students in their process of learning.

We developed algorithms to curriculum sequencing and student evaluation that contributed to a satisfactory performance of the system observed in tests and simulations related to distance learning activities. The system has been tested with a group of students in graduation course. During these tests, lots of mistakes were corrected.

Future work includes the development of the Expert Agent, introduction of mobility, load balance algorithms and the increment of the Assistant Agent.

8. REFERENCES

[1] L. A. M. Zaina, G. Bressan, R. M. Silveira, I. Stiubiener, W. V. Ruggiero, (2001) "Analysis and Comparison of Distance Education Environments" - International Conference on Engineering Education August 6 – 10, Oslo, Norway.

[2] N. R. Jennings, P. Faratin, M. J. Johnson, P. O'Brien, M. E. Wiegand (1996) "Using Intelligent Agents to Manage Business Processes", Proceedings of First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM96), London, UK, 345-360.

[3] B. Queiroz, "Automatic Content Generation in a Web Based Education System", Master Thesis, Universidade Federal de Uberlândia, 2003 (In Portuguese).

[4] N. Capuano, et al., "ABITS: An Agent Based Intelligent Tutoring System for Distance Learning", *Proceedings of the International Workshop on Adaptive and Intelligent Web-based Educational Systems*, Montreal, Canada, 2000.

[5] W. Hodgins. Draft Standard for Learning Object Metadata. [S.l.], 2001.

[6] E. L. Ragnemalm, "Student Diagnosis in Practice; Bridging a Gap", Department of Computer and Information Science, Linkoping University, Linkoping, Sweden, Technical Report 1995.

[7] E. M. Valldeperas, "Sistema Bayesiano para Modelado del Alumno", Doctor Thesis, Departamento de Lenguajes Y Ciencias De La Computación. Universidade de Malaga – May / 2000.

[8] Y. Shang, H. Shi, S. Chen, "An Intelligent Distributed Environment for Active Learning", *ACM Journal of Educational Resources in Computing*, Vol. 1, No. 2, Summer 2001, pp. 4 – 17.

[9] Russell, S.; Norvig, P. Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence, New Jersey, 1995.

[10] Lee, Joice, Vieira, Heloísa (2002) "Avaliação formativa em ambientes de EaD. Artigo. Instituto de Informática" . Universidade Estadual de Campinas (Unicamp). XIII Simpósio Brasileiro de Informática na Educação – SBIE – UNISINOS 2002.

[11] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa, (2003) "JADE PROGRAMMER'S GUIDE", http://www.iro.umontreal.ca/~pift6802/Jade/doc/programmersguide.pdf, (Accessed in March, 2003).

[12] FIPA, "Foundation for Intelligent Physical Agents", http://www.fipa.org/. (Accessed in July 2003).

[13] G. Weiss, "Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence", MIT Press, 2001.

[14] K. Nakabayashi, et al., An intelligent tutoring system on the WWW supporting ubteractive simulation environments with a multimedia viewer control mechanism. Proc. of WebNet'96, World Conference of the Web Society, San Francisco, CA, October 15-19, 1996, p. 366.

[15] P. Brusilovsky, E. Schwarz, and G. Weber, ELM-ART: An intelligent tutoring system on World Wide Web, in Intelligent Tutoring Systems, Springer Verlag, Berlin, 1996, p. 261.

[16] P. Brusilovsky, S. Ritter, E. Schwarz, Distributed intelligent tutoring on the Web Proceedings of the 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August 1997

[17] Musa, D. L., "Um Sistema de Alertas Inteligentes para Ambientes de Ensino na Internet", Porto Alegre, maio de 2001 (Máster Thesis).