http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (5): 681-689, 2007
ISSN 1812-5638

An Agent-Based Design for the Learner Behaviour Modelling

[1]Chahrazed Mediani and [2]Mahieddine Djoudi
[1]Department of Computer Science, University Ferhat Abbas of Setif, Algeria
[2]Equipe IRMA and Laboratoire SIC UFR Sciences, University of Poitiers Bât. SP2MI, Téléport 2, bd Marie et Pierre Curie BP 30179 86962 Futuroscope Chasseneuil Cedex, France

Abstract: In the field of Information and Communication Technologies, the e-learning holds an increasingly important place. With the development of the web technology, the learner benefits from knowledge constantly updated. From now, the interest focus is the learner him self. But learners have various attitudes, that it will be necessary to model them in order to adapt systems and platforms to their needs. In this research, a design is proposed and implement an agent based architecture for the learner behaviour modelling, in the distance learning. This user centred system is conceived to build the learner profile or model, by collecting information concerning him on the one hand and memorizing his navigation traces in the learning environment, on the other hand. For present environment design, a multi agent based methodology is used. In this way, the environment is thought as being composed of entities having communication and problems resolution capacities and organized in its focus with the learner profile.

Key words: Learner modelling, intelligent agents, multi-agent systems, mase, user interface

INTRODUCTION

In the last few years, the technological evolution pointed out the need to introduce new techniques and features in Education. Faced to the interactive pedagogic means and to internet, the learning user and as for his study context, can be in a situation or he is winner and loser at the same time. On the one hand, he has a simple access to the numerical means which he needs for his activity, he leads direct experiments with his environment and he manages his time and his rhythm. On the other hand, the learner's isolation, the loss of perception, of vision and immediate advice of the teacher, make the learner much more divested than in the class situation.

For students to feel involved in their study, they must have an individual assistance assured by the teacher human (the tutor). But so that the tutor can follow and analyze the activity of each learner, he owes last a very long time.

Recently, Considerable research works have been made in the area of automating the knowledge acquisition and distribution. There are several different e-learning management systems, for example, WebCT (2001), Ganesha (2007) and Claroline (2006) etc. the common point between all these systems is that they offer passive and static sets of services. In these systems, the users are logged and their actions are recorded in multiple log files. These log files provide a very consequent material, easy to collect but more difficult to analyze. The time spent by the tutor to follow and to analyse the student's activities is much more than the one spent in the class situation. Instructors are expected to check learners log files, grade books, etc. They do not have the time, budget, or technical skills to build true instructional underactivity to their on-line learning programs (Al-Sakran, 2006).

The objective of this study is define and implement an agent based architecture making it possible, on the one hand, to collect information (subjective and thus not automatic) of the user learning through questionnaires. These questionnaires can or must be filled by the learner before and after the access to a distance learning platform on Internet. They contain information on the learner attitudes, his state, motivations, etc. On the other hand, to keep the detailed history of all the learner actions at the time when he reaches a learning platform on Internet (pages, protocols, communication with others learners or teachers, time). The collected data are stored in what is called the learner model to be easy treated and exploited later by other agents.

Thus the contribution of present study is the construction of the learner model, that its analyse is very useful in several research orientations, for example, the learner assessment field (Zidat and Djoudi, 2006), the intelligent tutoring systems field (Talhi *et al.*, 2006), as it can be very useful in the field of the learning content adaptation according to the learner's preferences and needs (Behaz and Djoudi, 2005).

Corresponding Author: Chahrazed Mediani, Department of Computer Science, University Ferhat Abbas of Setif, 19000, Algeria

Information Technology skills have the advantage of being observable by computers. Therefore, when users interact with a computer, they provide a great deal of information about themselves (Kabassi and Virvou, 2003). Agents have been quite successful at observing users behaviour and therefore, they have been widely used in learning environments in order to capture the users characteristics and perform user modelling tasks (Belkada *et al.*, 2001; O'Riordan and Griffith, 1999). Software agents play an important role in human-computer interaction and in the coordination of the internal processes of the system (Aroyo and Kommers, 1999).

For present environment design, we chose the Multi agent System Engineering methodology (MaSE), which was developed by Deloach *et al.* (2001) within the artificial intelligence laboratory of the Air Force Institute of Technology. In this way, the environment is thought as being composed of entities having communication and problems resolution capacities and organized in its focus with the learner profile.

LEARNER MODELING

Present study belongs to the learner modelling field, which aims at the creation of a cognitive and affective model from the observation of the learner behaviour to the learning environment interface (Mediani *et al.*, 2005). A learner model is a computer-based data management component or system that contains information about a person's learning activity. It typically forms a part of a larger system such as a learning management system or an intelligent tutoring system (Tanimoto, 2005).

This model must represent the learner profile, his goals, his plans, his actions, his beliefs and his knowledge. And must be used there after to explain why a learner can not complete his training work correctly and to intervene during the problem resolution process. Jhon Self was the big defender of the learner model and he showed that a relatively vague learner model could be useful in the e-learning environment (Dimitrova *et al.*, 2001). There are several aspects which characterize the learner model, the learner model can be: implicit or explicit, static or dynamic, specific, of surface or of deep.

For the proposed architecture, two types of the collected information are identified:

Static information: The learner must fill the questionnaire relating to this information on the subscribing level, among this information we identify:

The username, password, name, birth date, sex, addresses, e-mail, learner degree of motivation (high, middle, low motivation), type of media preferred (text, video, etc.), type of exercise preferred (traditional exercise, semi-assisted exercise, Question with Multiple Choices (QMC), the learning style of the learner (principle-oriented or example-oriented, general-to-specific or specific-to-general) (Chen and Mizoguchi, 2006).

Dynamic information: This information must be collected after the access to the learning platform, among this information we identify:

Degree of concentration (high, middle, low), degree of mastery of a certain topic (poor, fair, good, very good, excellent), the degree of learner interest in a particular topic, etc.

THE ENVIRONMENT DESIGN

The multi agents systems present like characteristics to allow the knowledge division or distribution and to make correspond a whole of agents and to coordinate their actions in an environment for the achievement of a common goal. Considerable research work has been conducted in using agent technology in the field of education, for example, Pedagogical agent (Jhonson and Show, 1997), teaching agents (Marin *et al.*, 2004), e-learning system (Al-Sakran, 2006).

In present analysis and design we chose the methodology MaSE, which starts from a very modest and pragmatic agent and multi agent system definitions: In this case a multi agent system is regarded as a whole of software processes which communicate between them to achieve a given common goal. The primary focus of MaSE is to help a designer take an initial set of requirements and analyze, design and implement a working multi agent system (DeLoach *et al.*, 2001).

MASE covers completely the software life cycle in the multi-agents systems. MaSE comprises seven steps divided into two phases. Each step has as a result, one or more diagrams.

The analysis phase includes three steps:

- Identification of the goals, whose result is the goal hierarchy diagram.
- Identification of the use cases, whose result is a whole of use cases expressed by one or more sequence diagrams using Unified Modelling Language (UML).
- The refining roles whose result is the roles model and a whole of concurrent tasks diagrams.

The design phase comprises four steps:

- The creation of the agent classes, whose result is the agent class diagram.
- The construction of the interactions between agents, whose result is a whole of interaction diagrams.
- Assembling the agent classes whose result is the agent internal architecture.
- System design whose result is the deployment diagram (using UML).

The analysis phase

The goal hierarchy diagram: The principal purpose of the goal hierarchy diagram is to identify and organize the system requirements into a structured set of system goals and sub-goals.

A learning environment is conceived to make it possible to each user to have access, in a simple way and through the networks, to the communication tools, services and numerical resources which he needs for his activity. Thus the principal functionalities which such an environment must offer to the learner are: to supervise the learner connections and disconnections, to manage his workspace and his communication space (chat in our case).

But the main objectives of present study are the collection of the learner information and the memorization of his navigation traces in the learner model. These two objectives were added to the principal functionalities of the learning environment.

So, we organized the objectives, which indicate the services that our system must provide in a goal hierarchy diagram as shown in Fig. 1. We also included there the data base saving and loading objectives.

Sequence diagrams: The sequence diagrams in MaSE indicate the interactions between the roles. For the whole of our use cases, we defined the following roles:

- **R_Interface:** It encapsulates the interface between the learner and the system. It is responsible for the interaction with the learner, presenting pedagogical contents to him. It also transfers the learner's actions/answers to the R_Assistant.
- **R_Assistant:** It encapsulates the tasks carried out by the learner agent, which consist in addressing requests relating to the knowledge (to load course, exercise, to save work, etc.).
- **R_Spy:** It encapsulates the tasks carried out by the learner agent, which consist in addressing requests to save the learner activities traces in the learner model (Mediani and Djoudi, 2006).

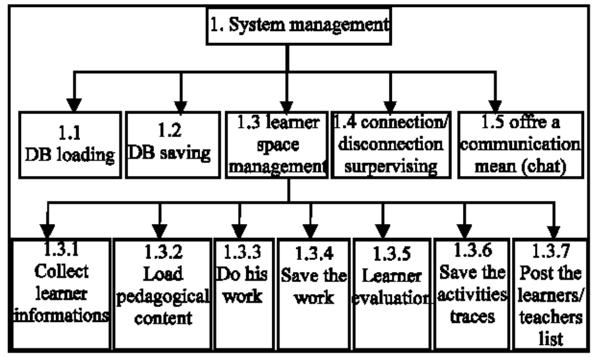

Fig. 1: Goal hierarchy diagram

- **R_Collector:** It collects the learner information's and saves them in the learner model.
- **R_Collaborator:** It accompanies the learner during his interaction with the others (learners or teachers) via the communication tools (chat in our case) and sends the interaction traces to the assistant role.
- **R_Evaluator:** It carries out the automatic evaluation of the QMC exercises by comparing the learner knowledge with those of the expert (teacher in our case) and sending the evaluation results to the assistant role which saves them in the learner model.
- **R_DB_Supervisor (DBS):** It encapsulates the tasks of the data base supervisor agent, which mainly consist in answering the requests coming from the teachers and learners agents relating to the knowledge (pedagogical contents, videos....). It is the only one to make transactions with the data base.
- **R_DB:** It encapsulates the interface System-DataBase (DataBase is the data base which contains the pedagogical content).
- **R_Learner_model (LM):** It encapsulates the interface System-LMD (LMD is the data base which contains the learner model).

As an example, we present in the following figures two sequence diagrams describing the two use cases: learner connexion (Fig. 2) and loading course or exercise (Fig. 3).

Use Case learner connection: When the learner connects, an Interface agent is created, this last sends the user name and the password to R_Assistant which in its turn sends them to the data base supervisor role for checking. The result of this checking will be transmitted to R_Assistant then to the Interface. If connection is established, R_Assistant informs R_Spy which saves the trace in the learner model.

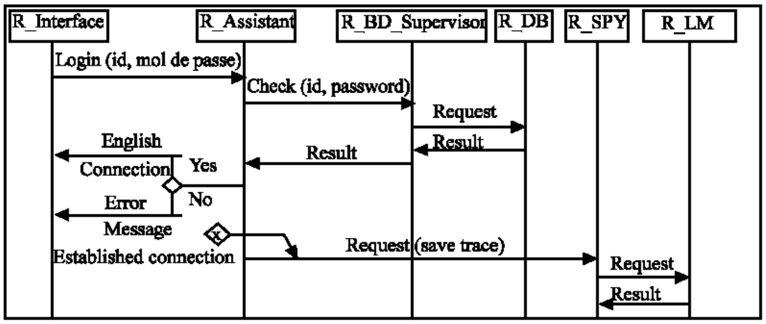

Fig. 2: Use case learner connection

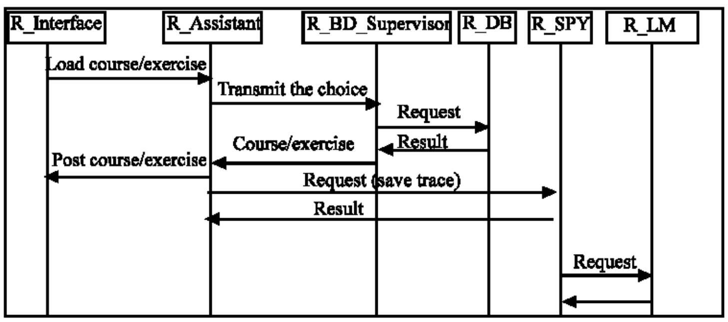

Fig. 3: Use case load course/exercise

Use case load a course/exercise: The learner informs his interface agent that he wishes to begin the study; the interface agent advertisements R_Assistant of the student choice, this choice will be transmitted to the data base supervisor which sends the course or the exercise wished to R_Assistant then to the interface. R_Assistant informs R_Spy which saves this activity in the learner model.

Roles and tasks diagram: The objective of this step is to transform the goal hierarchy diagram and the sequence diagrams resulting from the various use cases, to a roles diagram associated with the concurrent tasks. The tasks are independent processes that the role must carry out to achieve its goals.

The interactions inter-roles are drawn by communication ways between the tasks. To each goal or sub-goal of the goal hierarchy diagram is assigned a particular role. MaSE methodology indicates that each goal must be assigned to at least one role. In the following diagram Fig. 4, we identify the whole of the roles and the associated tasks for our environment.

Concurrent task diagrams: The goal of this step is to specify the various tasks and the roles transitions in the realization of a task. We show in the following diagrams the operation of a request for service formulated by the roles transmitting the request (Fig. 5) and the way in which the service supplying role carries out it (Fig. 6) and this by using two finite state automatons.

The design phase: The passage between the analysis phase and the design phase is done as follows:

- The designer creates the agent classes and assigns each role with at least one agent class.
- The communications between tasks determine the conversations between the agents.

The agent class diagram: With regard to present system, we directly transformed our roles into agent classes, i.e., we have affected each role of the preceding step to a different agent class, except for the Assistant agent which plays three roles. The methodology recommends noting in each rectangle representing an agent class the roles that it fills (Fig. 7).

For present application, we thus have seven agent classes:

- Interface (which fills the interface role).
- Assistant (which fills the Assistant, Spy, Collector roles).
- Evaluator (which fills the evaluator role) is the agent which evaluates the learner exercises (QMC) and sends the evaluation results to the Assistant agent.
- Collaborator (which fills the collaborator role).
- DB_Supervisor (which fills the R_DBS).
- Inter_DB (which fills the R_BD).
- Inter_LM (which fills the R_LM).

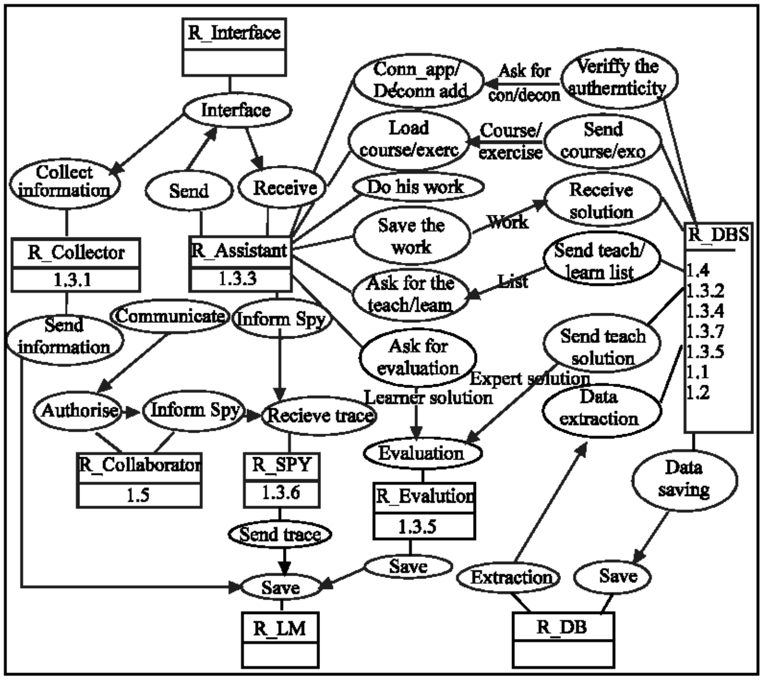

Fig. 4: Roles and tasks diagram

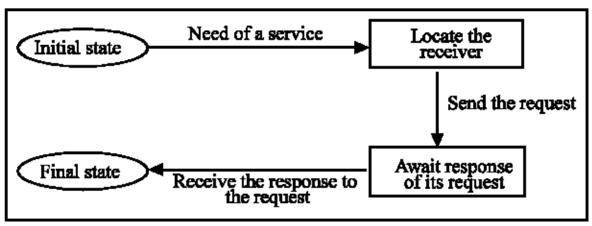

Fig. 5: State transition diagram of a role asking for a service

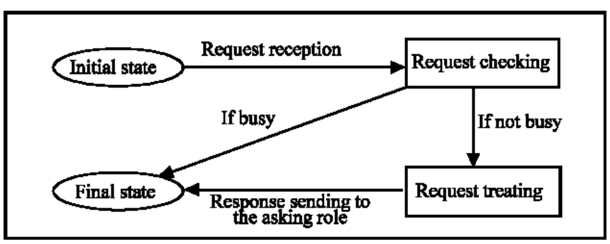

Fig. 6: State transition diagram of a role supplying a service

The construction of the inter-agents communications: The purpose of this step is to build the detail of the interactions between agents. These interactions are specified by using finite state automatons according to the same principle as the specification of the concurrent tasks of the third step. For each communication protocol between agents, the designer must define two communication classes' diagrams: one for the initiator and the other for the receiver.

AGENTS IMPLEMENTATION

The multi agents design methodology MaSE does not define or impose any internal architecture of the agents implemented. This task is left to the designer. We chose the architecture model JAM to define the internal structure of our agents.

Model JAM was developed by Huber and his colleagues (Huber, 2001). Architecture JAM makes it possible to implement capacities of mobility; it is built around a BDI (Belief-Desire-Intention) core. Each agent JAM is composed of five primary components:

- **A world model:** World model is a data base representing the beliefs of the agent.
- **A plan library:** Is a whole of plans which the agent uses to reach its objectives.
- **An interpreter:** Is the agent's brain, it allows the agent to reason on what it should do, when and how.
- **An intention structure:** is an internal model of the current agent goals, it keeps track of the agent engagement and progression in the achievement of its goals.
- **An observer:** Is a declarative procedure specified by the user which allows the agent to connect the steps of a plan in order to fill simple and periodic functionalities.

Agent JAM functions by using execution and behaviour semantic, it is a combination of UMPRS and SCS: the changes in the world model and the appearance of new objectives allow to start a reasoning process in order to seek the plans to be applied to the new situation, the interpreter selects a plan from the applicable plans list by making a reasoning of meta level or maximum utility.

DESCRIPTION OF THE LEARNER SCENARIO IN OUR ARCHITECTURE

According to Fig. 8, when a learner is connected to the application via Internet and by specifying a user name

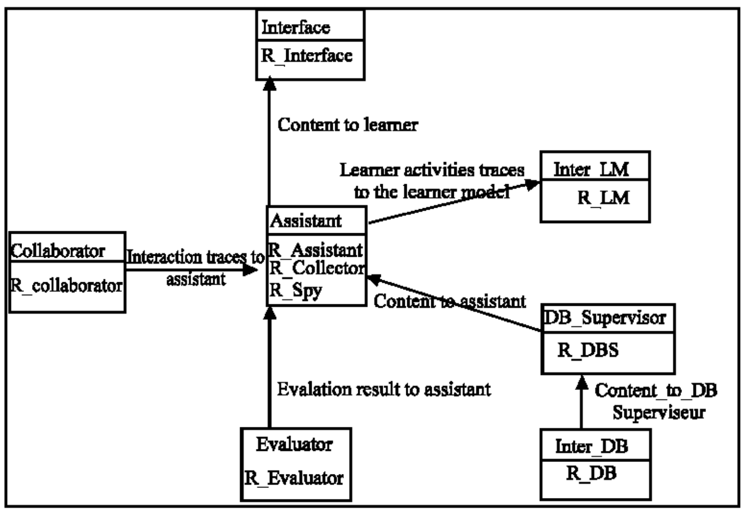

Fig. 7: Agent class diagram

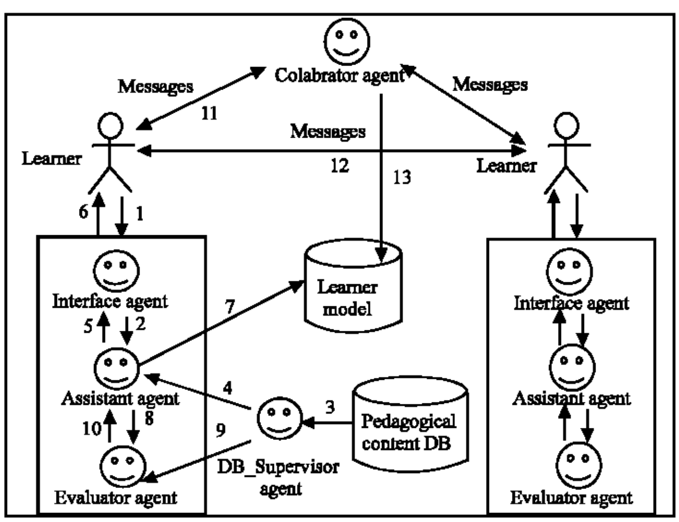

Fig. 8: Multi agent architecture

and a password, an Interface agent is him sent to supervise his activities. The Interface agent gathers information relating to the learner profile (1) and sends it to the Assistant (2), the DB_Supervisor agent gives access to the data base (3) which contains the pedagogical contents (courses and exercises that the learner should work on). The learner can choose among the courses and exercises list, the one on which he wants to work (4, 5, 6). The Assistant agent saves all the learner activities traces in the learner model (7). Once the learner thinks have ended his work, he can ask for an evaluation. The request of the evaluation is transferred by the Interface and Assistant agents to the Evaluator agent (8). To evaluate the problem solved by the learner, the Evaluator can request help from the DB_Supervisor agent (9) which has access to the contents base (3). And the result of this evaluation is transferred to the Assistant agent (10) which will modify the learner model (7).

If the learner wishes to communicate with his pairs (learners, teachers) the Interface agent sends a request to the collaborator agent asking him the learners and teachers connected list (11). An interaction can thus lead between the members of the class (12) under the supervision of the collaborator agent which updates learner model (13).

INTERFACE AND EXPERIMENT

According to Fig. 9, which show the learner interface, the Course tab gives access to the module contents and shows the list of the chapters and courses which it contains. The Exercise tab makes it possible to offer to learner a workspace enabling him to solve his exercises. There are several tools for evaluation. In our system, we use two types of exercises: QMC (automatic evaluation by the system) and traditional exercises (there the learner evaluation is carried out by his teacher). In bottom, we find a communication space between the system human actors.

On Fig. 10, we find the activities traces of one learner. On Fig. 11, we find an example of launching an intelligent agent following the architecture JAM which is the learner assisting agent.

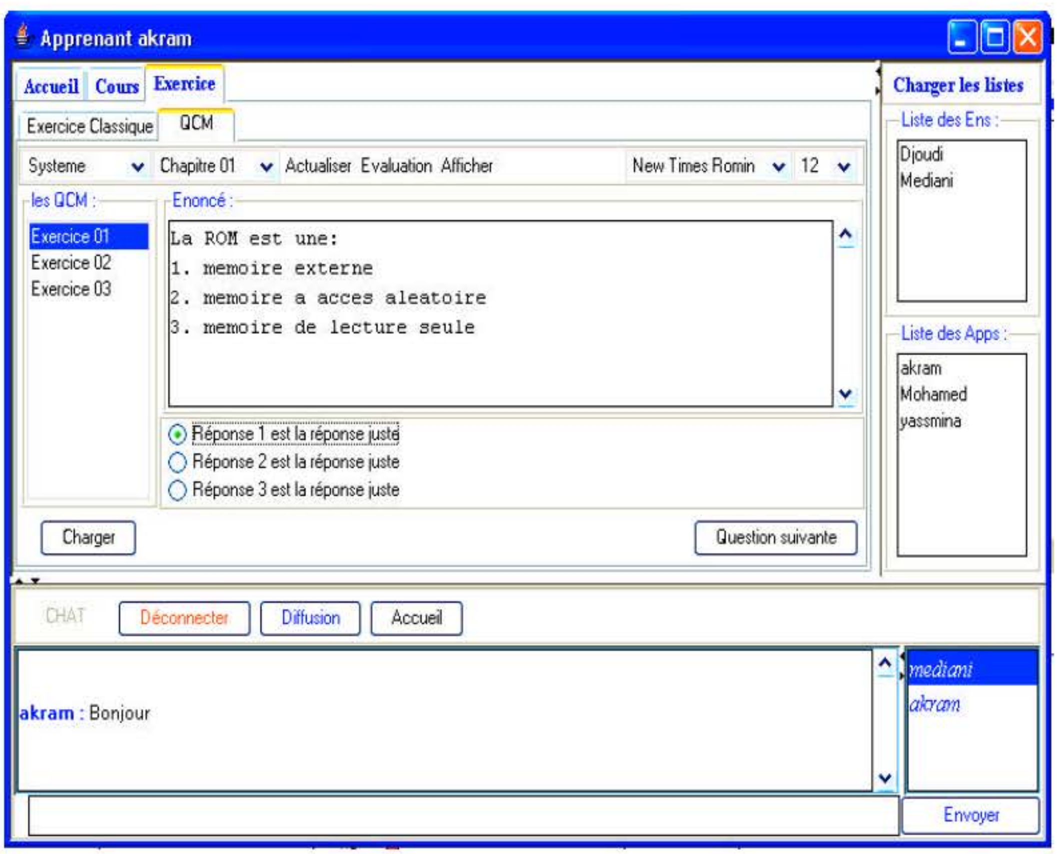

Fig. 9: Resolution exercises interface

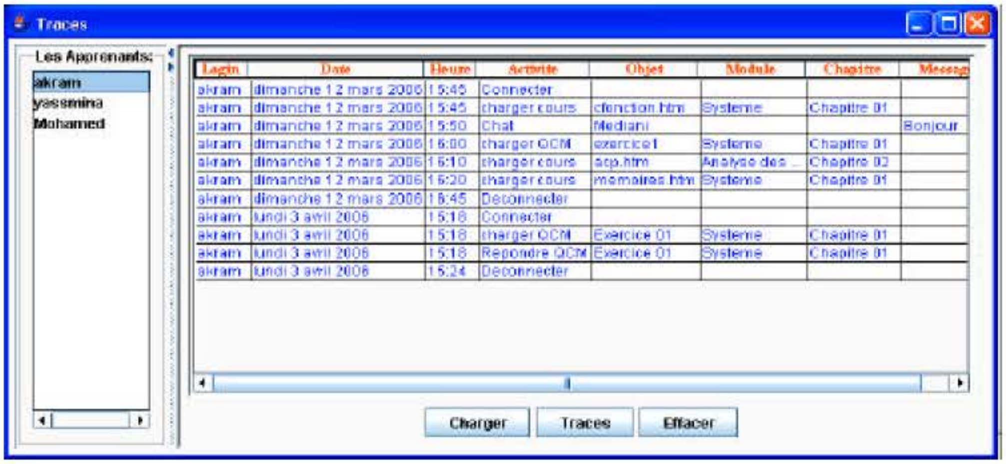

Fig. 10: Learner activities traces

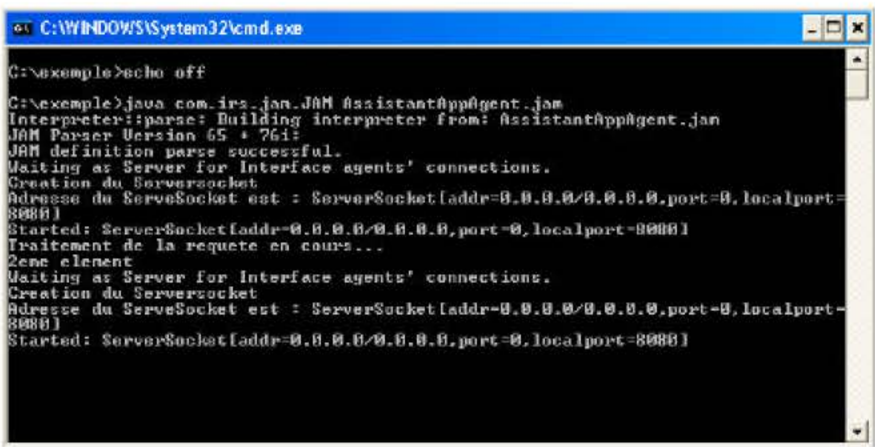

Fig. 11: JAM assistant agent

CONCLUSIONS

Through this study, we had recourse to the agent paradigm to propose an agent based architecture for the learner behaviour modelling who is in a distance learning environment. This system was analyzed, specified and designed according to the multi-agent system development methodology MaSE.

MaSE enormously helped us to model our system thanks to the simplicity of its concepts and the use of certain diagrams of UML.

Therefore, one of the contributions of this research is the construction of a learner model by the observation of his behaviour to a learning system interface and by taking into account his actions at the time of his work realization. This information is collected and stored in a user model in order to be further processed by a traces analyser agent. This last will analyse the learner activities (realized and not realized), the pedagogical resources consulted by the learner (access number, percentage of reading, consultation duration) and his communications (a number of received messages, a number of message sent, a number of messages read, the type of message, the transmitter of the message).

All this information will help the tutor to evaluate the learner, to know if the learner succeed in the resolution of his work or he has difficulties in his learning and thus he needs intervention from the tutor to assist him.

The results are also expected to be useful for personalized content developers, who adapt the environment's functionalities to learners needs.

REFERENCES

Al-Sakran, H., 2006. An agent-based architecture for developing e-learning system. Inform. Technol. J., 5: 121-127.

Aroyo, L. and P. Kommers, 1999. Preface-Intelligent agents for educational computer-aided systems. Special Issue on Intelligent Agents for Educational Computer-Aided Systems. J. Interactive Learn. Res., 10: 235-242.

Behaz, A. and M. Djoudi, 2005. Dynamic generation of adaptive hypermedia's documents in a numerical environment of work. Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées (ARIMA), 3: 25-53.

Belkada, S., A.I. Cristea and T. Okamoto, 2001. Measuring knowledge transfer skills by using constrained-student modeller autonomous agent. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'01). IEEE Computer Soc., pp: 375-378.

Chen, W. and R. Mizoguchi, 2006. Communication Content Onthology for Learner Model Agent in Multi-Agent Architecture. http://citeseer.ist.psu.edu /453995.html. Consulted Oct. 2006.

Claroline, 2006. http://www.claroline.net/ Consulted Sep. 2006.

DeLoach, S.A., M.F. Wood and C.H. Sparkman, 2001. Multiagent systems engineering. Int. J. Software Eng. Knowledge Eng., 11: 231-258.

Dimitrova, V., J. Self and P. Brna, 2001. Applying Interactive Open Learner Models to Learning Technical Terminology. In: User Modelling 2001: 8th International Conference, Sonthofen, Germany, July 13-17, Proceedings. Published by Springer Berlin, Heidelberg, Vol. 2109/2001.

Ganesha, 2007. http://www.anemalab.org/ganesha/ Consulted Jan. 2007.

Huber, M.J., 2001. JAM: A BDI-theoretic Mobile Agent Architecture. http:// marcush.net/IRS /irs_papers.html.

Jhonson, W.I. and E. Show, 1997. Using agents to overcome deficiencies in web-based courseware. Proceedings of the Workshop Intelligent Educational Systems on the World Wide Web. 8th World Conference of the AIED Society, Kobe, Japan, pp: 18-22.

Kabassi, K. and M. Virvou, 2003. Learner modelling in a multi-agent system through web services. Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03).

Marin, B. and Hunger *et al*., 2004. A Generic framework for an interface tutor agent within a virtual collaborative learning environment. IEEE International Conference on Advanced Learning Technologies (ICALT'04), Juensuu, Finland, pp: 31-35.

Mediani, C., K. Harbouche and M. Djoudi, 2005. The learner modelling in a collaborative learning environment. Congrès International en Informatique Appliquée, CIIA05, Algérie, pp: 27-32.

Mediani, C. and M. Djoudi, 2006. An Agent Based Memorizing Tool of the Learner Navigation Traces in Distance Learning. Colloque Euro Méditerranéen et Africain d'Approfondissement sur la Formation à Distance 3ème édition, CEMAFORAD3, Sousse, Tunisie, pp: 13-15.

O'Riordan, C. and J. Griffith, 1999. A multi-agent system for intelligent online education. Special issue on intelligent agents for educational computer-aided syst. J. Interactive Learn. Res., 10: 263-274.

Talhi, S., M. Djoudi and M. Batouche, 2006. Authoring groupware for intelligent tutoring system. Inform. Technol. J., 5: 860-867.

Tanimoto, S., 2005. Dimensions of transparency in open learner models. Proc. Int. Workshop on Learner Modelling for Reflection, to Support Learner Control, Metacognition and Improved Communication between Teachers and Learners, in Conj. with AIED'05, Amsterdam.

WebCT course Management Software, 2001. http://www.webct.com.

Zidat, S. and M. Djoudi, 2006. Online evaluation of Ibn-Sina elearning environment. Inform. Technol. J., 5: 409-415.