

Agent-Based Design for E-learning Environment

¹Khadija Harbouche and ²Mahieddine Djoudi

¹Department of Computer Science, University Ferhat Abbes of Setif, Algeria

²Equipe IRMA & Laboratoire SIC

University of Poitiers UFR Sciences Bât. SP2MI, Téléport 2, bd Marie et Pierre Curie
BP 30179 86962 Futuroscope Chasseneuil Cedex , FRANCE

Abstract: We presented an agent-based e-learning environment. Our aim was to allow many users to interact collectively and intelligently with the environment. In this cooperation model, human users and artificial agents carry out tasks in the learners' service. We define the internal structure of our kernel supposed to work within Internet/Intranet settings. Design was structured in three parts: individual learning space, collaborative space, and cooperative space. We advocate the employment of an agent-based approach, a suitable for two main reasons: agents were a natural metaphor of human acts, and the learning systems are generally complex. Prometheus methodology used for the design and emphasis placed on the agent-based features.

Key words: Artificial agent, human agent, agent-based design, e-learning environment, cooperation

INTRODUCTION

The incorporation of computer science and telecommunications has profoundly changed the implementation problems of learning systems. An e-learning environment is an individual and customizable implement allowing every user to have access, in a simple way, and across networks, to the communication tools, services and digital means, which he needs for his activity.

The implementation of such environment is becoming possible because of the development of the Internet / Intranet and the recent developments in Multi Agents Systems "MAS". In effect, cooperation is a very important characteristic of Multi Agents Systems where a distributed resolution of a problem or the sharing of tasks and results is the result of cooperative correlation between different agents.

Problems: Active commitment on behalf of learners to construct their own knowledge, as well as the processes of verbalization, clarification, and formulation, motivation, assistance, collaboration and cooperation are essential from a cognitive perspective [1, 2]. The creation and the implementation of a system that supports such criteria is complex due to the difficulty in predicting what is going to take place and of the necessity to make interceding specialists from very various disciplines.

The complicacy and the multidisciplinary of learning systems find a technological foundation in the development of computer networks, and emergence of agents-based systems. However, it raises new problems of cooperation and communication, among which, we can raise the protocols of communication between agents "human or artificial" of various kinds, the contents of correlations between computer agents and human didactic agents.

Objectives: We aim to design an agents-based learning environment. This environment has to:

- Take into account problems underlined by different forms of distance learning, such as: the sociological isolation of the learner, the loss of motivation and the learner independence
- Create necessary means to make of a distance situation, a reasonable equivalent of a face-to-face situation while allowing more suppleness in comparison with the constraints of time and space
- Propose means to attend and accompany learners and to replace the teacher during a working session
- Re-aim groups of learners in a productive direction "pay attention to the members left out of the correlation"
- Augment and favour correlation between different agents "human / human, human / artificial and artificial / artificial "
- Coordinate cooperative tasks between learners.

We will choose the suitable agent methodology allowing us to achieve our goals and resolving the agent-technology problems such as the human/agent co-operation and communication.

Overview of Proposed System: The architecture overview of the system allows capturing the agents' space in the system, the boundaries of the system and its interfaces in terms of actions and percepts, but also in terms of data and code that are external to the system.

The proposed distance learning system is dedicated to the management of both actors – students and teachers – while providing a structured and documented design. It is structured in three levels known as the three tiers architecture [3]:

- The user space;
- The application level;
- The storage space, which contains all the persistent data.

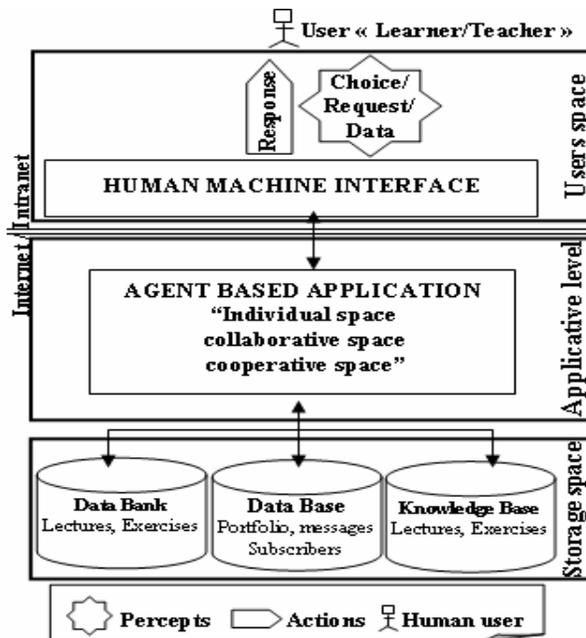


Fig. 1: Overall system architecture

ACTORS AND PASSIVE OBJECTS

An environment is a space representing the world in which the agents evolve. We differentiate in general between the agents who are active entities, and passive objects, which are located in the environment.

Defining the working environment of the human agents "actors" and artificial agents implies defining pedagogic, social, administrative, epistemic ... means and/or resources, similar or close to those unfolded in a face-to-face learning situation.

Human Agents "Actors": We have two main human actors: the learner and the teacher.

- The teacher, the tutor or the trainer: He assumes the tasks of a content tutor and those of an animator tutor, at the same time. He must assist learners, answers their queries by using the interne mailing, the chat or the forum, supervises their jobs " review, ongoing work, evolution ...", corrects all the exercises that a computer can not correct, updates the contents of lessons and/or exercises. The teacher will have, also, as task to supervise working groups: organization, animation, cohesion of working group, harmonization of communications, support the development of a feeling of membership in the group, notice and follow the evolution of the group and the evolution of the learner in the group.
- The learner or the student: He is the main actor, in the core of the learning process. He is the only one to be able to transform the information received into appropriate knowledge by: his own activity and by interaction with the other. He can exploit the different tools to consult lessons, solve exercises, communicate with other actors, and accomplish individual jobs or groups jobs:
 - Individual work: In this situation, the learner consults his course, solves the exercises and annotates his course. He asks for help neither to the teacher, nor to a co-learner.
 - Collaborative work: Here two situations arise:
 - Desired collaborative Work:* The learner request the assistance to another learner or to the teacher by the means of the chat. He annotates his course, places it at the disposal of everyone, and asks for teacher assistance to explain him the course or the exercise.
 - Imposed collaborative Work:* The learner belongs to a working group "WrGr", to whom a task is affected to be completed and thus he must collaborate with his Co-learners and communicates by using the forum.
 - Cooperative work: The various learners groups have to cooperate to carry out a total task.

The group: A group is a set of learners, who share certain aspects, interact, accept rights and obligations as

members of the group, share a common identity and work on a common task. We recommend a hierarchic structure close to that of the cooperative systems networks one^[4].

We consider the Virtual Class –VC-, which is a community of learners, as a big group, divided into sub-groups named Working Groups - WrGr. The structure of each group is centralized. Every member of the group has a very definite function within the group: the chief for example is the only member who has the right to make a decision after consultation of the other members, considered as subordinates. A global coordinator “teacher or artificial agent” supervises the global job.

We are in reality in a teamwork situation and sub-teams, but we prefer the appellation of class and working groups.

Pedagogic means “The portfolio”: The portfolio or the schoolbag is a log, a process, a step of learning and a formative valuation which helps, on the one hand, learners to learn and to progress^[3, 5], and on the other hand, the teacher to take measures to help the learner in order to ameliorate his learning^[6, 7, 8]. Among, the types of pedagogy which melt the practice of the portfolio, we have the cognitivism and constructivism^[8]. We find five kinds of electronic schoolbags:

- Learning portfolio “Learner file”: Contains jobs accomplished by the learner. This portfolio shows the progress of the learner in the course of the learning process.
- Presentation/communication portfolio “Group file, Learner work file and the election box”: Contains the productions that the learner likes to keep while annotating his opinion over to other pedagogic actors.
- Valuation portfolio “Valuation file”: Contain evaluations and judgments carried by the teacher on chosen jobs accomplished by the learner.
- Projects portfolio “Project file”: Progressive File gathering the state of progress, ongoing situation, coordination and correlations between different parts of a project accomplished by working groups.
- Teacher rack “Teacher file”: gather the whole of course and exercises proposed by the teachers with their standardized test solution.
- **Pedagogic objects**^[3, 5]: The learning process is divided into two steps. The first step consists in producing an explicit representation “written” of the knowledge. The second one consists in making an “encapsulation” of the knowledge in a world where the access to knowledge is not limited to the “reading” but in the construction of knowledge made by resolution of problems.

- Lectures: it is of course an essential element in any learning activity. The course can be annotated by the learner in order to share his assets with the others or be indexed at the beginning by the teacher.
- Evaluation tools: They allow bettering evaluating of the student, this one completely delivered to him-self. We can divide the evaluations into three categories: sommative evaluation “direct examination in face to face”, formative evaluation “during the course” and diagnostic evaluation “to have an idea on the cognitive level of the learner”. These tools arise in the form of projects and/or of exercises of various levels “simple, average or high” and in various forms: *The MCQs*: It is the simplest form of exercises. It does not require a human evaluator, the learner thus does not have withdrawal period between the end of the exercise solution and its correction. We have here: the alternate MCQs “to answer by Yes or No”, the elementary MCQs “only one answer is correct”, the simple MCQs “one or more choices are correct” and the MCQs of control” with a percentage of conviction”. The MCQs require an individual work of the learner.

Traditional Exercises « Problems »: it is the same kind of exercises that we find in a traditional learning. As the results are more complex to be analyzed and evaluated by a computer, it is necessary to resort to a teacher to carry out the correction. The learner can collaborate with other learners to resolve the problem.

Assisted exercises: These are traditional exercises whose solution is guided by the system, which proposes indications to the learner throughout the resolution process.

The project: It is a co-operative work between the various Working groups “WrGr” and collaborative and/or co-operative work within the same group.

Communication elements: We distinguish two basic models depending on whether communication is made between an individual I and an individual J, or from an individual I to a group. When we consider groups, these are necessarily, for our environment, closed and the nominal list of the individuals, which composes it, is known. To enter a closed group assumes a special procedure of amenity piloted by the coordinator or the chief of the group. A group can be that we called a «virtual class» or a working group. Communication can be made with confidentiality or without confidentiality; it can be asynchronous or synchronous. From these basic concepts, we will dispose of the following tools of communication:

Table 1: Communication models

Model	Realisation Possibilities	Correlation degree	Notes
I ↔ J	Interne mail asynchronous, confidential-Chat	- Relatively low	The interne mail, chat and forum cost is relatively low
	synchronous, confidential-Forum	- High	
I ↔ WrGr	asynchronous, non confidential	Normal	

AGENT-BASED DESIGN

Why agent? Defining agents is not straightforward. An agent is widely understood to be a software entity situated in an environment, autonomous, reactive to changes in its environment, proactive in its pursuit of goals and social [9, 10].

Whereas some characteristics cannot be used as determining factors since, they are grey shades of a scale that encompasses both objects and agents.

The Multi Agent Systems adapt well to the design of our learning environment because:

- The distance learning systems are open, dynamic and complex;
- Agents are a natural metaphor of human acts;
- The distribution of the data, control and the expertise is self-imposed;
- Agent has a high-level representation of behaviour.

Which methodology? To conceive our agent-based environment we will use the Prometheus methodology [10]. Prometheus is a Goal-Oriented approach, where roles are identified from the analysis of system goals. It, as a methodology, is intended to be able to support the design of BDI “Beliefs, Desire, Intention” systems, although it is not limited to such; all but the lowest level of design, leading into code, can be used equally well for non-BDI systems. The use of Prometheus was beneficial thanks to its usability, refinability, the Multi Agents System environment modelling and the agents-environment interactions.

System goals: As we saw above, the teacher has multiple and diverse tasks to assume, moreover he has to follow-up and communicate synchronously with learners. Since he cannot be ubiquitous, and he is exceeded and overloaded by his tasks, some situations require artificial agents to assist him.

The system, we are conceiving, has to:

- Interact with pedagogic users “learners and teachers”:
 - Display appropriate screens

- Get choices, requests and data

- Output results, questions

- Manage access and updating of the persistent information:

- Data Base “DB”

- Data Bank “DK”

- Knowledge Base “KB”

- Correct the MCQs;

- Assistance and orient the learner during an assisted exercise resolution session followed upon the request of the learner or the thinking time collapsing or a learner erroneous answer;

- Explain the course to the learner (maieutic explanation);

- Manage whole project topic realization:

- To propose a calendar of the tasks;

- To animate the dialogue and take care and motivate the learner left apart from the interaction;

- To Divide, pursuit and synthesize the project

- Ensure synchronous and asynchronous communication between members;

- To manage the meeting time taking account of the availability of the various actors;

- To form the Working groups and elect the group chief;

Cohesiveness and coupling: Selecting the agent types that will exist in our environment is maybe the most important decision that is made.

Prometheus takes account of two important metrics of the Software Engineering design: cohesion and coupling [5]. Ideally, each agent type should be cohesive and coupling between agents should be low.

First, we identify the functionalities: groups of coherent goals provided by the system. A useful tool for suggesting groupings of functionalities is the *data-coupling diagram* (Fig. 2). This depicts each functionality and each persistent data showing where functionalities read and write data.

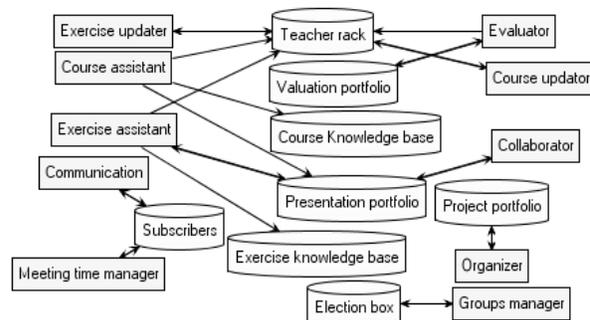


Fig. 2: Data coupling diagram

The artificial agents: These agents have to communicate with the users “actors”, make decisions,

assist learners, help teachers, consider and modify the users Database “DB”, access to the Knowledge Base “KB”... The result of the data coupling diagrams is the following Agents (Fig. 3):

- User Interface: This agent has to get, announce and return available information relating to the user needs “Percepts: request, choice, time out; actions: response and sending information”.
- The assistant “Companion”: It has to assist the learner, to orientate him in the resolution of the assisted exercises, to answer his questions directly and dialogue with him “subcontracting of the teacher”.
- The collaborator: It notices and diagnoses correlations between learners of the same group. To re-aim the group in a productive direction and pay attention to the members left except correlation and save the group session work “delegated by the teacher”.
- Communication: Manage synchronous/asynchronous, confidential/public communications, between the different dealers.
- Evaluator: It evaluates MCQs, returns result instantly and updates the learner valuation file “substitute of the teacher”.
- Cooperation: It facilitates co-operative work between working groups, and updates the project file “delegated by the teacher”.
- Scheduler: It finds suitable meeting time according to the groups chiefs disponibility, schedules meetings according to a preset or improvised planning “programs, cancels or defers”, inform the chiefs, and warns the absents.
- Updater: It updates the pedagogic contents “exercises and lectures”.
- Election: It forms the Working groups, updates the election box and returns the result of the election.
- Supplier: It performs access to the Database.

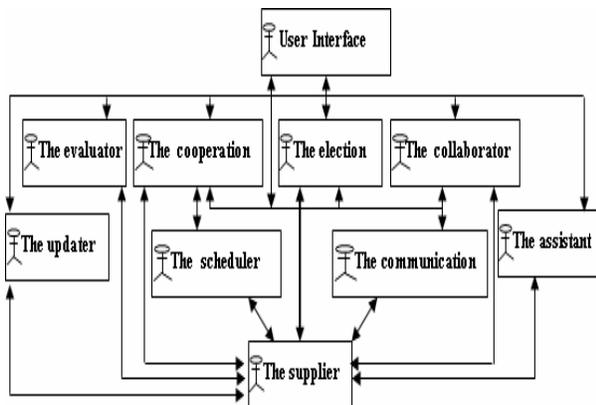


Fig. 3: Agent-types diagram

AGENT INTERACTIONS SAMPLES

The interactions between agents are developed using interaction protocols. Specifically, the notation used is the sequence diagrams of Agent-UML (AUML) [11]. We will present here two samples of interactions:

Exercise assistance: As soon as the learner loads a work on an assisted exercise, the assistant clock will be also launched. The assistant intervenes in one of the following cases: if the thinking time granted to the exercise is exceeded; if the provided answer is erroneous or at the learner request. Following a correct answer, the assistant validates and congratulates the learner. The learn-agent dialogue is maieutic.

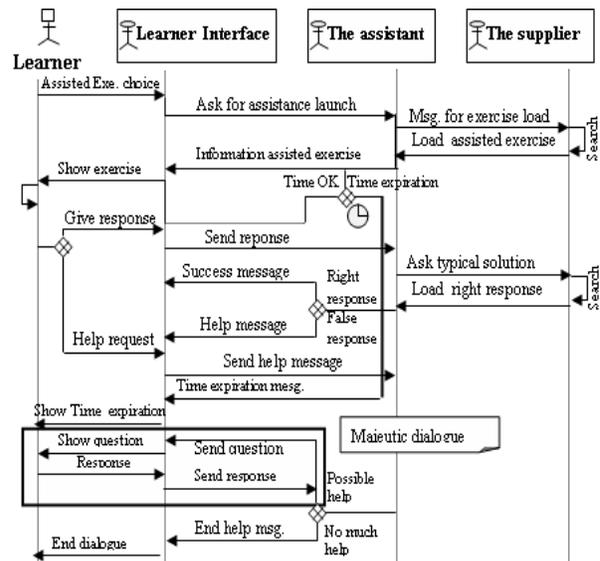


Fig.4: Exercise assistance protocol

Collaboration: In an imposed collaborative learning situation, the learner joins his group following a membership or identity check. The active learner communicates with his/her schoolmate and the collaborator. This one, having entered the time of entry of the learner keeps trace of the last participation. Every inactive learner is awaked by a message from the collaborator. At the end of the meeting, the group chief writes his report, which will be saved in the learning portfolio as well as the next meeting-time. The communication is stopped after having informed all the members.

Remarks: For each human actor “client post”, an interface agent is created. For whole the group of learners, only three agents are created “at the server post”: a supplier, a collaborator and a communication. Learneri, Learnerj and Chief Interfaces are instance of User Interface.

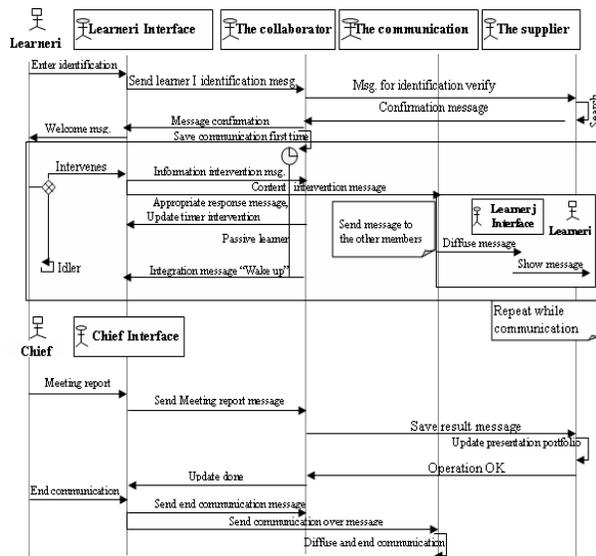


Fig. 5: Collaboration protocol

Implementation platform: We adopted for the design a goal-oriented methodology, where the design of how each agent achieves its goals is expressed in terms of plans, events, and data. At this point, the design process needs to make certain guesses about the implementation platform. In particular, agent should be implemented using a platform that supports plans that are triggered by goals. Such platforms include, but not limited to, JACK Intelligent agents or JACK [12].

JACK is an *Agent Oriented* development environment built on top of and integrated with the Java programming language. It includes all components of the Java development environment as well as offering specific extensions to implement agent behaviour. JACK has been developed to provide agent-oriented extensions to the Java programming language. JACK source code is first compiled into regular Java code before being executed. In agent-oriented programming, a system is modelled in terms of *agents*. These agents are autonomous reasoning entities capable of making

pro-active decisions while reacting to events in their environment.

Interfaces samples: Screens shown below represent some learner interfaces.

- Assisted exercise resolution: The learner can choose the exercise to be solved. A time of thinking is allotted to him. This time is variable according to the exercise and its degree of complexity. Four spaces are placed at the disposal of the learner. The first space allows displaying the body of the exercise. The second space is intended for the question/response between the companion and the learner. The third space makes it possible progressively to display the dialogue, which proceeded during the exercise resolution session. The fourth space is intended to the progressively helps shown to the learner.

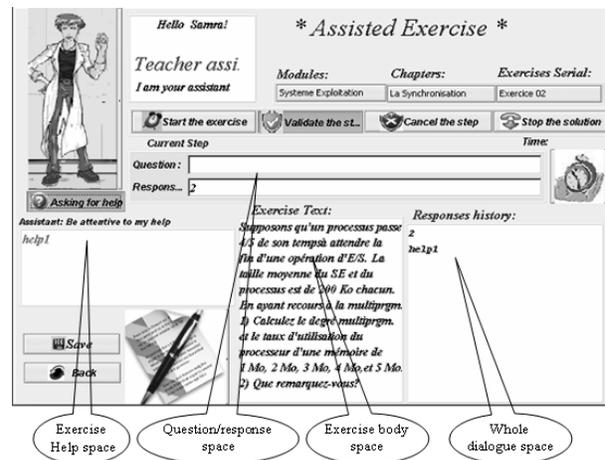


Fig. 6: Assisted exercise resolution

N.B: We have translated the interface above, from French into English, for this paper

- Lecture session: The learner has, here, four spaces. The first space intended for displaying the lesson. The second space for the indices, allowing the learner to seek the definition of a particular term. The third space is intended to the annotations. It allows the learner to annotate his course or to import annotations made public by the others learners. The last space is intended to the maieutic dialogue between the learner and the assistant in order to explain the course.

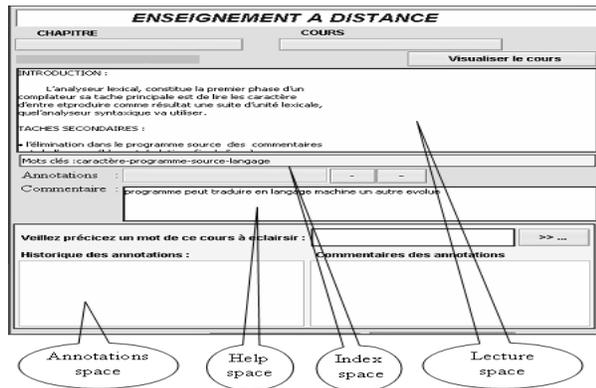


Fig. 7: Lecture session

Relationships: One of the thorniest points in any agent-based application is to depict the relationships between various agents “human or artificial” and the knowledge. These relationships allow determining when and how different agents have to pass the relay to each other’s. The following table (Tab. 2) describes the ten bipolar relations between passive and active entities of our learning environment:

CONCLUSION

We have presented an agent-based design for a learning environment using Prometheus methodology and JACK platform. The interne architecture is divided into three spaces “individual space, collaborative space and cooperative space”. It remains us to evaluate the system and study its impacts at different levels such a pedagogic, social and economic. Then this system will constitute the main tie in a long learning systems chain. Specifically it will serve as an input to an analyzer system, which will depict the epistemic and cognitive profile of the learner to adapt, thus, the learning according to his capacities and will.

REFERENCES

1. Jhonston, R., 1997. Distance learning medium or message. *Journal of Further and Higher education*, 21 (1): 107-122.
2. Tchounikine, P., 2002. Pour une ingénierie des Environnements Numériques Informatiques pour l'Apprentissage Humain. *Revue Information Interaction Intelligence*, 2 (1): 59-93.
3. Harbouche, K. and A. Hamdi-cherif, 2004. UML-Based Elearning Architecture. *Processding of the 1st Al-Baha Technical Meeting*, El-Baha, Kingdom of Saudi Arabia, pp: 34-45.

4. Faerber, R., 2002. Le groupe d'apprentissage en formation à distance : ses caractéristiques dans un environnement virtuel. In: Larose F & Karsenti T. (CRP Editions), pp: 99-128. Sherbrooke University.
5. Harbouche, K. and M. Djoudi, 2005. Environnement numérique d'apprentissage à distance à base d'agents. *Proceeding of Conférence Internationale sur l'Informatique et ses Applications*, BBA, Ageria, pp: 66-72.
6. Perrenoud, Ph., 1991. Pour une approche pragmatique de l'évaluation formative. *Mesure et évaluation*, 13 (4) : 49-81.
7. Goupil, G., 1998. *Portfolio et dossiers d'apprentissage*. Chenelière/Mc GRAW Hill, pp. 78.
8. Muller, F., 1999. Vers une évaluation personnalisée. *Apprendre, revue Sciences Humaines*, 98 : 30-33.
9. Luck, M., P. McBurney and C. Preist, 2003. Agent technology: Enabling next generation computing. *Agent LinkII*, pp: 5-10.
10. Padgham, L. and M. Winikoff, 2005. Prometheus: A Practical Agent-Oriented Methodology. In *Agent-oriented methodologies* (eds B. Henderson and P. Giorgini) pp. 107-135. Idea Group Publishing (IGP).
11. Odell, J., H.V.D. Parunak and B. Bauer, 2001. Representing Agent Interaction Protocols in UML. In: *Agent-Oriented Software Engineering: First International Workshop, AOSE 2000*, Limerick, Ireland, June 10, 2000, revised papers (Springer Berlin, Heidelberg) pp. 201-218.
12. Agent oriented software group, 2006. Jack around the world. www.agent-software.com/shared/home.
13. Sylvie, P., W. Carine and B. Nicolas, 2001. Baghera: ww-Baghera.imag.fr.
14. Bendella, F., B. Messabih, 2004. G++: A Platform of E-learning. *Information Technology Journal*, 3: 240-244.
15. Mbala, A., C. Reffay and T. Chanier, 2003. SIGFAD : un SMA pour soutenir les utilisateurs en formation à distance. *Environnements Informatiques pour l'Apprentissage Humain*, Strasbourg. <http://lifc.univ-fcomte.fr>
16. Deloach, S.A., M.F. Wood and C.H. Sparkman, 2001. Multi Agent Systems Engineering (MASE). *International Journal on Software engineering and knowledge engineering*, 11 (3): 231-258.
17. Odeh, M., 2003. A Reflective Approach to Improve Learning and Teaching of Software Engineering in Large Groups. *The International Arab Journal of Information Technology*, 1 (0): 11-17.