A two-level multi-agent architecture for a distance learning environment

Carine Webber[1] Sylvie Pesty

Laboratoire Leibniz – IMAG
46, avenue Félix Viallet
38031 Grenoble Cedex France
{Carine.Webber, Sylvie.Pesty}@imag.fr

Abstract. In this paper some results achieved in the domain of multi-agent methodology applied to educational systems are presented. We have developed a distance learning environment based on a two-level multi-agent architecture. Two-levels are distinguished since the application is composed by two multi-agent systems (MAS). The higher-level MAS is composed of cognitive agents, which provide the main functions of the system. The lower-level MAS is composed of a large number of reactive agents responsible for diagnosing students' conceptions. The two levels communicate through tutor agents, whose educational decisions are based on *emergent* results coming from the lower level. Besides the multi-agent approach, our work is founded on the emergent theory and employs a mechanism of voting for capturing group decision. This text presents methodological and theoretical aspects of our platform, the architecture proposed and the first prototype implemented for teaching geometry proof.

Keywords: distance learning environment, multi-agent systems, educational software engineering, emergent theory, voting theory.

1. Introduction

Multi-agent methodology has recently appeared as an alternative to conceive AI-based educational systems. The traditional architectures have proved to be too monolithic to deal with the new expectations of systems that should be able to provide "learning anytime and anywhere". Aspects such as data persistence and mobility become extremely important in the design of this new class of educational systems. Besides, researchers in the educational field have shown that it is not possible to find a general strategy of teaching if we take into account human differences but it is rather probable to think that learning is an emergent result of rich and coherent interactions occurred during time [2]. The multi-agent methodology can certainly bring several advantages to the development of educational applications since it deals well with applications where such crucial issues (distance, cooperation among different entities and integration of different components of software) are found. As a result, multi-agent systems (MAS), together with technologies of networking and telecommunications, bring powerful resources to develop educational systems.

Several projects implement learning systems based on multi-agents architectures. Some of them work on a generic platform of agents [6, 17, 24, 25]. For example, JTS is a web-based environment for learning Java language [28] based on a CORBA platform and using Microsoft agents. In this environment, students have access to their student models and they are able to change it, in the case they do not agree with the information represented. Another example is I-Help [25], a web-based application that allows students to locate human peers and artificial resources available in the environment to get help during learning activities. I-Help is an example of a large-scale multi-agent learning environment [26]. Moreover, interesting results have been achieved by pedagogical agents [16] regarding the student motivation and companion agents [7] acting sometimes as mediators [8] of the learning process. Finally, tutor agents [20] are usually related to student modelling and didactic decision taking [21].

[1] Scholar of Capes/Cofecub, Brazil.

In this paper we propose a two-level multi-agent system consisting of a higher-level MAS, the whole application, and a lower-level MAS, dedicated to the diagnosis of students' conceptions. Interactions among artificial agents and humans (student and teachers) determine the whole behaviour of the application. The lower-level MAS is launched by the main MAS whenever a diagnosis is requested. The emergent approach guides the development of the multi-agent architecture in both levels: at higher-level, different interactions (agent to agent, agent to student/teacher, teacher to student, student/teacher to agents, and so on) determine its behaviour; at lower-level, reactive agents interact to produce a local "picture" of the student when solving a problem, a set of these pictures characterize student's conceptions. Conceptions theory was developed in the domain of mathematical education as a way of representing learners' knowledge [1] and it is formalised later in this text.

This paper is organised as follows: next section introduces a methodological approach to design multi-agent systems. Third and fourth sections describe higher and lower levels of multi-agent systems. Fifth section briefly explains the behaviour of agents from the lower-level MAS based on emergent and voting theories. Sixth section illustrates Baghera, a system based on the architecture proposed. Finally, the last section concludes this paper and presents a few perspectives.

2. AEIO methodology

We have employed the **AEIO** methodology for a multi-agent-oriented analysis and design of systems [9]. This methodology considers the problem to be modelled as composed by four elements: **A**gents, **E**nvironment, **I**nteractions and **O**rganization. The first step of the methodology consists in defining the four elements. According to the specificities of each problem, one or more of the followings approaches may orient the system designer:

1. Agent-oriented: when it is possible to define the number of agents, their roles, their type (cognitive, reactive, hybrid) and their behaviours to guide problem solving;
2. Environment-oriented: when a well-represented world is the most important part of the system (e.g., mobile robots moving in a room);
3. Interaction-Organisation-oriented: when powerful mechanisms of coordination and cooperation among agents are needed; when interactions among agents are central to the application; and finally, when agents need to form coalitions to solve problems.

Once the four components are identified, the methodology proposes how to specify them. Of course the best tool to specify the component's behaviour depends on its function, role and type. Usually agents' behaviours are expressed by finite automates (for reactive agents) or more complex models as knowledge-based systems (for cognitive agents). Environment modelling is dependent on the application domain and very often spatial models are employed. Interaction languages can be based on models of force or allow a higher level of communication based on speech act theory. Finally, organisation model may be inspired by behaviours observed by researchers on biology or sociology.

In the case of the target application, the approach employed has privileged interactions between artificial agents and humans (students and teachers). The interaction-organisation-oriented approach was chosen and we started by defining the agents without an extensive description of their roles. Next, natural-language-based scenarios were created to study interactions among users and the application (the agents) for each use case. This process was repeated many times until a coherent set of agents, interactions and behaviours has came out. Through this process a so-called *higher-level MAS* was specified and it is presented in the next section. We distinguish the main MAS, whose behaviour provides the main functions of the application, from the secondary MAS, or *lower-level MAS*. Reactive agents compose the secondary MAS and they are responsible for diagnosing student's conceptions. This society has a behaviour based on prior theoretical work made in the domain of mathematics educational and it is presented later in this paper.

3. Higher-level MAS

Students and teachers interact with different kinds of agents. Persistent data are kept in students' schoolbags and teachers' electronic folders and they are in the model represented by objects belonging to the environment. They are personal repositories of data and teachers have access rights to schoolbags of students belonging to their classes. Each student counts on three artificial agents:

- *Companion - Student's Personal Interface Agent*

  It is an agent associated with the student's interface with a wide range of goals. Mainly it monitors the student's actions, notifying other agents when needed and giving access to system resources. This agent controls the access to the student's schoolbag and brings to the user information about the whole learning environment.
- *Tutor Agents*

  Tutor agents propose the most suitable problem/situation to the student, regarding educational goals and the context of learning. Furthermore, their didactical decisions are based on students' conceptions. To accomplish their goals they are able to launch the lower-level MAS whenever a diagnosis is needed (e.g. a student has finished an exercise, so the tutor has to decide what to propose as next activity) and, once diagnosis phase is over, they plan interactions with other agents and users.
- *Mediator Agent*

  The aim of this agent is to choose an appropriate problem solver to send the student's solutions. As it is shown later for the case of geometry proof learning, this agent is connected to an automatic theorem prover, being able to perform proof verification, propose alternative proofs and build counter-examples. Besides these functions, this agent implements techniques to analyse and present proofs.

Similarly, each teacher counts on the two following artificial agents:

- *Teacher's Personal Interface Agent*

  It is an agent associated with the teacher's interface. This agent controls the access to the teacher's electronic folder and brings to the user information about the whole learning environment. This agent mediates interface functions related to: communication with other human and artificial agents, edition of new activities to the students, distribution of such activities to students, and supervision of work done by students.
- *Assistant Agent*

  An assistant agent is also a kind of personal agent whose goals include assisting the teacher with the creation and distribution of new activities, which are kept in the teacher's electronic folder. This agent controls the access to the teacher's electronic folder and, when demanded, it hands the activities out to students.

As an open multi-agent system, the number of agents in the society increases or decreases depending on the number of users logged in. For instance, in a specific moment, given a number n of students and m of teachers logged in, the number of active artificial agents is 3n+2m. We consider this an important remark since the number of connections is not limited and the number of agents is not fixed in the society.

The general architecture of the agents is shown in the figure 1 and it was inspired by the BDI model [19]. The knowledge base represents internal states of the agent itself and the others. Agents communicate by interaction protocols. In addition, an agent may have plans, goals and actions to run. Actions are expressed by one or more messages destined to other artificial agents or users. The reasoning and control module is responsible for assuring the execution of protocols and determining the sequence of actions.

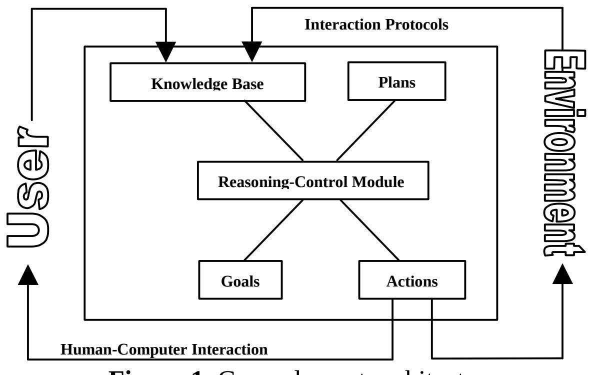

Figure 1. General agent architecture.

The higher-level MAS is operational and its first implementation is presented in section 6. However, tutor agents have in this version an incomplete behaviour since it is dependent of the diagnosis from the lower-level society under development.

4. Lower-level MAS

The lower-level MAS has the task of diagnosing student's conceptions. This is one of the bottlenecks of research on learning environments. Our approach was conceived taking diagnosis as the emergent result of collective actions of reactive agents under development. Before describing the architecture of this level, the theoretical framework guiding our approach for modelling student's conceptions is explained.

4.1 Defining conceptions

Before introducing the formal definition of conceptions, we give two examples of conceptions that learners can have in the domain of reflection in geometry [12]. The first example of conception is called parallelism and students holding it *believe* that if two line segments are symmetrical then they are parallels and have the same size (see construction on figure 2). The second example corresponds to the correct conception about reflection and it is shown on figure 3. Even holding incorrect conceptions, learners are effectively able to solve some problems correctly. Observe the example on figure 4 where students holding both conceptions may construct the correct symmetrical line segment of AB.

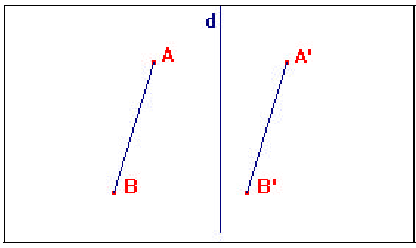

Figure 2. Misconception of parallelism.

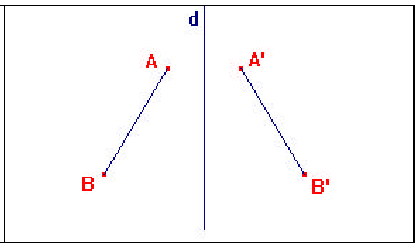

Figure 3. Reflection.

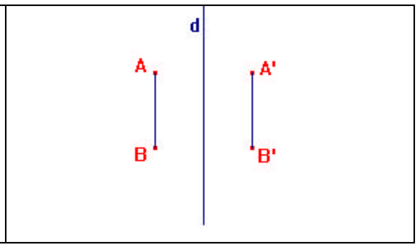

Figure 4. Parallelism or Reflection?

We assume that learners have conceptions, which may be correct or incorrect, but in any case these conceptions have a domain of validity; they exist because they solve some problems correctly [1]. Recognising which conception (correct or incorrect) students hold may help to understand the reasoning guiding their problem solving decisions. Furthermore, it may guide the choice of new problems to be presented next to the student. However, describing precisely conceptions is a difficult problem, and for this purpose we use a model developed in mathematics education with a cognitive and didactical foundation [1]. In this model a conception is characterised by a quadruplet *C (P, R, L, S)* where:

- *P* represents a set of problems, which describe the conception's domain of validity, it is the seminal context where the conception may appear;
- *R* represents a set of operators, which are involved in the solutions of problems from *P*;
- *L* is a representation system; it allows the representation of problems and operators;
- *S* is a control structure, which guarantees that the solution holds the conception's definition; it allows making choices and decisions during problem solving.

Indeed, one element could contribute to the characterisation of several different conceptions; for example two conceptions may share problems in their domain of validity or may have in common operators, or controls, or even may rely on the same representation system. This opens the possibility to compare and relate one conception to the others.

4.2 Lower-level architecture

Conceptions are characterised by sets of agents. The society of agents is composed of four categories: problems, operators, language and control. Each element from the quadruplet *C (P, R, L, S)* is the core of one reactive agent. So, each agent taking part in the society belongs to one category and has a unique behaviour inside the society. An agent, in a given time slot, can be either active, because it is satisfied, or inactive and unsatisfied. This state can vary according to changes it perceives in its environment, in the objects and in other agents' states.

The general role of any agent is to check whether the element it represents is present in the environment. In the presence of the element, the agent becomes satisfied. Once satisfied, the agent is able to influence the satisfaction of other agents by voting. Voting mechanism is introduced later in this paper. In the case of the absence of the element represented, the agent looses the right to vote but still may have some influence over the satisfaction of other agents. A description of the role of each category of agents is given below.

Problem Agents: a problem agent becomes satisfied when the category of problems it represents is present in the environment. In the domain of reflection, a category of problems is described by four didactical variables named: line of symmetry orientation, line segment orientation, angle formed between line of symmetry and line segment and intersection formed between the line of symmetry and line segment. For instance, the configuration shown on figure 2 is described as having vertical line of symmetry, oblique line segment, an unspecified angle between them and no intersection. The combination of the different values these didactical variables could take, leads to more or less complex problems, allowing to focus on different aspects of the learning of reflection and most important, allowing the expression of different conceptions.

Operator Agents: An operator agent becomes satisfied when the element r of *R* it represents, is present in the solution constructed by the student. An operator transforms a problem in a new problem. A sequence of operators leads to the problem solution. An example of an operator in the domain of reflection is the following: *If two segments are symmetrical with relation to an axis, then their midpoints are symmetrical to each other.*

Language Agents: A language agent becomes satisfied when the element l of *L* it represents, is present in the solution constructed by the student. It can be a grammar, a graphical representation, or an alternative way of expression allowing the description of the problem and the solution.

Control Agents: A control agent becomes satisfied when the element s of *S* it represents, is present in the solution constructed by the student. In problem solving, learners choose operators, validate actions and validate the final result. Each one of these three decisions is guided by control structures. Control elements are perceptive when attached to the fact that the learner makes assertions based on something "seen" on the screen and uses this information to take and validate decisions. On the other hand, control structures are theoretical when a learner bases decisions and validations on knowledge previously acquired. Reflection involves many visual elements of control; for instance, a learner holding the conception of parallelism may accept that a problem is correctly solved when the image line segment "looks" parallel to the original line segment.

Recall that the lower-level MAS is activated by the higher-level MAS whenever a diagnosis is requested. The lower-level environment is shared by reactive agents, objects representing the solution constructed by the learner, objects describing the problem solved and simple objects representing candidate conceptions. Reactive agents are able to perceive the state and voting choices of other agents.

In the next section, we present the foundations of our approach for defining the lower-level MAS and the process for diagnosing learners' conceptions.

5. Foundations of the lower-level behaviour

5.1 Emergent Theory

The literature about emergence is very extensive and rich. In essence, the concept of emergence comes from British philosophers of the XIXth century and sets that properties of a complex physical system are emergent whenever they are neither (i) properties of its parts taken in isolation nor (ii) the result of a mere sum of the properties of its parts. Furthermore, the concept of emergence is opposed to the concept of reductionism, where the global properties of a system can always be explained and predicted from the properties and the structural relations of its units. For the last three decades, this approach has influenced many sciences other than philosophy (mind-body problem), such as social science, economics, biology, complex theory, and cognitive sciences.

In computer science, research on emergent phenomena has been traditionally focused on the simulation of physical or biological process, simulation of colonies organisation, games, coalition formation, emergence of political structures and emergent behaviour [13,22,23]. According to some authors, there is emergence whenever a lower-level phenomenon can be recognised by higher-level detectors and the emergent product has properties impossible to deduce from the properties of lower-level elements [3,10]. Rather than the simulation of emergent behaviour, this work is concerned with the emergence of information (student’s conceptions) for further processing.

5.2 Voting Theory

Voting measures are widely used in social sciences and have their roots in Game Theory. Social sciences research about voting has been investigating new approaches to studying voting schemes, voter behaviour, and the influences of manipulation of votes and insincere voting. Studies based on simulation of elections have led to models providing explanations to voter behaviour, so as explanations to group decisions and coalition formation.

In the domain of multi-agent systems, voting theory has been used as a technique for reaching consensus in a negotiation process and group-decision making [27]. Applications for negotiating meetings have used voting to reach a consensus in an acceptable time for the meeting. In the simulation of coalitions formation, agents have been used to demonstrate how decisions emerge from individual voter preferences. Furthermore, it has been shown that emergent structures can be resultant of a voting process. Schreber [23] has demonstrated through multi-agent simulations that elites and parties are emergent consequences of the behaviour and preferences of voters. Similarly, political structures (as collections of instruction, rules or conventions that regulate political actions or activities) may emerge from the voting behaviour of agents.

In essence our interest in voting theory relies on the possibility of capturing group decision as well as modelling the influence of an agent preference over the preferences of the rest of agents. So far, no former attempts were made to obtain an emergent diagnosis from voting behaviour. Nevertheless, Maxion [18] has demonstrated how diagnoses may be the emergent result of a process when in a microscopic level symptoms are recognised and characterise a macroscopic finding.

5.3 Agent’s general behaviour

Even if the elements each agent perceives in the environment change from agent to agent, all agents have a similar general behaviour. Four main steps describe a general behaviour:

- Initialisation

The same environment is shared by operator agents, problem agents, language agents, control agents, objects representing the solution constructed by the student, objects describing the problem solved and simple objects representing candidate conceptions. Agents are initialised in an inactive state, meaning that they cannot vote.

- First iteraction

Agents perceive the environment. If an agent perceives in the environment the presence of elements represented, it becomes active. Since the agent is active, it is able to vote for one or more candidate conceptions. On the other hand, if elements represented are not present in the environment the agent do not change its inactive state.

- Next iteractions

Agents keep updating their states according to changes perceived in the environment. Voting occurs in turns and active agents can vote to one or more candidate conceptions.

- Stabilisation

When no significant changes are perceived in agent's states and voting preferences, diagnosis is considered to be over. The state of an agent is conditioned by the presence of the elements it is able to perceive in the environment. An inactive agent can perceive its environment, and this inactivity may influence other agents. An inactive agent has no right to vote. When an inactive agent becomes satisfied, it changes its state and becomes active. An active agent is able to vote and influence other agents because of its activity and its voting behaviour. An active agent may become inactive again if at least one of the elements it represents is not present in the environment anymore.

5.4 Voting Turns

Voting has been used as a way for both group-decision making and explicit representation of the emergent result. Agents have simple strategies to vote: each agent can vote for one or more candidate conceptions and the vote is given based on the agent's belief to the fact that the candidate conception is a good representation of the conception employed by the learner in the solution currently considered. The category of the problem solved, the state of other agents in the society and the previous votes given by other agents in the society influence as well agent's voting. Moreover, when an agent votes it may influence the votes of other agents.

Majority voting method is used by the lower-level MAS. Majority voting method considers that votes have equal weight and the candidate with the highest number of votes is the preferred one. Voting is organised in multiple turns. Each active agent votes for one or more candidate conceptions in each turn. For the first turn, agents have no information about other agents' states or voting. It is only from the second turn on that such information will be known by the society. When a new turn takes place, as the environment has changed, agents review their voting and vote again. When new votes corresponding to the last turn are published, they might again influence agents' voting and agents' states. During voting, an agent can suddenly become active and able to vote, or inactive, and loose the right to vote.

Multiple turns are necessary so the agents can apply particular voting strategies and can be informed of the others votes, until the stability of the voting results. Information about the preference of others allows an agent to change its vote in the next turn of voting, according to its voting strategy.

Some agents are pivotal in determining the voting outcome. This is possible because of its influence on voter's behaviours. Even if their vote has the same weight as the votes given from the rest of the agents, its voting power is greater because the element they perceive in the environment is as well pivotal to characterise some conception.

Agents have preferences among candidate conceptions. Additionally, they can be influenced by other agents' votes. This is due to the fact that an agent may consider the voting behaviour of another agent, as a reason to review its own voting choices. As a result of this review, a vote may be strengthened, meaning that the agent becomes more confident about its decision, or may be weakened, because the agent finds not enough indices to believe that such a conception is a good candidate. When voting becomes stable, it is considered that agents have reached a consensus about the strongest candidate conception.

5.5 An Example of Diagnosis

The next example considers just a few agents as it is related to one step of one possible proof to solve a specific problem. In order to illustrate a complete diagnosis much more agents should be considered. However, through such example the basic mechanisms to diagnose conceptions are described in a way to facilitate its comprehension.

Let's consider the figure 5 that presents the statement and the graphical representation of a problem in geometry.

The line segment [AB] is parallel to the line *d*. [A'B'] is symmetrical to the line segment [AB] with respect to line d. What kind of quadrilateral is ABB'A' ?

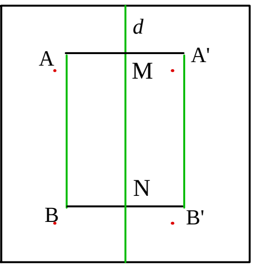

Figure 5. A problem in the domain of reflection.

Assuming that a rectangle is a parallelogram having one right angle, a strategy to solve this problem could be composed by the following steps:

1. Prove that AB is parallel to A'B';
2. Prove that AA' is parallel to BB';
3. Prove that AA' is perpendicular to AB;
4. Conclude that ABB'A' is a rectangle.

A student may propose another proof, although wrong, which could be to assume that a rectangle is a quadrilateral having opposite sides parallel and with the same length:

1. Prove that AB is parallel to A'B';
2. Prove that AA' is parallel to BB';
3. Prove that AB and A'B' have the same length;
4. Prove that AA' and BB' have the same length;
5. Conclude that ABB'A' is a rectangle.

Let's observe the step number 1, common to both schemes. To prove that AB is parallel to A'B', three examples are presented in table 1.

Let an agent A_1 be an agent responsible for checking whether the relation of parallelism is present in the student's proof and judge whether it is correct. This agent observes the presence of elements in the proof, the problem solved and the possible active status of other agents.

Let's consider that the student constructed a proof like the one shown in table 1 - example 1. Lines 1-4 activate an operator agent A_2 responsible for checking the use of symmetrical properties (symmetry conserves parallelism). Lines 5-9 activate an agent A_3 responsible for verifying the use of another property (transitivity of parallelism). In addition, a problem agent A_4 representing the category of problems including those having the original line segment and the axis of symmetry in a vertical orientation becomes active.

Table 1. Proving that AB is parallel to A'B'.

1	It is known that AB is parallel to line d	1
2	and AB is symmetrical to A'B' with respect to line d	
3	and since reflection respects relations of parallelism	
4	we conclude that A'B' is parallel to line d	
5	As AB is parallel to line d	
6	and A'B' is parallel to line d as well	
7	and by the transitivity of parallelism	
8	we conclude that **line segment AB is parallel to the line**	
9	**segment A'B'** .	
1	**Line segment AB is parallel to the line segment A'B'**	2
2	Because	
3	A'B' is the symmetrical line segment of AB	
4	and symmetrical line segments are parallel	
1	**Line segment AB is parallel to the line segment A'B'**	3
2	Because	
3	AB is a vertical line segment and	
4	A'B' is a vertical line segment	
5	and it is know that two vertical lines are parallel.	

Next to the activation of agents A_2, A_3 and A_4, the agent A_1 becomes active as well since this agent witness the presence of parallelism. For this particular case, agent A_1 votes for the conception of reflection, because correct operators were detected and the agent believes that the student has the correct conception about reflection. Agents' voting is presented in table 2.

Table 2. Agents voting for example 1.

Agents	Parallelism	Reflection
A_1		vote
A_2		vote
A_3	vote	vote
A_4	vote	vote
Total	2 votes	4 votes

Observe that A3 and A4 vote for both conceptions because the elements they represent may be present in both conceptions and they cannot decide for the one or the other. As voting is organised in turns, for next turns, their voting may change if they are influenced by other agents' voting.

Let's now consider the example 2 from table 1. Lines 1-4 activate an agent A_5 that recognises the use of an incorrect assumption (symmetrical line segments are parallel) which applies the notion of symmetry to demonstrate the parallelism between two line segments. Suppose that the agent A_4, used in the last example, becomes active since we take the same previous problem. Next to the activation of agents A_4 and A_5 agent A_1 becomes active. However, this time the agent A_1 is active as a result of the activation of a different set of agents and it votes for the conception of parallelism. Voting is shown in table 3.

Table 3. Agents voting for example 2.

Agents	Parallelism	Reflection
A_1	vote	
A_4	vote	vote
A_5	vote	
Total	3 votes	1 vote

Since a conception of parallelism is an incorrect conception about reflection, agent A1 becomes pivotal. It means that even if its vote has no more power that others votes, it may influence agents' voting because of the importance of the information detected.

When agents from the lower-level MAS become stable, since no significant changes are perceived in agent's states and voting preferences, diagnosis is considered to be over. At this moment, the tutor agent (from the higher- level MAS) uses the results from voting as an input to decide how to respond to the student. For instance, once a correct conception is diagnosed, tutor agent may want validate the diagnoses by proposing a problem where correct procedures of resolution are tested and possibly reinforced. On the other hand, if an incorrect conception is diagnosed, the tutor agent may propose problems where the procedure of resolution, previously employed by the student, fails or it may propose problems where the correct procedures of resolution appear, making the student aware of them. Besides the interactions with the lower-level society of agents, other sources like problem solvers -ATINF, discussed later - and other tutor agents may be useful providing more inputs for tutors' pedagogical decision making.

6. Baghera learning environment

We have implemented an environment, based on the higher-level multi-agent architecture (section 3), to provide individualised support for problem solving. As already mentioned, the lower-level MAS is under development and it is not yet integrated to the prototype shown in this section. The current version of the platform was developed using JatLite [15], a package of programs for creating software agents. Each agent was extended by an interaction module. It provides support for creating protocols and coordinating interactions and the execution of protocols [14]. Agents have the ability to communicate with other agents and take decisions. Communication among agents is based on the speech act theory in accordance with FIPA-ACL standards [11]. Baghera is an operational system, which has been experienced by researches in mathematics education and most recently by partners of the Baghera European project.

Students and teachers have access to Baghera through our website (http://www-baghera.imag.fr). A user identification and password are required by the applet, which gives access to the application. Baghera was implemented using Java language and Swing libraries. Figures 6, 7 and 8 show student's interfaces.

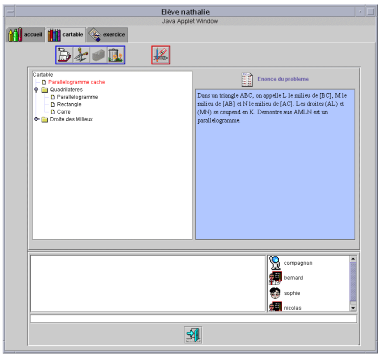

Figure 6. Student's electronic schoolbag.

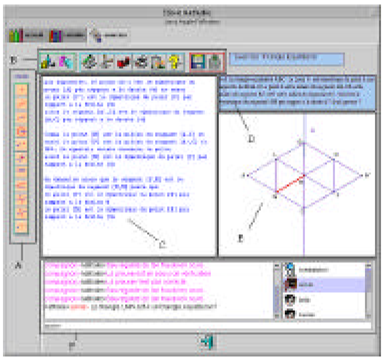

Figure 7. Problem solving interface.

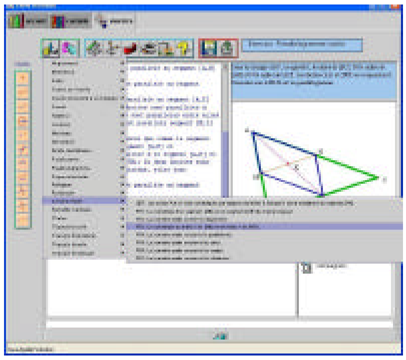

Figure 8. Library of theorems in geometry.

Figure 6 illustrate the student's schoolbag, where problems sent by teachers are kept and organised. Once a problem is chosen, the student goes to problem solving interface, shown on figure 7. On this interface students find tools to edit a proof (A and B) as a free text (C). The problem proposed is composed by a statement (D) and a dynamic figure (E) constructed using Cabri-Java [4]. The dynamic figure can be manipulated in some ways that the student may verify the geometrical properties of it and new problem hypothesis may come out. Following, figure 8 illustrates the library of theorems and properties from geometry that students can consult and insert into their text. The proof can be verified whenever wanted by the student. And at this moment, proof is translated and sent to ATINF, an automatic theorem prover instantiated for the domain of geometry [5]. The companion agent informs the student about the state of his/her proof.

Teachers' interfaces have extra functionalities, as partially shown on figures 9, 10 and 11.

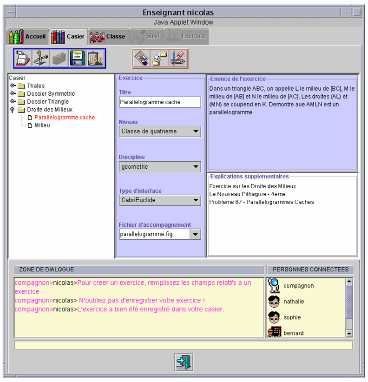

Figure 9. Creating new problems.

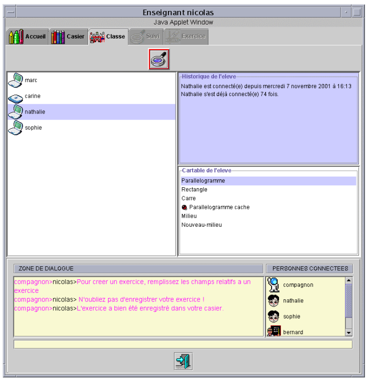

Figure 10. Virtual class.

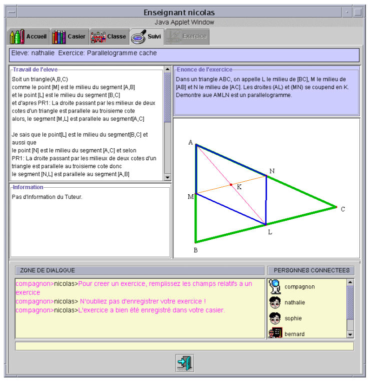

Figure 11. Supervision of student's work.

The first capture (figure 9) shows the interface for creation and diffusion of problems. The second one (figure 10) presents the interface that allows the teacher to control student's connections and access to schoolbags. Finally, the last capture (figure 11) allows the teacher to supervise student's work by the observation of proofs constructed regarding particular problems.

7. Conclusions

In this paper we have presented a hybrid multi-agent architecture of a learning environment. Cognitive agents form the higher-level and their interactions determine the behaviour of the whole system. On the other hand, reactive agents are launched to collectively diagnose student's conceptions. Diagnosis is taken as the emergent result of the interactions of reactive agents.

The mechanism of diagnosis presented is based on multi-agents modelling and emergent theory and it is under implementation. We have followed this approach since we recognise that we may not search for an a priori student model, but look for the model of a process which may allow the construction of the student model depending on the specific circumstances which contextualise the student.

Based on this multi-agent architecture we have implemented a distance learning environment called Baghera. Baghera is intended to constitute an educational community of artificial and human agents, which cooperation helps students to learn. In addition, it is important to remark that our main goal is not only toward the development of a new distributed architecture for educational systems. We intend to go further and search for models based on emergent theory, which could overcome the absence of a general pedagogical model of teaching by allowing the dynamic construction of strategies based on local educational solutions.

References

[1] Balacheff, N. *A modelling challenge: untangling learners' knowing*. In: Journées Internationales d'Orsay sur les Sciences Cognitives: L'apprentissage, JIOSC2000, Paris, 2000.

[2] Balacheff, N. *Teaching, an emergent property of eLearning environments*. In: Conférence IST 2000. Nice, France, 2000. Available at http://www-didactique.imag.fr/Balacheff/TextesDivers/IST2000.html.

[3] Bonabeau, E.; Dessalles, J.; Grumbach, A. *Characterizing Emergent Phenomena (1): A critical review*. In: Revue Internationale de Systémique, vol.9, nº3, 1995, pp. 327-346.

[4] Cabri Java. Available at http://www.cabri.net/cabrijava/.

[5] Caferra, R.; Peltier, N.; Puitg, F. *Emphasizing human techniques in geometry automated theorem proving: a practical realization*. In: Workshop on Automated Deduction in Geometry, Richter-Gebert J., Wang D.(eds). Zurich, Switzerland, pages 38-59. 2000.

[6] Capuano, N.; Marsella, M.; Salerno, S. *ABITS: An Agent Based Intelligent Tutoring System for Distance Learning*. In : Proceedings of the International Workshop in Adaptative and Intelligent Web-based Educational Systems. Available at http://virtcampus.cl-ki.uni-osnabrueck.de/its-2000/.

[7] Chan, T.W. *Learning Companion Systems, Social Learning Systems, and Intelligent Virtual Classroom*. Invited Talk, World Conference on Artificial Intelligence in Education, Washington, DC, USA, Journal of Artificial Intelligence in Education, Vol. 7, No. 2, pp. 125-159. 1996.

[8] Conati, C. Klawe, M. *Socially Intelligent Agents to Improve the Effectiveness of Educational Games*. In: Proceedings of AAAI Fall Symposium on Socially Intelligent Agents - The human in the loop. 2000. Available at http://www.cs.ubc.ca/~conati/publications.html.

[9] Demazeau, Y. *From Interactions to Collective Behaviour in Agent-Based Systems.* In: Proceedings of the 1st. European Conference on Cognitive Science. Saint-Malo, France, 1995.

[10] Forrest, S.; Miller, J. *Emergent Behavior in Classifier Systems*. In: Physica D 42, 1990, pp. 213-227. Elsevier Science Publishers, North-Holland.

[11] Foundation for Intelligent Physical Agents. Available at http://www.fipa.org/.

[12] Grenier, D. *Construction et étude du fonctionnement d 'un processus d ' enseignement sur la symétrie orthogonale en sixième*. Thèse. IMAG, Université Joseph Fourier. 1988.

[13] Holland, J.H. *Emergence from chaos to order*. UK: Oxford University Press, 2000. pp. 258.

[14] Huget, M-P; Koning, J-L.; Bergia, L. *Une Plate-forme pour l'ingénierie des protocoles d'interaction. Application au projet de télé-enseignement Baghera*. In: Systèmes multi-agents: méthodologie, technologie et expérience, 8emes Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents, JFIADSMA'00, S. Pesty & Claudette Sayettat-Fau (eds.),2000, pp 297-301.

[15] JATLite - Java Agent Template, Lite. Available at http://java.stanford.edu/.

[16] Johnson, W. L.; Rickel, J.W.; Lester, J.C. *Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments*. In: The International Journal of Artificial Intelligence in Education, 11, pp.47-78. 2000.

[17] Machado, I.; Martins, A.; Paiva, A. *One for All and All for One A learner modelling server in a multi-agent platform*. In: Proceedings of the Seventh International Conference on User Modelling, pp. 211-221, 1999.

[18] Maxion, R.A. *Toward diagnosis as an emergent behavior in a network ecosystem*. In: Physica D 42, 1990, pp. 66-84. Elsevier Science Publishers, North-Holland.

[19] Rao, A.; Georgeff, M. *BDI Agents: From Theory to Practice*. Proceedings of the First International Conference in Multi-Agents Systems, San Francisco, USA. 1995.

[20] Ritter, S. *Communication, Cooperation and Competition among Multiple Tutor Agents*. In: Proceedings of the 8th World Conference of Artificial Intelligence in Education. Du Boulay,B.; Mizoguchi, R. (Eds). IOS Press, 1997.

[21] Ritter, S.; Koedinger, K.R. *An architecture for plug-in tutor agents*. Journal of Artificial Intelligence in Education, Vol.7, pp. 315-347. 1996.

[22] Sawyer, R.K. *Simulating Emergence and Downward Causation in Small Groups*. In: S. Moss and P.Davidsson (Eds.) Multi-Agent-based Simulation. Second International Workshop, MABS 2000, LNAI 1979, pp.49-67, 2000. Springer-Verlag Berlin Heidelberg 2000.

[23] Schreiber, D. *The emergence of parties: an agent-based simulation*. In: The Annual Meeting of the Midwestern Political Science Association, Chicago, Illinois, 2000.

[24] Silveira, R.A.; Bica, F.; Viccari, R.M. *JADE - Java Agents for Distance Education Framework*. In: Proceedings of the Third Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems. Alvares,L.O.,Bazan, A.L.(Eds.), Brazil, 2000. pp.112-122.

[25] Vassileva, J.; Greer, J.; McCalla, G.; Deters, R. *A Multi-Agent Approach to the Design of Peer-Help Environments*. In: Proceedings of AIED'99, Le Mans, France, pp. 38-45. 1999.

[26] Vassileva, J.; Deters, R.; Greer, J. McCalla, G.; Bull, S.; Kettel, L. *Lessons from Deploying I-Help*. In: Proceedings of the Workshop Multi-Agent Architectures for Distributed Learning Environments, AI-ED'2001, San Antonio, TX, USA. pp. 3-11.

[27] Weiss, G. (Ed.) (1999). *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. The MIT-Press.

[28] Zapata-Rivera, J.D.; Greer, J. *SMODEL Server: Student Modelling in Distributed Multi-Agent Tutoring Systems*. In: Artificial Intelligence in Education. J.D.Moore, C.L.Redfield and W.L.Johnson (Eds.). IOS Press, 2001. pp.446-455.