Towards Adaptability in Intelligent Tutoring Systems: A MultiAgent Approach

Maria das Graças Bruno Marietto[1], Nizam Omar[2], Edson Pinheiro Pimentel[3], and Vilma Fernandes de França[3]

[1] Universidade Católica Dom Bosco, Campo Grande, MS
{marietto} @ec.ucdb.br
http://www.ec.ucdb.br
[2] Universidade Presbiteriana Mackenzie, São Paulo SP
{Omar} @mackenzie.br
http://www.mackenzie.br
[3] Instituto Tecnológico da Aeronáutica, São José dos Campos, SP
{edsonpp} @uol.com.br
{vilmaff} @comp.ita.br
http://www.comp.ita.br

Abstract. This paper presents the description of an Intelligent Tutoring System under its agents' aspect, whose main characteristic is high adaptability degree. ITStrategic is a prototype of a Multiagent System that uses cognitive agents in its structure. The agents of ITStrategic use the KQML communication language, and the interactions among them are based on five communication / negotiation protocols. Its main objective is to complement the teaching in the classroom, through an intelligent computational system. It is a friendly system that allows to the student a simple navigation through the contents, to choose exercises to be resolved, and verify how he is improving his knowledgment.

1 Introduction

An adaptative Intelligent Tutoring System (ITS) changes its behavior depending on the students' mental states. This is an important feature in such systems which works in dynamic environments, and need to guarantee a personalized instruction. To deal with such complex question, the ITS area borrows theories, methodologies and techniques from several areas like Psychology, Distributed Artificial Intelligence and Education.[10].

Considering the Distributed Artifical Intelligence (DAI) area, it can brings to ITS a social philosophy where problems are modeled in a cooperative, collaborative and distributed way, using processes called agents. Such philosophy can be applied in ITS's architectures, since this social approach is common to both areas.

STI ITStrategic prototype was developed from a Multiagents architecture of cognitive agents. Such agents are called Receiver, Guide, Information, Tutor and Delivery.

This paper is organized as follows. Section 2 presents the used vision of Distributed Artificial Intelligence and Intelligent Tutoring System in Education. In Section 3 the ITStrategic system structure is presented, emphasizing its agents and describing the communication language and protocols . Finally in Section 4 some conclusions are given.

2 Artificial Intelligence in Education

DAI techniques associated to teaching methods can contribute to reach better learning levels. In this context, Intelligent Tutoring Systems have been shown to be highly effective at increasing students' performance and motivation [11].

Multiagent Systems (MAS) is a DAI's subarea which is concerned with the study, behavior, ande construction of a collection of possibly preexisting autonomous agents that interact with each other and their environments [12]. MAS focus is in the agents' structuring, and not in problem structuring. The designer worries about the development of an agents' architecture to work in an autonomous and social way, this approach consists of coordinating the intelligent behavior in a group of autonomous agents, whose existence can be previous to the appearance of a specific problem. MAS are classified in reactive and cognitive. Cognitive MAS are able to reason according to their intentions and knowledge. They have deliberative control because they deliberate which action will be executed, and later they execute it.

The concept of cognitive agents interacting to solve a problem is a strong metaphor of human being behavior, which can be used to structure Intelligent Tutoring Systems. This cross-fertilization takes more importance when considering the possibility to model different pedagogical strategies to be adopted depending on the student's mental states. This is a challenge on the area, since the most current ITS works with only one pedagogical strategy throught all the instruction process.

ITStrategic's chief motivation is the dynamic definition of several instructionals strategies in computerised tutorial processes through its Multiagents structure.

3 ITStrategic and Adaptability in ITS

The ITStrategic is composed by five modules: *Perception*, *Student*, *Domain*, *Tutorial* and *Delivery*. Additionaly, the system has three Graphical User Interface (GUI): *Administrator*, *Student* and *Educator*. Figure 1 illustrates the system architecture.

In the following will be presented the ITStrategic's agents principles, agents architecture, communication language and protocols and the overview of each specific agent.

ITStrategic's agents skills are based on five design principles, which are:

- Benevolence principle: Agents will help each other, as long as their objectives are not conflicting;
- Honesty principle: Agents don't pass on deliberately way, incorrect information to other agents or to the environment;
- Self-knowledge principle: Each agent has a correct and complete representation of themselves;
- Asynchronous Communication principle: The communication among the agents isn't necessarily synchronous;
- Open Society principle: Insertion and exclusion of Agents in the society take place in a very dynamic way.

The architecture is organized in layers, as described below :

- Interaction layer : This is responsible for the interaction between the agent and the environment. The three possible interactions are perception of the environment, action on the environment and communication with other agents;
- Behavior layer : This determines whether the agent will have a reactive or cognitive behavior;
- Coordination layer : This is linked to social behavior. It is responsible for the agent's collective action taking into account such topics as coordination, cooperation, contribution and negotiation.

The relationship among layers complies to a social organization based on activities.

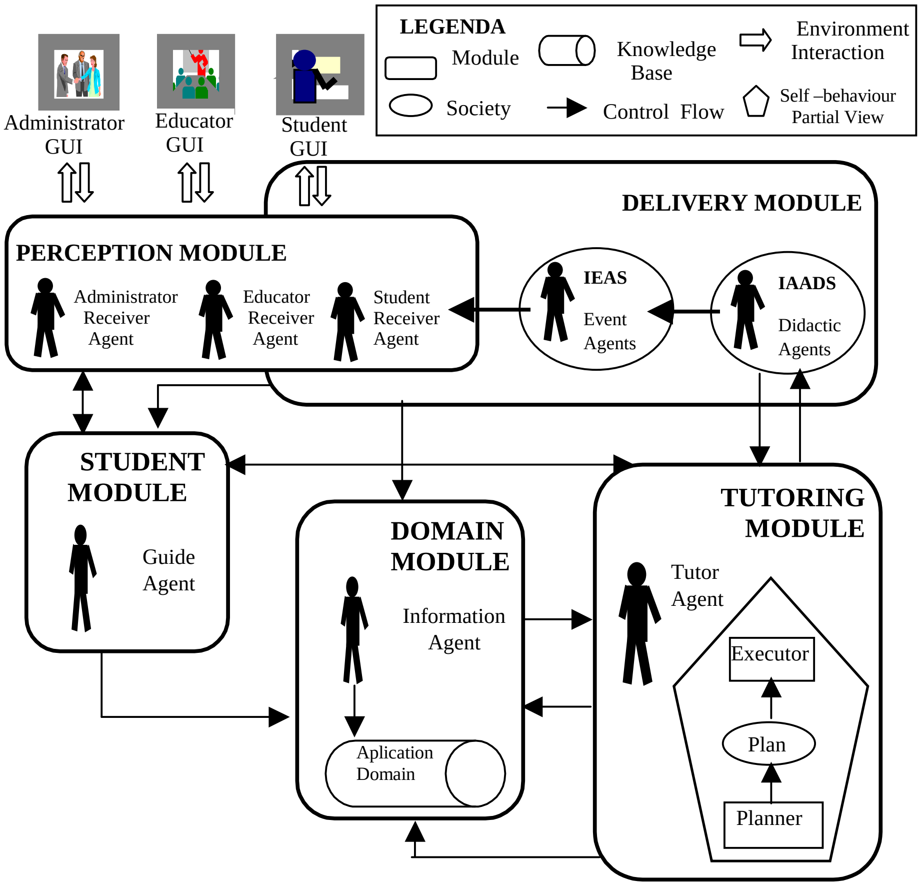

Figure 1: ITStrategic General Functional Architeture

Agents Communication

The communication language used by the agents of ITStrategic is KQML (Knowledge Query and Manipulation Language) [5], [6].

The messages in KQML are called performatives. A message KQML consists of the name of the performative (action to be executed), their arguments and a group of optional arguments [6]. In the Figure 2 an example of a KQML performative is shown.

```
(Ask-About
:Sender  DidacticDirective Agent
:Receiver Guide Agent
:Reply-With Query<#>
:Language Java
:Ontology Tutorial Structur /Functionality/Information
                on Mental State
:Content :Info Query to inform Conceptual Status
            :Par Which Concepts ? <Concept Directory> )
```

Figure 2 : *Performative Ask-About*

The communication process among agents involves protocols that seek the identification, simplification and normalization of tasks inside of a context. In that sense a MAS has communication protocols, where agents request messages or inform answers under certain conditions aiming at the global solution of the system.

Communication Protocols

In general terms, communication protocol is a group of rules to give support to the communication in a net. Protocols can accomplish, besides the communication task, the aid in the negotiation process. In this healthy sense denominated negotiation protocols that are political that the agents should proceed to guarantee the interactions with the other agents.

D' Ambrósio, Darr and Birmingham [4] and Costa [3] present the following communication/negotiation protocols: consultation protocol, contract of a task/action protocol, protocol registration, protocol announcement, protocol of acting of a task/action. In the following an abbreviation description it will be given of each one of these protocols that were used in the system ITSTrategic.

The established protocols in this section specify all the rules for the which the agents of ITStrategic interact amongst themselves. These protocols define sequences of messages, including those that begin and they conclude a protocol, and possible answers for each received message.

- Protocol Registration: it allows the an agent to inform the other agents of the society its existence;

- Protocol Announce: it offers the ability for an agent to inform the other agents of the society services/actions/tasks that it is willing to offer;
- Protocol Consultation: it defines a sequence of two messages, allowing an agent to do a consultation to the other;
- Protocol Contract of a Task/Action: it defines a sequence of messages, allowing to the agent to hire a task of another agent:
- Protocol Acting of a Task/Action: it defines a sequence of messages among two agents, allowing an agent to communicate the results of a task the other.

To obtain a deeper understanding of ITStrategic's proposals and functionalities, it is necessary to take a look on their agents as well as on the interactions among them.

Receiver Agents are responsible for the initial communication between the user and the system and are formed by the Administrator Agent, Educator Agent and Student Agent. In this version of ITStrategic Student Module is just formed by Receiver Student Agent.

The dialogue maintained among system and the user can take to the following actions:

- To identify student: ReceiverStudent Agent has an pro-active action trying to identify the student before his explicit identification.
- To present template: ReceiverStudent Agent communicates with one of Event agents seeking to present to the student the microstrategy that is being considered.
- To inform historical data: It allows to the MAS the access to the student's historical data including his pattern of behavior in the system.

Guide Agent aids in the definition of the student's mental state and could execute the following actions, as it be activated by the system administrator:

- To initialize agent: It consists of the registration and the agent's connection Guide in Router of JATLite [7].
- To introduce student in the system: Guide Agent receives from the recently identified student's agent ReceiverStudent information.
- To supply information on the Student's Model: In case the student is already registered in the system, Guide agent begins to work with the Student's Model. In case the student has been newly registered, you/he/she creates a new Model of the Student that should contain besides the personal data, the data of each done session.
- To update the Student's Model: After the Student's introduction in the system, Guide agent opens an auction process close to MAS for the accomplishment of the task "Studend Tutoring". DIAAS will request alterations in the Student's Model, regarding its acting in a certain session.

The process of determination of each student's mental state is known by diagnosis and, to do it, ITStrategic uses techniques called actions diagnosis.

Through the access to the Student's Model, Guide agent aids ITStrategic to make decisions as: to detect the learning of a concept, to detect bugs in the student's mental state, to detect the correction of an missconception.

As observed by McCalla [11], one of the main features of ITS is that they "understand" the application domain. **Information Agent** is responsible for the access to the database of the application domain in an intelligent way, and could accomplish the following actions:

- To initialize agent: It consists of the registration and agent's Information connection in Router of JATLite.
- To inform instructional architecture: Tutor Agent establishes a consultation protocol with Information Agent requesting the determination of the appropriate instructional architecture for the current student. This action is requested by the agency through performatives ASK-ABOUT.
- To inform library about practices and tests of the domain of the application.
- To inform Objectives of the domain of the application.
- To inform definitions, examples, results, hints, megapinciples and counterprinciples.

The information contained in the domain of the application may be briefly separated into two groups :

- Tutorial Structure : Formed by the library of bugs, for the relationship between the concepts and the objectives and for the remedial operators.
- Instructionals Materials: Formed by the library of examples, library of practices and test, library of definitions, library of clues, library of texts, megaprinciples library, counterprinciples library and library of results.

Tutor Agent is responsible for the following actions:

- To initialize agent: It consists of the registration and agent's Tutor connection in Router of JATLite
- Tutoring student: Tutor Agent develops a personalized instructional linear plan for the student at issue, and for each node of this plan, Tutor Agent decides together with Infomation Agent which instructional architecture will be adopted. Such architecture will be the metastrategy which defines the tutoring general lines of this node. Tutor Agent also active and it controls the tutoring process of each node of the generated instructional plan.

The Tutoring Module uses the concept of instructional planning to generate and to control the execution of the instructional plan to be introduced to a student. In agreement with [2], instructional planning is the process of sequence configuration of instructionals operations. If executed with success, they will result in the modification of the student's mental state in the effect desired. The system, through a global plan of instructionals objectives, interacts with the student and, if necessary, it reschedules his action course.

In the architecture of ITStrategic, the concept of the instructional planning is represented in the Tutor Agent's own behavior, through a mechanism of planning called linear Planner, and a mechanism and administration control called Executing. The Planner is responsible for the development of a teaching plan personalized to each student of the system. Executing bases itself on the plan generated by the Planner for the choice of the instructional architecture (metastrategy) of each node of this plan, as well as for the subsequent tutoring control.

For each node of the instructional plan generated, Tutor Agent will start a process with the agents of ITStrategic for the execution of one of the following actions:

- To Tutor directing architecture
- To Tutor architecture by discovery
- To Tutor exploratory architecture

Delivery Agents are responsible for the instructional dialogue between the system and the student and are formed by the Didactic Instructional Architecture Agent Society (DIAAS) and by the Instructional Event Agent Society (IEAS). This dialogue occurs through a communication network in two levels. In the first one it is specified, for DIAAS, the macro and microstrategies to be applied. In the second level, the microstrategies are executed by IEAS.

Each agent of DIAAS is known as Didactic agent, and each agent of IEAS as Event agent.

The Didactic Instructional Architecture Agent Society is composed of DidacticDirective, Didactic-DiscoveryOriented and DidacticExploratory agents, that can be activated by Tutor Agent (through bidding related to the tutoria of a concept), as a result of a process of negotiation of DIAAS, or by a student (in the case of Exploratory architecture):

The Didactic agents will be responsible for the following actions, with their respective subordinated actions:

1. to initialize the agent
2. to Tutor directive/oriented architecture for discovering/exploration

a. to structure pre tests presentation
b. to structure presentation of the goals
c. to structure presentation of the recovery of previous knowledge necessary to the learning
d. to structure presentation of the instructional material
e. to structure presentation of the tutorial attendance to the instructional material
f. to structure practice presentation
g. to structure presentation of corrective return of practice
h. to structure summary presentation
i. to structure post-test presentation
j. to structure presentation of corrective return of post-test
k. to structure presentation of retention activities and transfer

It is worth emphasizing that the action "Tutoring architecture" is related to each Didactic Agent.

The Instructional Event Agent Society operationalise the nine instructionals events proposed in Gagné [8]. Such events represent the instructionals microstrategies of ITStrategic, giving support to the internal processes of the student's learning.

The interface between the student and the system is executed by the Receiver agent, under Event Agents direct orientation. The Receiver Agent works with template, screens compositions pre-determinated for the allocation of instructionals objects. Filling these positions is personalized by Event Agents, Guide and Information Agents taken into consideration.

The execution of a microstrategy has the following effects:

- Presentation of an instructional event;
- Updating of the Student's Model;
- Movement for other microstrategy, when needed.

5 Conclusions

The article presents ITStrategic System, an ITS based on an architecture of agents able to dynamically define its instructionals strategies to increase the adaptability of the system to students.

Agent-based specification favours a high degree of abstraction. Each agent process performs particular roles besides their cooperative, colaborative and distributed participation.

Noteworthy is the fact that ITStrategic is a Multiagents ITS capable of dynamics determination along with its explicit goal (instructionals macro and microstrategies).

References

1. Beck, J., Stein, M., Haugsjaa, E.: Aplications of AI in Education. ACM Crossroadas. URL: http://www.acm.org/crossroads/xrds3-1/aied.html, September, 1996.

2. Brecht, B.,McCalla, G., Greer, J. E.: Planning the Content of Intruction. In: Bierman, D., Breuker, J., Sandberg, J. (eds), Proceedings of 4th Internacional Conference on AI and Education, Amsterdan, 24-26 May, p. 32-41, 1989.

3. Costa, Augusto C.P.L. Expert-Coop: Um Ambiente para Desenvolvimento de Sistemas Multi-Agentes Cognitivos. Tese de Mestrado. Florianópolis: Universidade Federal de Santa Catarina, 1997.

4. D'ambrosio, J.G., Darr,T., Birmingham, W.P.: Hierarchical Concurrent Engineering in a Multiagent Framework. In: Concurrent Engineering: Research and Applications (CERA) - An International Journal, v. 4, n. 1, March, p. 47-57, 1996.

5. Finin, T. et al.: Specification of the KQML Agent-Communication Language. URL: http:// cs.umbc.edu/kqml, 1993.

6. Finin, T. et al.: KQML: A Language and Protocol for Knowledge and Information Exchange. URL: http:// stm2.Kaist.ac.ke/witch/study/Agent/KQML/KQML.htm, 1994.

7. Frost, R.: Java Agent Template, Lite (JATLite), URL: http://java.stanford.edu, 1996.

8. Gagné, R. M.: The Conditions of Learning, 4th ed., New York: Holt, Rinehart and Winston, 1985.

9. Genesereth, M., Ketchpel, S. P.: Software Agents. Communications of the ACM, v. 37, n. 7, p. 48-53, 1994.

10. Marietto, M. G. B.: Definição Dinâmica de Estratégias Instrucionais em Sistemas de Tutoria Inteligente: Uma Abordagem Multiagentes na WWW. Tese de Doutorado. São José dos Campos: InITStuto Tecnológico da Aeronáutica, Fevereiro, 2000.

11. McCalla, G.: The Search for Adptability, Flexibility, and Individualization: Aproaches to Curriculum in Intelligent Tutoring Systems. In: Nato Asi Series, v. F85, Jones, M., Winne, P. H. (eds), Spring-Verlag, Berlim, 1992.

12. Sycara, K. P.: Multiagent Systems. AI Magazine, Summer,p. 79-92, 1998.