Knowledge-based system architecture for Distance Learning (KBS-DL)

under distributed artificial intelligence and case-based reasoning

Pedro Salcedo Lagos
Department of Educational Research and Information Technology,
Faculty of Education, University of Concepción
CHILE - psalcedo@udec.cl

Yussef Farrán Leiva
Facultad de Ingeniería, Dpto. de Ingeniería Informática
y Ciencias de la Computación, Universidad de Concepción
CHILE - yfarran@udec.cl

Summary
This study searches on design and setting up problems of a knowledge-based system for distance learning from the perspective of distributed artificial intelligence with MAS-CommonKADS methodology as background for both component and multi-agent description. In this way, the tasks of a tutorial intelligent system have been distributed in a set of independent items that cooperate each other with the only aim of guiding the teaching-learning process through an adaptation of teaching strategies used in the process.

Key words: Knowledge-based system for distance learning, knowledge modeling, intelligent agents, multi-agents, MAS-CommonKADS methodology, CommonKADS, chaos based strategy.

Introduction

Day after day, software intended to help improving educational process and administration of educational organizations of distance learning with high amounts of students is increasing. These kind of software do not seem to be indicating their utility, because teachers' dependence on the system is the same and even higher to what they had before to use it. Teachers must produce all material to be included in the system and concern about quality and copyrights involved. Dependence created by the systems of a permanent control on students must be added and activities that allow achievement of course aims have to be proposed.

This situation has driven to produce this paper, with the main objective of automating teaching-learning process produced between a distance learning skilled teacher and its students. In this case, general and reusable procedures for a shifting from these models to a system that supports teaching – learning process in a real and effective way, can be found. In other words, it is intended to achieve reasonably clear, complete and accurate descriptions of the tasks that both educator and students take into account when distance teaching learning process is performed. In this way, a model at knowledge level that links all different activities and relationships described by the students and the educator is about to be obtained. This model should allow getting a formal model underlying to this description and available to be linked with Knowledge Based System (KBS)

Proposed KBS allows teachers to produce a course that, based in specific objectives, allows detailing both materials and accurate teaching strategies for a "common" student. These strategies will be modify in time and will be adapted to users according to their needs. On the other hand, the system is able to guide students along particular domain of knowledge proposed by the teacher and solving, during the process, tasks such as creation of mentoring strategies, proposal of complementary materials, record of users activities and generation of new strategies for emerging cases.

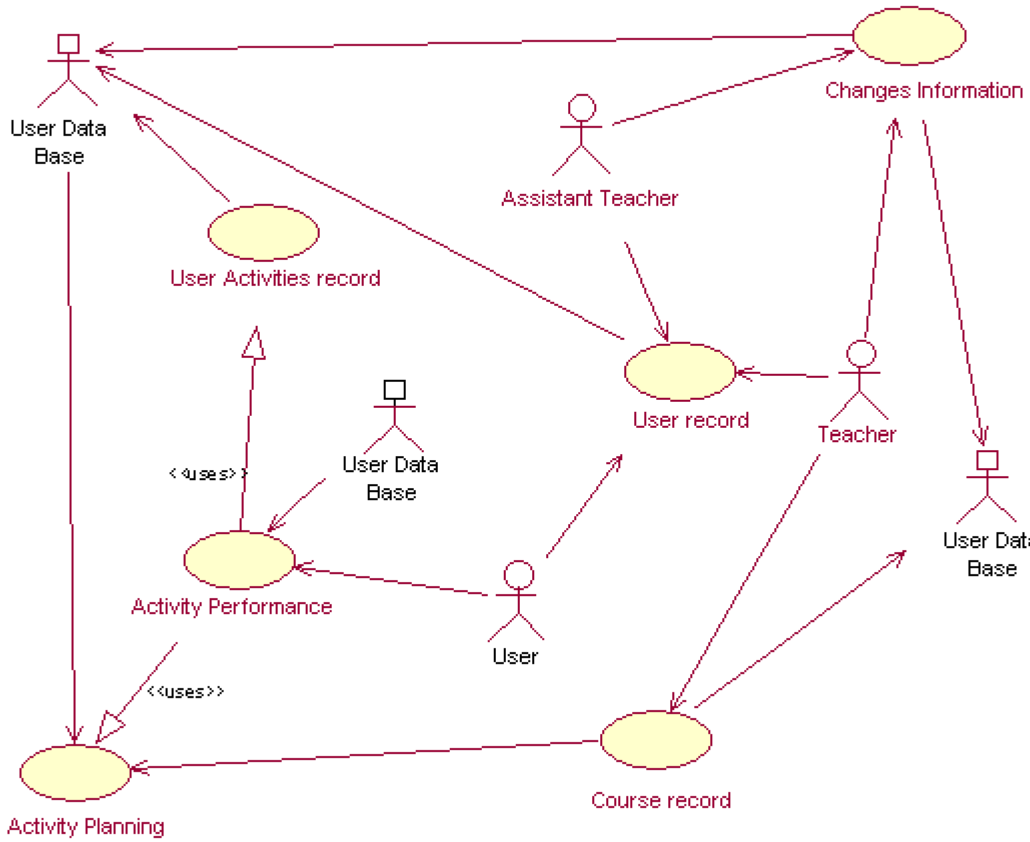
Focus on the process is presented by the means of a case model of using (fig. 1.1) where actors (teacher, assistant teacher, students, and course database and users database) interact with several processes. Under this focus, every module of those mentioned before will represent a set of one or more intelligent agents. These entities will cooperate with each other, with the aim of reaching the final objective of the system: knowledge transmission to the student and generation of courses in a more automated way through generation of teaching-learning strategies adapted to the students.

Distributed Artificial Intelligence (DAI) we will use are about to be defined as also the MAS-Common KADS methodology for system construction and we will finished when presenting basic elements of KBS analysis and design.

2. Distributed Artificial Intelligence

Iglesias (1998) defines DAI as "that part of the AI centered in collective intelligent behaviors resulting from the cooperation of several agents" These agents are collaborating entities.

System description from the viewpoint of DAI can be made according to two main elements (Dem, 1990). The first one is focused on fundamental components of the system as a composition of these ones. This is to say as the society our agents live in (Classic DAI).



An agent can be defined as an entity semi or absolutely autonomous, which acts rationally according to its perceptions from the outside and the state of its knowledge.

Society of agents is organized as a network in which nodes represent such entities and links, bridges information exchanges. From organization of links, information exchange can be made whether or not directly or indirectly by the means of an intermediary.

From control organization mentioned before, it can be said that there are two main kinds of societies in Distribute Artificial Intelligence: centralized and decentralized.

In the first case, centralized society, the autonomy of the agents is highly restricted, because there is a controller or omnipresent master that rules the state of the system during a problem solving in a special way and, in some cases, in a total way. This is the only one capable of sending commands to subordinated agents or slaves in order to perform all necessary activities to carry out the solution to the task. In this centralized model, there is a high degree of coherence in the distribution of tasks, though the system becomes closed because of centralization itself.

De-centralized society, agents have maximum autonomy, there is not a central controller emitting commands and organizing cooperation, but every agent knows its own skills and those of the rest and knows when asking for help. By means of these knowledge, every agent can decide which part of the problem can deals with and who asks for help. Distribution of tasks becomes, in this model, less coherent, but the systems is more open.

MAS-CommonKADS

MAS-CommonKADS is a multi agent system oriented methodology (Iglesias, 98), an extension to CommonKADS (KADS, 93). With this methodology it is possible to describe the whole system, based in graphic representations and templates to be filled in by the user. It uses graphic representations and templates to represent models that cover every one of the aspects of the system.

MAS-CommonKADS uses the following models for developing multiagent systems:

- **Agent Model (AM):** it specifies features of an agent, its reasoning capacities, skills, services, sensors, effectors, group of agents to whom it belongs and kind of agent. An agent can be human, software or any entity capable of using an agent communication language.
- **Organization Model (OM):** This is a tool used to analyze the human organization in which multiagent is about to be introduced and to describe the organization of software agents and it relationship with its environment.
- **Task Model (TM):** It describes the tasks that agents can analyze, the objective of every task, its decomposition, ingredients and methods of problem solving for every objective.
- **Experience Model (EM):** It describes knowledge needed by agents in order to achieve their objectives. It follows the decomposition of CommonKADS and it re-uses the libraries of generic tasks.
- **Communication Model (CM):** It describes interactions between a human agent and a software agent. It focuses on considering human factors for such interaction.
- **Coordination Model (CoM):):** It describes interactions among software agents
- **Design Model (DM):** While other five models refer to a multiagent analysis, this model is used to describe both architecture and design of multiagent system as a previous step to implement it.

Life cycle model proposed to develop these tasks in the methodology is the spin model directed by risks, following project management of CommonKADS. For small projects, the reutilization of waterfall model is proposed.

3. System construction

Because of the extension of providing a detailed description of every model and entity, we will focus on describing only a few models and agents which represent, according to us, the main component of a KBS-DL.

3.1 Organization model

The constituting process indicates the temporary order in which functions of the function component are performed, is indicated in fig. 3.1

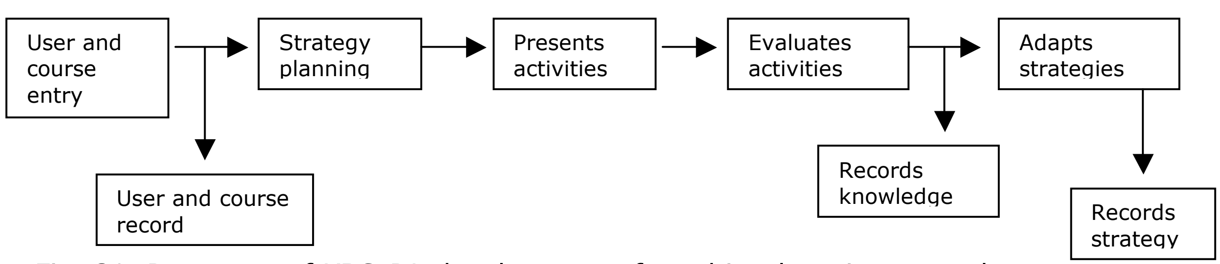


Fig. 31. Processes of KBS-DL development of teaching-learning control process.

Introduction of multiagent system will make detection of the best strategy for the aim and the user, faster. In the same way, the development process of a distance course and the learning process of the user will also be faster.

3.2 Conceptuation

3.2.1 Actor's identification

In our system, several actors that interact with the system can be identified: Teacher, student, users database, courses database and assisting teacher. However, we can consider the necessity of access to the system for development staff in order to up-dating it or for maintenance.

3.3 Agent Model

The purpose of the agent model is describing agents participating in problem solving and the subsecuent outcome of the system on human agents.

3.3.1 Identification of agents (first iteration)

Starting from actors in the conceptuation stage, external agents to the system can be identified: student, teacher, assistance teacher, course database and user database. Databases will be controlled by the system and for the rest of the agents, which are humans, it will be necessary to create a different graphic interface for every one, because in the case of the student, this one will interact only with proposed activities. On the other hand, teacher will enter objectives, activities and strategies and the assistance teacher will modify information only.

All these agents apart from each other exclusively through the cognitive component, which it allows to determine assigned tasks within society, as also to display a specific behaviour. If we describe the pedagogic agent, we will find:

Pedagogic agent:

The main objective of pedagogic agent is generation of instructional plans (teaching strategies) adapted to current needs of students. That is to say, agent must be able to work as dynamically as to detect as fast as possible the needs of the student and, according to them, change the teaching strategy.

The general idea for construction of reasoning capacities of this agent has its origin in Reasoning-based cases (RBC). RBC can be defined as: "Generation of solutions to current problems on the base of past plans, which are stored in memory as cases. Cases which features match with current situation and applying strategies in order to adapt them to current needs. For this situation, it is necessary a metric of similitude

In our case, the working memory is constituted by a set of elements called trajectories. The structure of these strategies is as follows:

$(ID,(Obj_{i,l}, SubPlan_{i,l}, Aprov, Term),...,(Obj_{i,n}, SubPlan_{i,n}, Aprov, Term))$

where:

ID: Student identification

$Obj_{i,k}$: work topic

Aprov: Relationship between correct activities and total amount of answers

Term: It determines if working topic was completely defeated or if the work was interrupted.

$SubPlan_{ik}$: Vector of the form:

(present concept1, propose media, ..., present $concept_k$, propose med_k)

$concept_i$: Concept to be taught

$medios_i$: List of exercises to be proposed

These trajectories can be divided in two classes: Standard trajectories, included in the working memory from the beginning and applicable to any student and, general trajectories during the work.

Metric of similitude accomplishes a filtering function in two steps: First, those trajectories that indicate a general profit lesser than 75% or do not include the working topic, are removed from the searching space. The second filtering selects between the set obtained in the previous step, the most similar trajectories to the current student. This process, can be described as follows:

1. For the student trajectory, the V_0 vector is obtained, so:

   $V_0 = (Obj_1, concept_{1,1}, ..., concept_{1,m}, ..., Obj_k, concept_{k,1},... , concept_{k,m})$
2. For every one of the trajectories obtained from the second filter, a Vi vector, similar to V0 must be obtained and the following relation with every one of them must be formed:

   $$R = \frac{\text{Total of coincident indexes}}{\text{Vector's length}}$$

   Where an index represents a component of the vector and its length is the total amount of components.
3. Previous $R_{V0,Vi}$ in a resulting list L_{sim}. Must be placed in a descending order.

According to this algorithm, it is clear that the most similar trajectory to the analyzed one, will be the corresponding to the first element $L_{sim.}$ From this trajectory, we will take the instructional subplan, corresponding to the working topic, which it will be applied to the student.

During the working session, which it constitutes an efficiency test for the selected subplan, it will be exposed to variations made through mending strategies. Changes are made from standard trajectory components and those contained in L_{sim}, which are combined having into account the degree of similarity and the effectiveness reached by the student.

In this way, the instructional plan of a particular student it is dynamically generated starting from: subplans contained in the working memory and variations to these plans according to strategies. Final subplan is stored in the student's trajectory. Fig. 3.2 indicates the whole process.

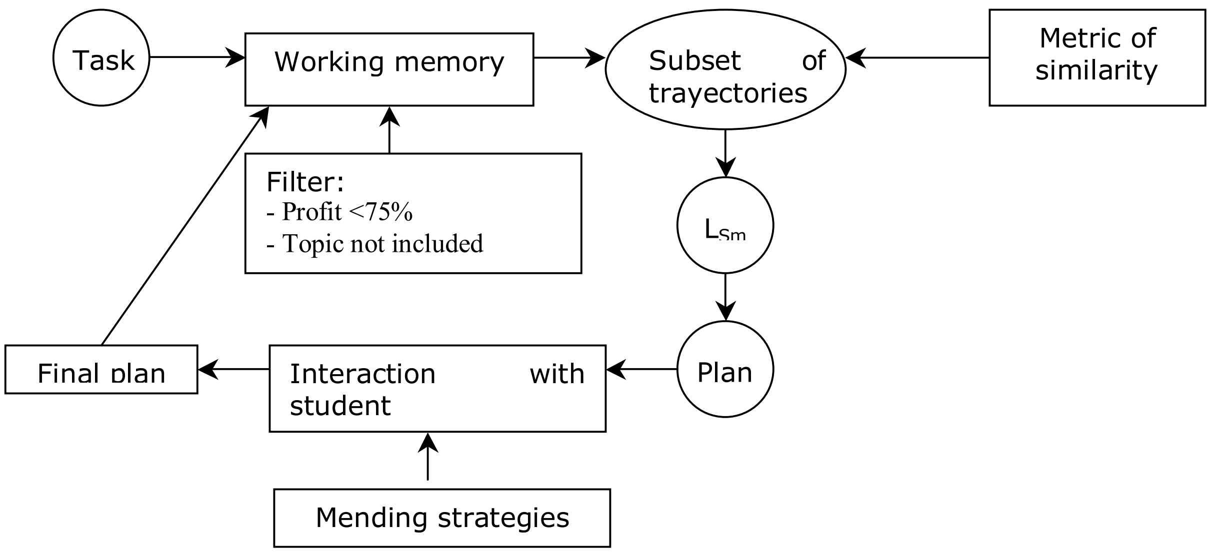


Fig. 3.2: Arrangement of a teaching strategy process

3.4 Design Model

Because of the flexibility of this platform for adapting teaching strategies, courses defined with it are adapted to students, having into account its own psychosocial features, as the set of actions performed during the teaching process. There is a structure, associated to every student in its interaction with the system restored at the beginning of the session. Courses managed by the system are defined in terms of teacher tasks and rules. Rules identify relationships among tasks which, in turn, correspond to conceptual units defined by the teacher of the course.

The architecture of the system is based on a standard Web model, in which the server receives requests from the students through browsers.

There is a process for every one of the students connected into the system, which controls student learning during the whole session. If one student is performing more than one course, there will be, therefore, one process for every course. The main set of programs and data used by the system, is indicated in fig 3.3.

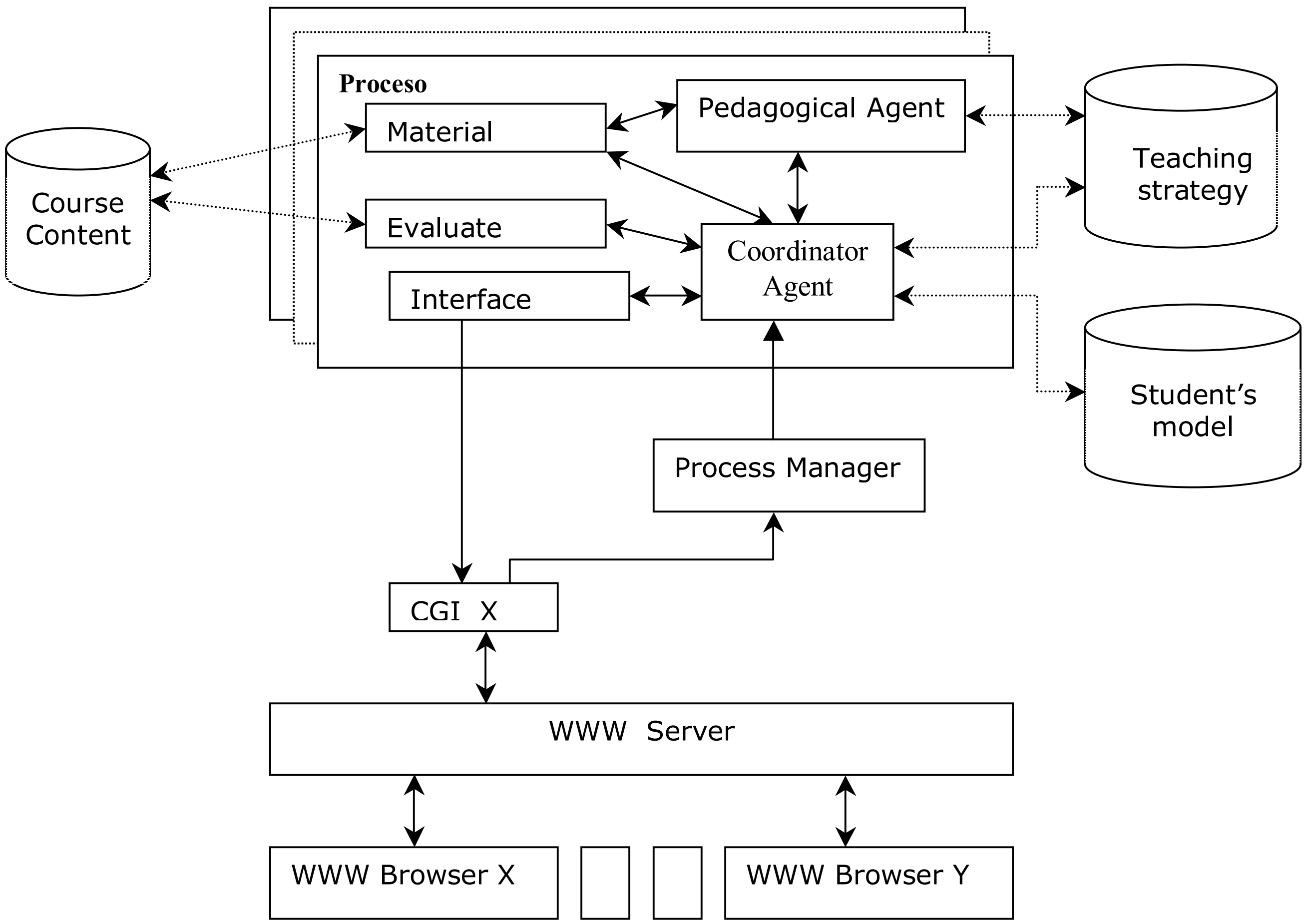


Fig. 3.3 System design

The main modules of the system are those managed by the coordinator agent and they are placed in the server in which students are connected. These, are accessible through a CGI program.

For the system to be operated, process manager must be always active, waiting for requests from students.

When the CGI receives a request, it sends to the process manager these parameters and stays still until an answer is given from the interface agent. When generated HTML pages are received, they are sent to the student.

Whether the strategy selected by the student for learning or personal data of the same students, are obtained by the means of a test presented before the students during the first time they access to a course. Data are stored by the system to be used during the current session or for incoming sessions. Coordinator Agent provides information about actions performed by students and their results (amount of pages visited, activities carried out, amount of exercises solved correctly, etc.). This information is delivered to the Pedagogic Agent, who generates the strategy according to reasoning based on the most effective case (RBC). Coordinator Agent, in contact with

Material Agent, will determine activities to be performed by the student in time and space according to his/her profile.

On the other hand, it is the Evaluator Agent who, in contact with course contents and with Coordinator Agent, will determine evaluations, more accurate to a specific objective, according to the user, as not achieved.

Finally, Coordinator Agent sends to Interface Agent those parameters associated to the active task to be used to, dynamically, generate HTML pages to be recieved by the student. Such parameters are related to the profile and actions of the student.

5. Conclusions

Knowledge-based system for distance learning (KBS-DL) allows creating and chasing adaptive courses through Internet. Both structure and course content are managed independently, making adaptation mechanisms to every student easier, as also the reutilization and information maintenance. Automatic generation of pages makes possible maintenance of the coherence in pages of the menus, even after changes on structure of the course have been made.

Structure of every course is composed of a set of objectives, tasks and teaching rules, while activities and multimedia elements used to generate dynamically HTML pages presented before the students form contents.

This system represents a notable advance in the adaptation capacity of the teaching strategy and in the adaptation capacity of the courses developed, by allowing dynamic choosing of activities intended to achieve a goal.

This vision is crucial for creating intelligent educational platforms, because it provides an interactive environment of great reaction capacity. This is achieved because of the distribution of tasks, to agents who work concurrently. While students performs the final exercises of some subject, the Planning Pedagogic Agents deciding the following teaching strategy and the new concepts.

References

[Ber,92] S. Berthet, Y. Demazeau, O. Boissier, "Knowing each other better", 11th International Workshop on Distributed Artificial Intelligence. Glen Arbor, Febrero 1992.

[Dem,90] Y.Demazeau & Mueller, "Descentralized Artificial Intelligence", en "Descentralized AI", Demazeau & Mueller eds., North-Holland/Elsevier, pp 3-13,1990.

[Dem,92] Y. Demazeau & O. Boissier, "A Distributed Artificial Intelligence View on General Purpose Vision Systems", en "Descentralized AI 3", Demazeau & Werner eds., North-Holland/Elsevier, pp 311-330, 1992.

[Dem,93] Y.Demazeau, "Distributed Artificial Intelligence & Multi – Agent Systems". Conferencia impartida en el X Simposio Brasileño de Inteligencia Artificial, 1993.

[Iglesias,96] C.A. Iglesias, J.C. González and J.R. Velasco. "MIX: A general purpose multiagent architecture" en Intelligent Agent II: Agent Theories, Architectures, and Languages, volume 1037 of Lecture Notes in Artificial Intelligence, pages 251-266. Springer-Verlag, 1996.

[Iglesias,98] C.A. Iglesias, "Definición de una Metodología para el Desarrollo de Sistemas Multiagentes. Tesis Doctoral", Febrero 1998, Departamento de Ingeniería de Sistemas Telemáticos de la Universidad Politécnica de Madrid.

[KADS,93] R. De Hoog, R.Martil, B. Wielinga, R. Taylor, C. Bright, and W. van de Velde. The CommonKADS model set. ESPRIT Project P5248 KADS-II/M1/DM..1b/UvA/018/5.0, University of Ámsterdam, Lloyd`s Register, Touche Ross Management Consultants & Free University of Brussels, Dec.1993.

[Mor,94] F. Morales, "Sistemas Tutoriales Basados en Casos", en Informática 94, Cuarta Convención Internacional y Feria de Software, La Habana, Cuba, Febrero 1994.

[Mor,95] F. Morales, "Implementando Cooperación en un Mundo MultiAgente". Reporte Interno de Investigaciones. Universidad de La Habana, Fac. Matemática y Computación. 1995.