Multi-Agent Framework for Intelligent Questionnaires on the Web

Fernando Alonso, José M. Barreiro, Sonia Frutos, César Montes
Facultad de Informática, Universidad Politécnica de Madrid
28660 - Boadilla del Monte, Madrid, Spain
{falonso, jmbarreiro, sfrutos, cmontes}@fi.upm.es

ABSTRACT

This paper presents a multi-agent framework for conducting an intelligent survey on the Internet network through Web capabilities. The advances in multi-agent technology and network computing form the basis of this framework. Performing questionnaires on the Internet has the drawback of network traffic, which, however, is more than offset by its advantages, including distribution, communication, platform independence.

We have developed an agent-based framework to solve the questionnaire problem that increases reusability and understanding. Each stage of the questionnaire is assigned to an autonomous agent working alone or with other agents forming a society. The underlying knowledge of questionnaires and decision rules is transformed into objects belonging to an agent or an agent society. Agents can interact with each other through the interaction process, providing a dynamic, interactive and evolving system. All these components are located in an environment.

Finally, an application using this framework, the ALBOR system, is described. The ALBOR system (Barrier Free Access to the Computer) was conceived to publish intelligent questionnaires on the Internet evaluating the physical, sight, hearing and cognitive aptitudes of computer users with disabilities and give advice about technical aids or adaptations depending on their evaluated capabilities.

Keywords: Multi-Agent System, Blackboard System, Interactive User Interface, Web-based Computing, Intelligent Questionnaire.

1. INTRODUCTION

This paper seeks to describe a framework for developing interactive systems on the Web. This framework has been designed taking advantage of multi-agent technology. The disadvantages of network computing were solved using the agent concept and providing intercommunication mechanisms. As a result of this, multi-agent and networking technologies were used as the basis of this framework. Due to the complexity of designing a generic framework for interactive systems, we opted to solve the questionnaire problem using this field of knowledge, because it is a typical interactive problem that can be tackled in distributed environments, like the Internet.

An introduction to multi-agent systems and their relationship with Internet computing is given in the next section. The main goals and possible phases of intelligent questionnaires are discussed in section 3. The multi-agent framework for intelligent questionnaires on the Web and its main components are discussed in section 4. In section 5, an application that uses this framework is explained and briefly described. Finally, section 6 draws some general conclusions.

2. MULTI-AGENT SYSTEMS ON THE WEB

Distributed Artificial Intelligence (DAI) is one of the up and coming fields in Artificial Intelligence. Multi-Agent Systems (MAS), a field of DAI, and above all the agent concept [1,2,3,4], are achieving the biggest successes. The property of distributed systems allows parallel execution, provides robustness and offers easy expansions. The autonomous mode of the agents gives them independence and decision-making capabilities. The intercommunication among agents allows them to meet, communicate, collaborate and co-operate to reach a common or particular goal. All of these characteristics have contributed to the success of MAS.

Let us define, then, a MAS as a computer system composed of several agents capable of interacting with each other and with the environment in which they are located. A MAS is not only a set of agents; the environment and the possibility of communication among agents are active components [5]. They, therefore, have to be very much taken into account. So, the key components that we are going to consider in a MAS, are:

- The environment: the place where agents and objects are located, which determines and conditions their behavior.
- The agents: the active entities of the system, that is, those that have the capability to act in the environment.
- The objects: the passive entities of the system, that is, those on which the agents act. The passive property of the objects does not imply that they are static. They can change as a result of the actions of one or more agents.
- The agent-agent and the agent-object interaction: the link component between agents and objects distributed in the environment. It is the means used by agents to act, to get to know each other and to interchange knowledge. It is also the reason why agents evolve and objects vary in time.

Advances in communications and the popularization of the Internet network provide a powerful, distributed and universal platform for the development of MAS. It is important to point out the following characteristics of the Web:

- Distribution: it is possible to place agents at different servers located in distant places in order to rationalize resources and allow execution parallelism among agents.
• **Communication**: to use communication links so as to avoid the isolation of agents.
• **Independence**: to program agents in operating systems and platform independent languages so the system can be accessed by any computer type and with any operating system using only a standard Web navigator.
• **Mobility and variability**: thanks to hardware independence and communication links, agents can travel from one computer to another, either keeping their behavior or mutating as they move.
• **Lack of knowledge**: computers connected to the network and, therefore, the agents they accommodate do not directly know about the existence of the other connected computers. These machines are only accessible via Domain Name Servers (DNS) and using communication protocols.
• **Ease and possibility of use**: due to the popularization of the Internet, any user with an Internet access can connect to and easily work with these systems, without noticing what there is behind, thus providing transparency to the user.

A MAS on the Web [6] has to exploit the above Web services effectively. The coordination mechanisms supported by these systems change as regards traditional MAS.

### 3. INTELLIGENT QUESTIONNAIRE

Computers have totally changed the traditional way of answering survey questions. An electronic questionnaire can be more dynamic and evolving, can optimize the time required for completion by answering only pertinent questions, can be evaluated incrementally obtaining an approximate evaluation without it having been finished, and can have other associated media, like sounds, images, videos or animations.

The desired properties of an intelligent computer-based questionnaire, are listed below:

• **Dynamic**: the order of the questions is not fixed in advance and depends on the answers to the previous questions.
• **Sequential and non-linear**: the questionnaire questions are distributed sequentially due to the requirements of current evaluation models and also because the human brain processes questions one by one. But the questionnaire will not be limited to a linear execution. In each question and/or answer, written text complements can be consulted, intermediate evaluations can be made before finishing the questionnaire or the user can go back to previous questions to change an answer.
• **Incremental Evaluation**: the possibility of asking questions and evaluating results in parallel allows the questionnaire to be evaluated at any point.
• **Customised**: each user fills in a different questionnaire even though all of these are based on the same model. Moreover, this customised questionnaire contains the minimum number of questions required to evaluate all the aspects contained in the questionnaire goals.
• **Interactive**: the communication between the user and the system is intrinsic whether the questionnaire is electronic or manual. Nevertheless, the interaction should be more natural and closer to the user with this kind of systems.
• **Adaptable**: if the type of user is known, the system can be adapted to her/his particular needs, changing the form of the questions and answers, associating media, that is, more classical or more modern, using more or fewer help messages, or even improving the visual, sound and tactile appearance.
• **Scalable and Modifiable**: the questionnaire and results evaluation models make it possible to add, modify and delete questions, answers, associated media, evaluation rules and recommendations.
• **Understanding**: the use of intelligent models increases the understanding of the underlying knowledge and the execution models used for the experts.
• **Multimedia**: each question has different associated media (visual and sound) to ease understanding, ruling out the ambiguities of written language.

The operating mode of an intelligent questionnaire system is determined by the above characteristic factors of these systems. When a user enters the system, she/he must identify herself/himself in order to validate her/his login, to customize the session and also to save its state to continue later. Then, a system session starts. A session is structured in several stages (Figure 1) described below:

1. **User Identification**: user personal particulars and other information are collected in order to start the session.
2. **Session Preparation**: the user is informed about the goals of the questionnaire, how the session will be performed and whether any preliminary training is necessary.
3. **Survey Taking**: the questionnaires can be structured in several groups. In each group, the user is asked several questions, depending on the answers to earlier questions, and only those needed for her/his evaluation.
4. **Result Evaluation**: after collecting all questionnaire data, an evaluation report is sent to the user. This evaluation can contain only one recommendation or even several alternatives (sorted by priority) in order for the user to decide which is best suited for her/him according to objective or subjective criteria.
5. **Performance Analysis**: user performance is analyzed and evaluated to customize and adapt the interaction process throughout the entire session (from beginning to end).

![Figure 1. The Execution of an Intelligent Questionnaire System.](image-url)

Although the order of the stages could suggest linearity, this is only used to clearly describe system operation. During
execution, stages 3 and 4 can be performed in parallel (Figure 1) to get results in intermediate states and to cut the evaluation time. Moreover, stage 5 (performance analysis) is performed during the entire life cycle of the session.

4. MULTI-AGENT FRAMEWORK FOR INTELLIGENT QUESTIONNAIRES ON THE WEB

Multi-agent technology provides the means and tools to develop this kind of system. We have developed an agent-based framework to solve the questionnaire problem that increases reusability and understanding. Each stage of the questionnaire is assigned to an entity capable of solving this stage relying on its own knowledge alone (autonomous agents). Autonomous agents allow independence and parallelism between stages. Any agents can work together so as to reach common knowledge (agent society). The underlying knowledge of questionnaires and decision rules is transformed into objects of the environment, which can be owned by an agent (internal agent knowledge) or the agent society (common society knowledge) and is used by these entities in information, survey-taking and decision-making processes (performances of agents in the environment). Agents can interact with each other (interaction process), providing a dynamic, interactive and evolving system. All these components are located on the Internet network (environment), taking advantage of its properties.

In short, our framework is composed of five key elements: the agents, the agent societies, the objects, the interaction and the environment, which are described in the following subsections.

The Agents

The agents are the active entities of the framework, that is, they can act in the environment. Each agent is responsible for one or more system tasks and makes use of internal knowledge stored in a knowledge base. The agent that interacts with the user allows her/him to be considered as an active component of the framework and is also responsible for performing the analysis task [7].

First of all, the agent structure will be described. This structure is the result of the agent definition as an active entity (with performance capabilities). The aspects that describe an agent are as follows:

- **Identifier Name**: this name allows an agent to be identified in the environment.
- **Goal/s**: a goal is the reason why an agent acts. The agent acts in pursuit of a goal.
- **Task/s**: if several tasks are completed successfully, each goal will be achieved.
- **Proactive Performance**: agents perform actions as a result of merely internal reasoning processes and internal knowledge. These actions are not conditioned by anything outside the agent. Therefore, the proactive performance of an agent is described by:
  - Internal Knowledge
  - Reasoning Process
  - Actions
- **Reactive Performance**: agents perform actions, as a result of perceiving external knowledge. This collection of “perception-action” pairs represents agent reactions in the face of external stimuli or events, albeit inside the environment. Therefore, the reactive performance of an agent is described by:
  - External Knowledge
  - Perceptions
  - Actions

- **Interactive Performance**: the agent is in touch with the external world through relations with other agents in the environment, either relations established a priori or that change over time. Therefore, the interactive performance of an agent is described by:
  - Interaction Agents
  - Relations

These points underline the idea of the multi-agent system. Agents in a multi-agent world are not isolated items, they are located in a changing and evolving environment, where they can interact with other agents and objects.

Secondly, the life cycle of an agent (Figure 2) will be explained in order to understand its performance. The agent can start life at the beginning of the system (predefined agent) or later when it is created by another agent. When the agent is alive, a cyclic process begins that is repeated until the agent dies. The stages of this cycle are as follows:

- **Perceive**: the agent perceives external stimuli and messages from other agents. In the first case, the next stage will be the Act Stage, and in the second case, the Reason Stage.
- **Reason**: the agent begins the reasoning process using Internal Knowledge. This stage can follow on from the Perceive Stage (interaction performance) or otherwise (proactive performance).
- **Act**: as a result of the Perceive Stage (reactive performance) or the Reason Stage, the agent acts on the environment, which it modifies, for instance, by sending a message to another agent, generating an event or creating an agent or an object.

And, finally, if the agent achieves its goals, then its life ends.

![Figure 2. The Life Cycle of an Agent.](image-url)
agents in order to group MAS and Internet features. Control and domain agents are equivalent to server agents on the Web and interface agents to client agents. This general and simple classification of agents means that other agents can be included in the server agent group.

Fourthly, the agents of the framework will be identified. The Multi-Agent Framework for Intelligent Questionnaires is composed of six agents: the Session-Controller Agent, the Session-Initializer Agent, the Decision-Maker Agent, the Advisor Agent and the Interface Agent (Figure 3).

The Interface Agent is the only agent located in a client computer and is responsible for interaction with the user. The other agents, which perform the system reasoning, are housed in a server computer or are distributed across several computers. In the latter case, a DNS agent is needed to provide the physical location of agents and objects. In this manner, agents and objects can change their location in the future or can be transformed into mobile agents that travel from one machine to another in order to improve their behavior.

According to the agent life cycle description, there are two possibilities: first, an agent comes to life when the system starts up and, second, an agent comes to life when created by another agent. In our framework, the only agents present when the system starts up are the Session Controller Agent and the Interface Agent. When a user logs into the system and the Session Controller Agent okays her/his entry, this agent generates a copy of the other agents and a copy of the Interaction Blackboard Object (that will be described later), that is, creates a copy of the Survey Society (that will be described in the next subsection) for the user in question.

The agents of the framework are shown in Figure 3, and more information is listed below.

**Session-Controller Agent**: it decides, based on a login and a password, whether or not a user enters the system. This agent offers extra security, giving access only to authorized users. If the user enters the system, the other agents are created to begin a session with the system.

**Session-Initializer Agent**: it collects user personal particulars and other information in order to start the session.

**Survey-Taker Agent**: it completes questionnaires structured in several groups. In each group, the user is asked several questions depending on the answers to earlier questions, that is, the agent dynamically generates the questionnaire, which is customized for each user, and asks only those questions required for evaluation. Its internal knowledge (the questionnaire) is represented by a Questionnaire Graph Object.

**Decision-Maker Agent**: it makes decisions based on the answers to the questionnaire. Its internal knowledge is represented by a Leveled Rule Object.

**Advisor Agent**: the system evaluates different aspects of the user based on her/his responses in order to give her/him the best suited recommendation output. It expresses a list of recommendations arranged by priority.

**Interface Agent**: it analyses and evaluates user performance in the system, for the purpose of characterizing and adapting interaction.

**The Agent Societies**

Agents are embedded in a dynamic and evolving environment, but this environment is characterized by its social peculiarities. Agency, agent societies or artificial societies [8,9,10] are concerned with a subfield of AI introducing social features into agents in a multi-agent world.

Not all multi-agent system consider agent societies. We think agent societies offer an abstraction level upon the agent level which allows the solution to be more clearly structured. A society is a collection of agents characterized by having one or more goals in common, sharing objects of the environment, and using a secret communication language. But that does not mean that the society is closed. The society components can belong to other societies, communicate with other agents which are not located in the same society, leave the society or travel to another.
In our framework all server agents (except the Session Controller Agent) form a single society, Survey Society, as shown in Figure 4. This society is the owner of a common object, the Interaction Blackboard Object, which will be described in the next subsection.

The Objects

The objects are the passive entities of the framework, that is, they are the things on which the agents act. The passive property of the objects does not imply that they are static. Objects can change as a result of the actions of one or more agents. We have considered two kinds of objects.

Internal Objects: objects that belong to an agent and are accessible only to this agent. These objects represent the internal knowledge that the agent uses to reason.

In our framework, each agent contains one internal object represented by a knowledge base. These Knowledge Objects are the basis for the internal reasoning process of an agent. The second use is when an agent perceives an event in the environment, it searches its internal knowledge for whether and how it has to react. Finally, this knowledge is also used in the interaction process among agents.

All agents need internal knowledge, but this knowledge can be expressed in different representations. The internal representation or the Knowledge Object of agents are briefly described below:

- **User-History Object**: it belongs to the Session Controller Agent. Information about authorised users (for instance, logins, passwords and information about the last session) are saved in the History Object. Each user will have just one entry in the history. If the user enters the system more than once, her/his information in the History Object will be updated.

- **Session-Information Object**: it belongs to the Session Initializer Agent. Textual and graphical information about the session are organized in a sequential list that the user has to check in order.

- **Questionnaire-Graph Object**: it belongs to the Survey-Taker Agent. Questions are represented as nodes and are linked with answers (graph transitions). Associated media, like images, videos and sound are matched with semantic actions associated to a node or transition. A child node outputs the next question under the parent node question, if the user gives the answer stated in its transition. If questions are grouped in several parts, each part will have one associated graph.

- **Leveled-Rules-Base Object**: it belongs to the Decision-Maker Agent. Rules are classified into two levels from a level close to the questionnaire answers to a level close to the recommendations. Each rule is composed of answers, intermediate facts and final recommendations, combined with AND and NOT operators. The consequents of first level rules (intermediate rules) are intermediate facts and the consequents of second level rules (final rules) are final recommendations. If questions are grouped in several parts, each part will have its own level.

- **Priority-Recommendation-List Object**: it belongs to the Advisor Agent. A recommendation is the advice that the system outputs in a particular situation and for a particular user. Therefore, a recommendation represents a situation type. All of these are classified by priority, so as to give the user the order when more than one output is possible.

- **User-Internal-Model Object**: it belongs to the Interface Agent. This object models user peculiarities and necessities; for instance, the expertise level (beginner, intermediate or advanced) of the user in handling the system.

Figure 5 shows the Questionnaire-Graph Object of the Survey-Taker Agent, where question nodes and answer transitions are shown graphically.

External or Common Objects: objects belong to the environment or to the agent society and are accessible to all agents or society agents, respectively.

In our case, there is a common object belonging to the Survey Society, the Interaction-Blackboard Object. A blackboard is opportunistically and concurrently accessible, data sharing structure, and builds the solution parallelly and incrementally. It is a domain blackboard without control, because it is the agents that have to guide and control blackboard filling.

The Interaction-Blackboard (Figure 6) is divided into several levels (user information, questionnaire answers, intermediate facts and final recommendations) so that each agent accesses the right level.

All agents of the society access the blackboard levels, either to read or to write information. Depending on the agent’s task, it only has access to one particular level and with specific attributes (reading, writing or both). For instance, the Decision-Maker agent needs a read permission to get the answers of the questionnaire (answer level) and a write permission to state its answers.
decisions in the intermediate and the recommendation level. The Survey-Taker agent also accesses the answer level, and hence, these agents are communicated through this level of the blackboard.

Another restriction is that each level can only be accessed by one agent with write permission in order to allow concurrent access to the blackboard. If there were two agents with write permission for a specific level, the blackboard would have to implement blocking mechanisms to avoid knowledge inconsistencies.

**The Agent-Agent and the Agent-Object Interaction**

The interaction process is the most important feature of multi-agent systems. The interaction is the link between agents and objects distributed in the environment. It is the means used for agents to act, to get to know each other and to interchange knowledge. It is also the reason why agents evolve and objects vary in time. We have identified two interaction types, depending on the participating items:

- **Agent-Agent Interaction**: it allows agents to get in touch with each other. An agent can interact with another in the same or different society, with a server or client agent or with an isolated agent. Two types of agent-agent interaction are identified (Figure 7) below:

  - Direct Interaction: when two agents communicate without using any intermediate element. A means of communication can be a communication language that allows information interchange through messages and based on a communication protocol (for instance, KQML [11]). Examples of language primitives are: establish connection, interrupt connection, send message, send query, send response, and ask for confirmation.
  - Indirect Interaction: when two agents communicate using any object of the environment and without being aware that they are communicating. To access to this object, the agent-object interaction will be applied by all intercommunicated agents. This interaction is only possible when agents belong to the same society and, therefore, the object will be a common object of this society. This restriction rules out object misuse.

In the framework, the agents that are located in the server forming the Survey Society interact indirectly through a common object, the Interaction Blackboard Object. Some of these agents and another server agent (the Session Controller Agent) interact with the client agent (the Interface Agent) directly using messages.

**Agent-Object Interaction**: it allows agents to collect information from the environment when the object is external or shared, and is used to reason when the object is internal.

In our framework, all the agents have their own Internal Knowledge Object that they manipulate with their own mechanisms. Agents of the Survey Society use the Interaction Blackboard Object to communicate with each other. Therefore, this second Agent-Object Interaction is hiding an Indirect Agent-Agent Interaction. The Interaction Blackboard has no predefined knowledge (it is empty at the beginning). Furthermore, the Interaction Blackboard provides useful information to the agents by which it is accessed, but merely because other agents have left information there. This blackboard contains no valuable information, except the structure of the solution. This structure is filled by the agents.

Figure 8 shows the interaction between two agents (Survey-Taker Agent and Decision-Maker Agent) of the Survey Society and the Interaction Blackboard. The agent uses a simple language to access the blackboard. This access language consists of two primitives, read and write. The read primitive is used by the agent when it wants to get information from any level of the blackboard. The write primitive is used when the agent wants to leave information. This scenario shows how these agents share knowledge through the blackboard. The Survey-Taker Agent writes the answers to the questions answered by the user. The Decision-Maker Agent accesses the blackboard to get the answers which will fire its decision rules.

When a rule is fired, this agent writes the rule consequent in the blackboard. The Decision-Maker Agent may also access the blackboard to get additional information from the agents that manipulate the blackboard. The Interaction Blackboard Object contains predefined knowledge (it is empty at the beginning). Furthermore, the Interaction Blackboard Object provides useful information to the agents by which it is accessed, but merely because other agents have left information there. This blackboard contains no valuable information, except the structure of the solution. This structure is filled by the agents.

**The Environment**

The environment is the place where agents and objects are located and determines their behavior. These agents can work independently or in groups due to the environment properties. In our framework, we use the Internet network and the computers connected to it as an environment. Any system based on this framework is accessible from any computer with Internet access through any Internet navigator.
A collection of agents that execute the system task are placed in the environment. Agents are located at one or more servers, with the exception of the agent that interacts with the user, which travels to a client computer. Some server agents form just one society in which they do not communicate with each other directly, but through an object of the environment which is owned by the society, and, hence, they are not aware of the existence of other society agents. However, some server agents (whether or not they are members of a society) have to contact the client agent, since they need information in order to perform their tasks. This other communication is carried out by message passing after establishing a connection between the above agents. Therefore, one agent can work autonomously and together with others in the same society.

Figure 9 represents the entities and macro-entities present in the environment of a MAS. We have simplified the figure grouping the server elements in one box, but they can be distributed across different servers. As depicted, this environment framework can be applied to other systems. It would be necessary to choose the right agents and objects, to assign objects to agents, to decide whether the societies are suitable for the problem and, if so, to include agents as societies members and to assign objects to societies, and, finally, to establish the interaction between agents, objects and societies.

**Figure 9. Components of the Environment.**

5. FRAMEWORK APPLICATIONS

This framework has been successfully applied to the ALBOR system. The main goal of the ALBOR system (Barrier Free Access to the Computer) is to publish intelligent questionnaires on the Internet evaluating the physical, sight, hearing and cognitive aptitudes of computer users with disabilities and give advice about technical aids or adaptations depending on their evaluated capabilities.

The knowledge about the aptitudes evaluation and the advice about technical aids and adaptations has been collected by an expert group specialised in the disability field and composed of both Spanish and European experts, working together on the ALBOR project founded by the European Union Horizon Program (1997).

All this knowledge is being incorporated into a system capable of performing the intelligent questionnaire tasks identified in above sections. The ALBOR system follows the multi-agent framework. Both agents and objects of the framework appear in the system and preserve the majority of ideas. For instance, any user can enter the system and, therefore, the functionality of the Control Session Agent changes slightly. The user writes a login that will allow the session to be identified later, but does not need to write a password. This information is used by the system to check whether or not this session login exists. If it does, the last session is recovered, and if not, a new session is created. Another peculiarity is the classification of questionnaires into groups (called ‘aptitudes’) that correspond with the four aptitudes evaluated. And, finally, the user interface must be adapted for disabled people and, therefore, the Interface Agent has to detect the user’s disability type in order to adapt appearance and functionality.

Figure 10 shows questionnaire stages used by the Interface Agent. The first line activates the Session Initializer Agent, the second activates the Survey-Taker Agent, the third activates the Advisor Agent, and the last line is used to inform the user. The Decision-Maker Agent starts up automatically when the Survey-Taker Agent leaves data in the Interaction Blackboard Object, but cannot be activated directly from the Interface Agent.

**Figure 10. The Interface Agent shows questionnaire stages.**

For instance, when the user activates the Survey-Taker Agent, the Interface Agent changes its appearance to allow them to interact, as is shown in Figure 11.

**Figure 11. The Questionnaire Interface.**
The Survey-Taker Agent fills in the questionnaire, according to the following stages (Figure 12). First, the Survey-Taker Agent gets the group or ability selected by the user (Visual Ability) from its Internal Knowledge Object (Questionnaire Graph). Second, this agent sends questions (with their corresponding answers) and associated media to the Interface Agent. The Interface Agent asks the user these questions and sends answers to the Survey-Taker Agent. This latter agent writes these answers in the Interaction Blackboard Object.

Finally, this system has been developed by the Centre of Computing and Communications Technology Transfer (CETTICO) and was funded by the UPM (Technical University of Madrid) and IMSERSO (Migration and Social Service Institute).

6. CONCLUSIONS

In this paper, the problem of intelligent questionnaires on the Web is solved using a multi-agent framework to represent actors of the surveying and interaction process.

The framework takes advantage of both multi-agent systems and the Internet computing disciplines. From the multi-agent technology, the agent concept was applied to define the entities of the framework and agent societies were used to connect agents to each other. The Web and the Internet network provided a distributed and parallel platform to build a dynamic and evolving environment. Furthermore, it is important to point out that the use of societies hides and protects objects from the other agents. The societies provide a higher level of abstraction than the agents.

The framework is organised according to the standard multi-agent distribution. Its components (agents, objects, agent societies and interaction processes) are located in a dynamic environment.

Finally, the usefulness of the framework has been proved by applying it in a system to update the agents’ internal knowledge of the ALBOR system. This administration system improves performance by updating agents.

7. ACKNOWLEDGMENTS

This paper was written and prepared with the collaboration of CETTICO (Center of Computing and Communications Technology Transfer), Technical University of Madrid.

8. REFERENCES


