Knowledge Management with an Agent Network

P. Carrión, R. Laza, E. González, J.M. Corchado.

Artificial Intelligence Research Group
Department of Languages and Computing Systems
University of Vigo, Campus As Lagoas 32004, Ourense, Spain
Email {pcarrion, rlaza, nrufino, corchado}@ei.uvigo.es
Tlf.: +34 988387010 Fax.: +34 988387001

ABSTRACT

This paper presents a distributed systems developed using a network of agents. It approaches a Artificial Intelligence solution for implementing distributed systems and presents a system for the exchange of information between agents. This is a current research subject which be experienced for distributed network management. The goal of the system is to help students to select the most suitable optional subjects in a Computing Science Degree. The multi-agent system architecture is based on a number of agents capable of communicating between them. The system includes personal agents associated with students and lecturers, advising agent which use case-based reasoning systems to provide advice to students and a network of communication agents that facilitate the interaction between agents.

Keywords: Agent Communication Language, Case-Based Reasoning, Multi-Agent Systems.

1. INTRODUCTION

One of Distributed Artificial Intelligence's (DAI) most important sub-fields is the Multi-Agent Systems (MAS) paradigm. The MAS concept is largely based upon the idea that complex activities are the outcome of interactions between relatively independent software entities called agents. A MAS may be defined as a set of agents that interact with each other and with the environment to solve a particular problem in a co-ordinated (i.e. behaviourally coherent) manner. This can be achieved either by individual agents seeking localised behavioural optimisation (often referred to as self-interested agents) or by collaborative global system optimisation (e.g. master co-ordinated benevolent agents) [6]. In this paper, an agent is defined from the standpoint of functional characteristics that an agent should possess: autonomy, social ability, reactivity and pro-activeness [8,14].

It is quite apparent that the agent approach was significantly influenced by the communications network area, in the sense that one should not think about centralised global information, but rather in terms of more Open Systems, where geographical distribution of control information, the inter-working of heterogeneous systems, the increase in available processing power and the need to provide application specific services, is generally more pronounced that in closed (singularly owned deterministic) systems.

This paper presents a multi-agent advisory system called SAEAO (*Sistema de Ayuda a la Elección de Asignaturas Optativas*; Advisory System for Selecting Optative Subjects) that facilitates the exchange of information between *autonomous agents*. The system is composed of different types of agents that interact with others and with the environment in order to achieve well-defined goals. The aim of the system is to advise the students of the computing science degree of the University of Vigo about the optional subjects that they should enroll. The distributed system is composed of one advisory agent which uses a Case-Based Reasoning (CBR) system to help the students to select the must suitable subjects for then.

The Spanish educational system allows students enrolled in a university degree to select a number of optional subjects (See Figure 1). The optional subjects must complement the compulsory ones and give the students the opportunity to specialize in a particular area. Although there are students that know what they want to study in order to achieve their academic goals, many of them have difficulty to select the most suitable subjects.

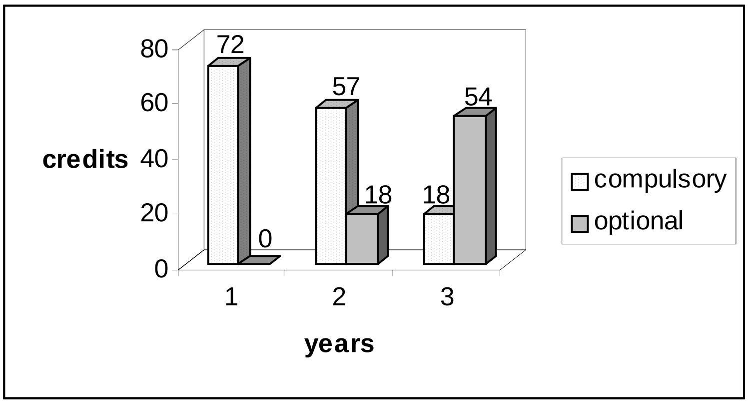

Figure 1: Distribution of compulsory and optional subjects in the "Ingeniería Técnica en Informática de Gestion", Universidad de Vigo. One credit is equivalent to 10 hours of classes [2].

Next section presents the architecture of the multi-agent advisory system SAEAO. After, it reviews the communication protocol used by the agents system. Section four presents a case based reasoning system and its application in the framework of this project. Finally the obtained results are shown.

2. ARCHITECTURE OF THE SAEAO MULTI-AGENT ADVISORY SYSTEM

Figure 2 shows the architecture of the SAEAO system. The agents that form part of this system use facilitators to enable the communication among them. The agents use the Agent Communication Language (ACL) in order to exchange information about their needs and capabilities with their local facilitator. These facilitators are in charge of searching for the most adequate path to send the information to other facilitators, which will pass this information to the agents of their own domain, which are capable of satisfying the request [4]. Each agent includes a router capable of sending and receiving messages using KQML, identifying a set of performatives (words) and satisfying a protocol. These routers are independent processes, so the communication among agents is asynchronous.

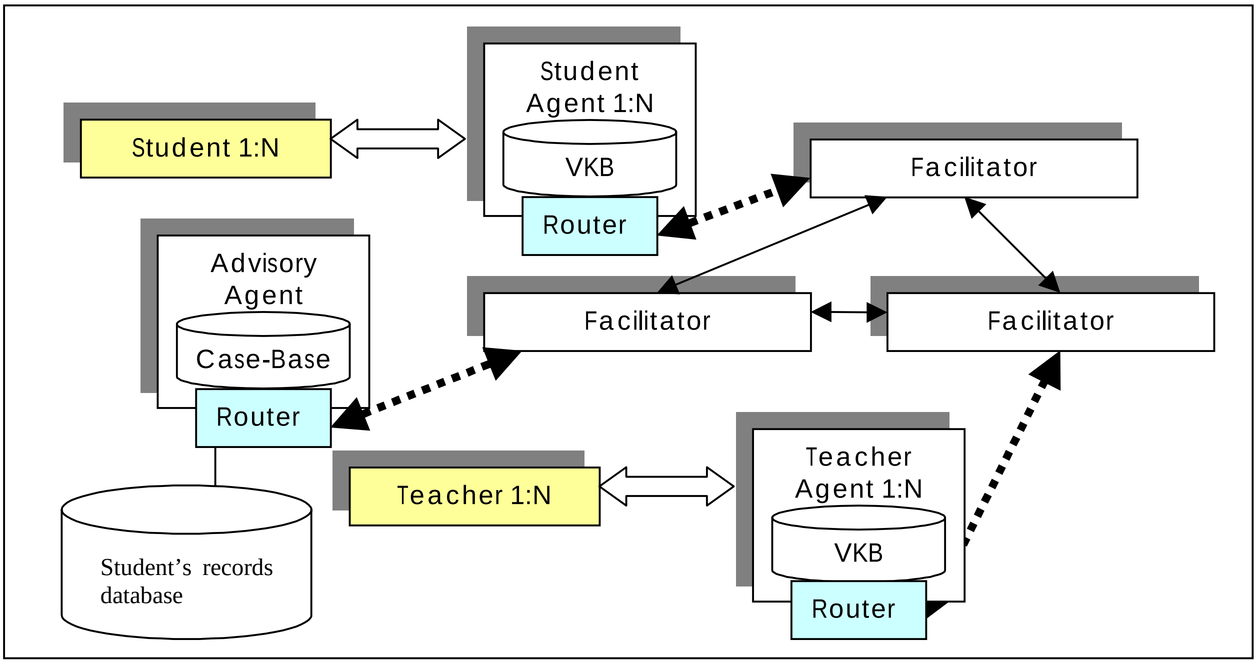

Figure 2: Architecture of SAEAO system.

The agents used in the system are:

Teacher Agents: Each lecturer is associated to a Teacher Agent who knows the ideal profile of the pupil for studying of the optional subject, that the lecturer associated to it carries out. The agent obtains this information by interrogating the lecturer (See Figure 3) and stores the information in it virtual knowledge base (VKB). The lecturer can modify this information at any time, sending the agent new information.

1. What subjects of previous years are they related with this subject?	E.g. *databases, Engineering of the Software, Introduction to the programming, etc.*
2. Would it be advisable to attend some seminar imparted in the University, if it is this way, to which?	E.g. *Advanced Programming, Introduction to Beginner net, etc.*
3. What programming languages are they necessary to know to overcome the subject?	E.g. *Pascal, C/C++, Java, none, etc.*
4. What operative systems are they convenient to know?	E.g. *Unix, MsDos, Windows, none, etc.*
5. What hardware type is it necessary?	E.g. *Pc, Wokstation, Mackintosh, none, etc.*
6. In what areas of interest would you center this subject?	E.g. *Algorithms and Structures of Data, Architecture, Artificial Intelligence, databases, etc.*
7. Would it be convenient to know some foreign language?	E.g. *English, French, German.*
8.What work could it act when studying this subject?	E.g. *Programmer, Analyst of systems, Director of projects.*
9. Schedule of the subject.	

Figure 3: Questionnaire used to characterise the ideal profile of the pupil for studying of the optional subject.

Student Agents: Equally to Teacher Agents there are Student Agents associated to students. Student Agents work in a similar way than Teacher Agents, each Student Agent is devoted to a particular student. The Student Agent learns about the background, areas of interest and academic activities of the student (See Figure 4). This information is stored in the virtual knowledge base of the agent (VKB), together with the advice provided by the Advisory Agent or the Teacher Agent.

1. What subjects have you approved in previous years?	E.g. *databases, Engineering of the Software, Introduction to the programming, etc.*
2. Have you attended some seminar, if it is this way, to which?	E.g. *Advanced Programming, Introduction to Beginner net, etc.*
3. What programming languages are they those that better you know?	E.g. *Pascal, C/C++, Java, etc.*
4. What operative systems are they those that better you know?	E.g. *Unix, MsDos, Windows, etc.*
5. What computer type do you prefer?	E.g. *Pc, Wokstation, Mackintosh, etc*
6. What areas of interest do you like more?	E.g. *Algorithms and Structures of Data, Architecture, Artificial Intelligence, databases, etc.*
7. Do you speak some foreign language?	E.g. *English, French, German.*
8. You would like to work as:	E.g. *Programmer, Analyst of systems, Director of projects.*
9. Do you want to continue your studies with the second cycle?	E.g. *Yes, Not.*

Figure 4: Questionnaire used to identify the profile of a student.

<u>Advisory Agent:</u> The Advisory Agent includes a case based reasoning system, such as the one present in section 4. A CBR system can be used to increase the autonomy and the adaptation capacity of agents and multi-agent systems.

These agents have been implemented using Java and, in consequence, are platform independent.

3. COMMUNICATION OF THE MULTI-AGENT ADVISORY SYSTEM SAEAO

The communication among the agents allows them to synchronize actions, to send and to receive information, to solve conflicts in the resolution of a task, etc [12]. This communication makes possible that the agents perform actions using a global knowledge and that coordinate their actions with the rest of the agents.

One of the desirable characteristics of the agents is their capacity of interaction and of interoperation among them. The intelligent interaction is achieved by sharing the knowledge that each agent has. Shearing the knowledge means that agents must be capable of understanding the shared knowledge and by communicating such knowledge among them.

Several communication paths can be established between agents in of the system:

Student ⟷ Student Agent

The students contact with their Student Agents when they require advice. Then the agents reply with the advice.

Student Agent ⟷ Advisory Agent

The Student Agent sends a message to the Advisory Agent asking for advice and the Advisory Agent replies with the proposed optional subjects (example 1). Then, when the academic year is finished, the Advisory Agent updates its Case Base.

Advisory Agent ⟷ Teacher Agent

The Advisory Agent asks to the Teacher Agent the ideal profile of the pupil for studying of the optional subject and this sends it (example 2). Therefore, when the Advisory Agent is not capable of providing reasonable advise (because there isn't enable information), this informs a predefined Teacher Agent about the impossibility to provide an answer. The Teacher Agent then enquires the lecturer and it reply to the Student Agent with the corresponding advice.

Teacher Agent ⟷ Lecturer
The Teacher Agent informs the lecturer that the Advisory Agent can not provide advise, then the lecturer studies the particular case and provides the advice to his/her corresponding personal agent.

Student Agent ⟵ Teacher Agent
The Teacher Agent sends the advice (provided by the lecturer) to the Student Agent.

The language used, in the SAEAO system, for the communication between agents is the KQML [3, 7, 10]. This language is based in a transport model of point-to-point message passing. KQML is both a message format and a message-handling protocol to support run-time knowledge sharing among agents.

In the following paragraphs, is presented an example of the communication protocol developed to implement this system. This example includes the performatives used to stablish:

- the dialogue between a Student Agent and the Advisory Agent (example 1):

  - ❖ Performative "*ask-if*": used by the Student Agent to ask for advice to the Advisory Agent.
    ask – if
    :*content* (<background>, <areas-interest>, <academic-activities>)
    :*reply-with* qi

  - ❖ Performative "*reply*": used by the Advisory Agent to provide advice to the Student Agent.
    reply
    :*content* (<advised-subjects>)
    :*in-reply-to* qi

- the dialogue between the Advisory Agent and a particular Teacher Agent (example 2):

  - ❖ Performative "*ask-if*": used by the Advisory Agent to ask information about of the ideal profile of the pupil for studying of the optional subject to the Teacher Agent.
    ask – if
    :*content* (<optional-subject>)
    :*reply-with* qj

  - ❖ Performative "*reply*": used by the Teacher Agent to provide information to the Advisory Agent.
    reply
    :*content* (<background>, <areas-interest>, <academic-activities>)
    :*in-reply-to* qj

The expression associated to the parameter *content* has semantically meaning in KIF (Knowledge Interchange Format) [7, 10], such expressions have been constructed using the vocabulary defined for this particular application domain.

4. ADVISORY AGENT

4.1. Introduction to CBR

Case-based reasoning is a problem solving paradigm that in many respects is fundamentally different from other major AI approaches. Instead of relying solely on general knowledge of a problem domain, or making associations along generalised relationships between problem descriptors and conclusions, CBR is able to utilise the *specific* knowledge of previously experienced, concrete problem situations (*cases*). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. A second important difference is that CBR also is an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems [1].

A typical CBR system is composed of four sequential stages that are invoked whenever it is necessary to solve a problem [1,9,13]. Figure 5 shows those four fundamental steps: retrieval, reuse, review and retain (learning). A new problem is solved by *retrieving* one or more previously experienced cases, *reusing* the case in one way or another, *revising* the solution based on reusing a previous case, and *retaining* the new experience by incorporating it into the existing knowledge-base (case base).

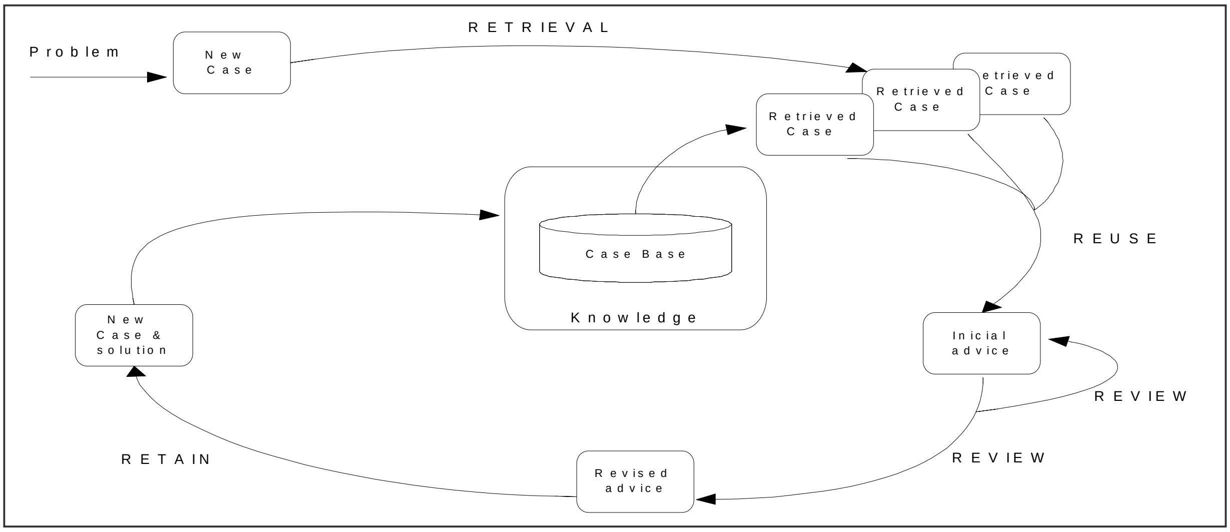

Figure 5: CBR Life cycle.

The Artificial Intelligence Research Group of the University of Vigo holds historical records of students, which show their progress in the subjects they enrol in with different degrees of success. These experiences can be used to help other students to select the optional subjects that may be of most interest for them. A CBR system has been developed as the expert system to advice and to help students in this matter. As mentioned before this CBR system has been implemented in the form of an autonomous agent which constitute the core of the SAEAO system.

4.2. – A CBR to advise university students

In the CBR system each case is represented by two vectors: the *Problem vector* and the *Solution vector*, which are commented in the following subsection.

4.2.1. - Representation of a Case.

The *Problem vector* is composed of a number of Boolean values, which provide an idea of the background, areas of interest and academic activities of the student.

$$\textit{Problem Vector: } P = \{p_1, \ldots, p_n\}$$

The value of the subindex *n* varies depending on the numbers of answers of the questionnaire 1 (See Figure 3), because this can suffer modifications (there are subjects that can disappear, new ones may arise, can be modified, etc).

The *Solution vector*, for a given problem vector, is composed of the optional subjects that have been passed by a student.

$$\textit{Solution Vector: } \mathrm{S} = \{s_1, \ldots, s_n\}$$

The different cases that are stored in the case base, are grouped to facilitate their recovery. The groups are carried out keeping in mind the compulsory subjects that the student has already passed.

When a new case is received, the system determines to which group it belongs, so only the cases belonging to this group are analysed to obtain a solution. The recovery and storage of cases are carried out in a simple way, once it is defined in what group it should be included.

4.2.2. - Life Cycle of the CBR System.

The CBR system developed to provide advise includes the four typical states: retrieval, reuse, review and retain (learning) (See Figure 5).

4.2.2.1. - Retrieval.

After studying several recovery algorithms (Artificial Neural Networks, Euclidean Distance, etc), it was decided to use the function Cosine, based on the model of vectorial space proposed for Jumping and Harman [5]. The dimension of the vectorial space is similar to the total number of terms used in the characterisation of the problem (in this case, the total number of answers of the questionnaire 1). In the model, the basic supposition made is that the relative distance between two vectors in the n-dimensional space represents the difference among the profiles that have been used to configure these vectors. The cosine function is an expression of the angle formed by the two vectors considered:

$$\mathrm{Cos}(P,N)=\sum_{i=1}^{n}(p_i * n_i) \Big/ \sqrt{\sum_{i=1}^{n} p_i^2} * \sqrt{\sum_{i=1}^{n} n_i^2}$$

Where **P** is the problem vector of a case stored in the case base and **N** is the problem vector of the new case.

The result of the cosine function provides values between 0 and 1. If the value of Cos(P,N) is 1, them both cases (**P**,**N**) are identical and therefore the optional subjects that are the solution associated to the vector **P** would be the solution proposed for the case **N**. This similarity function is applied to all the cases that belong to the selected group of the case base of the CBR and those that obtain a value greater than 0.7 are recovered (this value has been obtained empirically).

4.2.2.2 Reuse.

The solutions of the cases recovered in the previous phase are analysed and ordered. Each one of the subjects that form the initial solution should be analysed to check if it can be taken by the student that requests advice, to this end, the student's profile (new case) is compared with the ideal profile to study the subject. If some subject is not adequate, it is eliminated and another subject is selected.

The ideal profile of the pupil to study the optional subject is represented by a vector. The similarity function use for the case reuse is also the cosine.

$$\mathrm{Cos}(P,N)=\sum_{i=1}^{n}(p_i * n_i) \Big/ \sqrt{\sum_{i=1}^{n} p_i^2} * \sqrt{\sum_{i=1}^{n} n_i^2}$$

Where **P** is the vector that characterises the ideal profile of the student to study an optional subject and **N** it is the vector that characterises the student requesting advice.

If when applying the function cosine to the vectors **P** and **N,** it takes a value smaller than 0.7 the option of studying the subject characterised by vector **P** is discarded, and another subject of the solutions recovered in the previous phase is taken.

4.2.2.3. - Review.

The information received in the Reuse phase is also used to identify incompatibilities among subjects. It may happen that some of the optional subject classes take place at the same hour, or that with the proposed solution the student doesn't cover all the necessary optional credits to obtain his University Degree. Therefore, it would be in this phase where optional subjects are eliminated. If they are overlapped in the schedule and the Reuse phase is executed again to select another subject; so it is guaranteed that the student courses the credits he/she needs and that there are not schedule incompatibilities among the advised subjects.

It may also happen that more subjects than the ones the student needs are proposed, in which case the some of them could be eliminated.

4.2.2.4. - Retain.

The CBR stores details of the success of the proposed solutions. Once the student has concluded his/her academic year he/she will have succeeded or failed in the optional subjects. In case he/she has pass them, the case is stored in the case base of the CBR, and will be used to advise other students.

The stored cases are eliminated from the case base when they contain subjects that disappear and they are not replaced by other new ones. When new subjects appear the lecturer requests the system to create prototypical cases, and so the new subjects can also be recommended to students.

The CBR only learns from the success of the proposed solution, due to the nature of the problem. The fact that a student pass a subject depends on his knowledge, but mainly on the effort that he carries out, which is very difficult to measure. For all these reasons, it has been decided that the CBR only keeps in mind those cases whose proposed solution has obtain a certain degree of success. The failure of a student can be for many reasons (familiar problems, state of spirit,…); however, the success of a student is obtained thanks to his daily effort.

5. RESULT AND CONCLUSIONS

The application of agent systems to provide solutions to real problems that are important to real users, is a challenge facing researchers of agent-oriented systems. The objective is to design multi-agent systems (composed of autonomous agents) that bring assistance transparently into normal work environments, while relieving users of low level administrative and clerical tasks [11].

The results obtained with the CBR are promising since it has been proven empirically that this system provides better results. In the experiment carried out with 14 students, these were divided in two groups of seven students who had a similar academic performance. It has been observed that the students that followed the advice of the CBR have failed 12% less than those that didn't receive any advice from the CBR. Although the results obtained come from a very small sample, we believe that they are very optimistic due to the high degree of the students' satisfaction when using this system. The advice proposed by the CBR has been contrasted with advice requested to professors of the department (experts) and they have coincided in a 83%.

One of the part of the SAEAO system is a module (see Figure 6) that is used to create or to modify the student and subject questionnaire, to identify the professor responsible for a subject, etc.

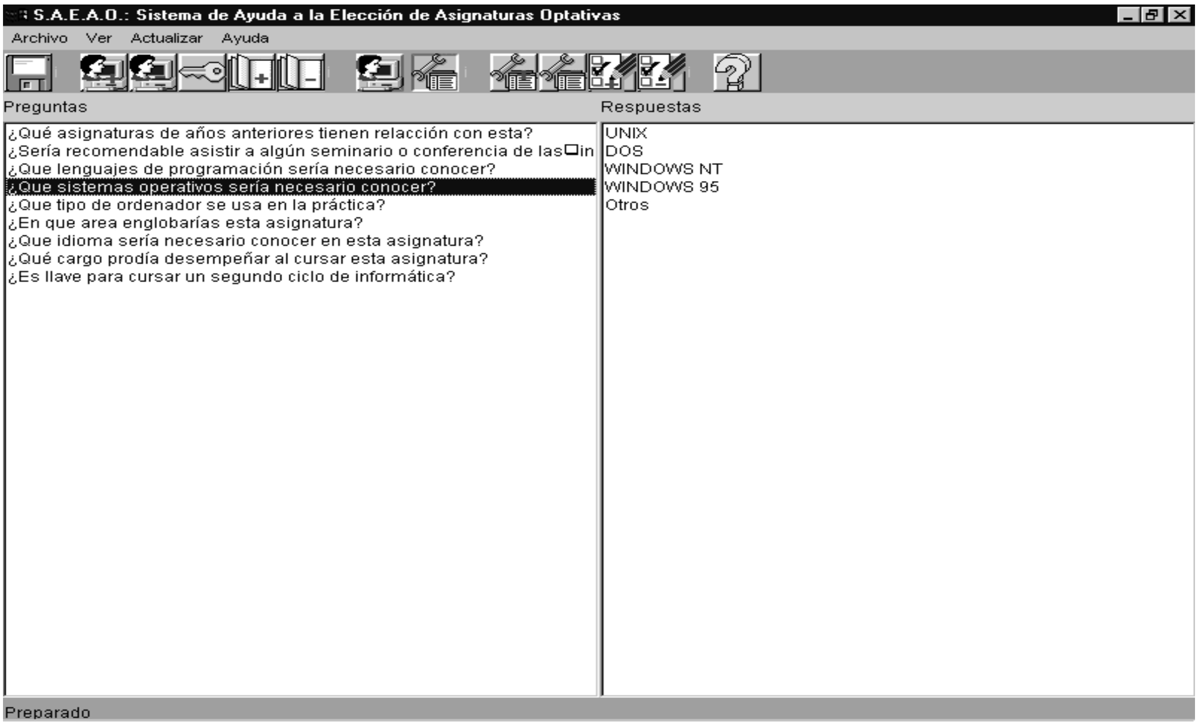

Figure 6: Application inspected by the administrator.

In this article a method for helping in decision making has been presented. This method has been shown to be useful to provide positive results.

Nevertheless, the work carried out in this project has shown that more research is required in order to:

- Study techniques to extract knowledge in the forms of rules from the CBR memory so it is possible to learn more about the context in which the agents inhabit.
- Improve the agents by providing them with better capabilities for taking decisions and negotiating. This potential improvement may be obtained by incrementing the number of services of the facilitators, and in particular, those services that allow the Teacher Agents to collaborate between them. So Teacher Agents could work together to provide stronger advice in case the Advisory Agent could not generate the answer.
- Study the possibility of using the specifications of FIPA 99 (Foundation for Intelligent Physical Agents) as the Agent Communication Language.

6. REFERENCES

[1] A. Aamodt, E. Plaza, "Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches", AICOM. Vol. 7. No 1, March 1994.

[2] BOE, Boletin Nacional del Estado, 18, November, 1997.

[3] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, J. McGuire, S. Shapiro, C. Beck, "Specification of the KQML Agent-Communication Language.", 1993.

[4] M. R. Genesereth, S. P. Ketchpel, "Software agents." Communications of the ACM 37, 7, pp. 48-53. 1994.

[5] D. Harman, "Ranking Algorithms", Ed. W. Frakes, R. Baeza, Information Retrieval: Data Structures and Algorithms, Prentice Hall, London, 1992.

[6] A. Hayzelden, J. Bigham, "Software Agents for Future Communication System", Springer-Verlag, 1999.

[7] KQML, Knowledge, Query and Manipulation Language, http://www.cs.umbc.edu/kqml/

[8] N. R. Jennings, K. Sycara, M. Wooldridge, "A Roadmap of Agent Research and Development", Autonomous Agents and Multi-Agent, Systems Volume 1, Issue 1, 1998. pp. 7-38, 1998.

[9] J. Kolodner, "Case-Based Reasoning. Morgan Kaufmann", 1993.

[10] KSE, The ARPA-sponsored Knowledge Sharing Effort project, http://www.cs.umbc.edu/kse/

[11] P. Maes, editor. "Designing Autonomous Agents", MIT/Elsevier,1993.

[12] S. Russell, P. Nourig, "Artificial Intelligence, a modern approach", Prentice Hall, 1995.

[13] I. Watson, "Applying Case-Based Reasoning: Techniques for Enterprise Systems.", Morgan Kaufmann, 1997.

[14] M. Wooldridge, N. R. Jennings, "Agent Theories, Architectures and Languages: A Survey", Procs. ECAI-94 Workshop on Agent Theories, Architectures, and Languages, 1994.