

Constructing Deliberative Agents with Case-based Reasoning Technology

J. M. Corchado

Departamento de Informática y Automática
University of Salamanca
Plaza de la Merced s/n, 37008,
Salamanca, Spain
Email: corchado@usal.es
<http://gsii.usal.es/>

R. Laza

Departamento de Informática
University of Vigo Campus As
Lagoas, s/n, 32004, Ourense,
Spain

To appear in the **INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS**

Abstract

This paper shows how autonomous agents may be constructed with the help of case-based reasoning systems. The advantages and disadvantages of deliberative agents are discussed, and it is shown how to solve some of their inconvenient, especially those related to their implementation and adaptation. Internet is one of the most popular vehicle for disseminating and sharing information through computer networks and it is influencing in the business world. An agent-based solution is presented to illustrate how the proposed technology may facilitate and improve an e-business strategy.

Key Words: Case-based Reasoning, Deliberative autonomous Agents

1 Introduction

This paper shows how to build deliberative agents, using a case-based reasoning (CBR) system. The proposed methodology facilitates the automation of their construction and provides them with the capacity of learning and therefore of autonomy. The technological evolution during the last few decades has been fast and constant and the commercial computer based systems should be developed with the required mechanisms so they can easily be adapted to the expected changes. It is necessary to build up systems with capacity of adaptation and provided with mechanisms, that allow them to decide what to do according to their objectives. Such systems are known as agents¹. Agents should be autonomous, reactive, pro-active, sociable and have learning capacity. They must be able to answer to events which take place in their environment, to take the initiative according to their goals, to interact with other agents (even human) and to use past experiences to achieve present goals. There are different types of agents and they can be classified in different ways². One of these types are the so-called deliberative agents with a BDI architecture, which have mental attitudes of Beliefs, Desires and Intentions; besides they have capacity to decide what to do and how to get it according to their attitudes^{1,3,4}.

Agents with BDI architecture have their origins in the *practical reasoning* of the traditional philosophy. These agents are supposed to be able to decide in each moment what action to execute according to their objectives. The practical reasoning undergoes two phases: in the first one the goals are defined and in the second one it is defined how to achieve such goals¹. As mentioned before deliberative agents, with a BDI architecture, are composed of beliefs, desires and intentions. The *beliefs* represent their information state, what the agents know about themselves and their environment. The *desires* are their motivation state, what the agents are

trying to attain; and the *intentions* represent the agents' deliberative state. Intentions are sequences (ordered sets) of beliefs (also can be identified as plans). These mental attitudes determine the agent's behaviour and they are critical to attain a proper performance when the information about the problem is scarce^{5,6}. The BDI architecture has the advantage that it is intuitive, it is relatively easy to recognise the process of decision-making and how to perform it; and besides it is easy to understand the notions of belief, desires and intentions. On the other hand, its main drawback lies in determining a mechanism that allows its efficient implementation. The formalisation and implementation of BDI agents constitutes the research of many scientists^{3,4,7,8,9,10}. Some of these researchers criticise the necessity of studying multi-modal logic for the formalisation and construction of such agents, because they haven't been completely axiomatised and they aren't computationally efficient. Rao and Georgeff¹¹ state that the problem is that there is a big distance between the powerful logic for BDI systems and the practical systems. Another problem is that this type of agents doesn't have learning capacity, a necessary attitude for them since they have to be constantly adding, modifying or eliminating beliefs, desires and intentions. Therefore it would be convenient a reasoning mechanism which would involve a final apprenticeship.

This paper shows how a BDI agent implemented using a case-based reasoning (CBR) system can substantially solve the two problems that have been previously mentioned. Implementing agents in the form of CBR systems also facilitate their learning and adaptation. Among the different disciplines of the cognitive science, the cognitive psychology has widely shown the importance of learning from experience¹². If the proper correspondence between the three mental attitudes of the BDI agents and the information that a case-based reasoning system manipulates is established, it will be obtained an agent with beliefs, desires, intention and

besides with learning capacity. Although the relationship between agents and CBR systems have been investigated by other researchers^{13,14,15,16,17,18}, we propose a novel approach, which main characteristic is its direct mapping between the agent conceptualisation and its implementation, in the form of a CBR system.

This paper reviews first the concept of Case-based Reasoning system. Section 3 presents the proposed model, in which a CBR system is used to operate the mental attitudes of a deliberative agent. This section also shows the relationship between the BDI agents and the CBR systems. The e-business application constructed with the help of the agent conceptualisation presented in this paper is then presented. Finally the agent and the e-business system are evaluated and some conclusions are presented.

2 Case-based Reasoning Technology

Case-based reasoning is used to solve new problems by adapting solutions that were used to solve previous similar problems¹⁹. The operation of a CBR system involves the adaptation of old solutions to match new experiences, using past *cases* to explain new situations, using previous experience to formulate new solutions, or reasoning from precedents to interpret a similar situation.

The CBR systems analyse and obtain solutions through algorithms of index, recuperation, comparison techniques and adaptation of problems to a determined situation. To reach this objective, they are based on the knowledge stored in their memory, in the form of cases or problems. Figure 1 shows the reasoning cycle of a typical CBR system that includes four steps that are cyclically carried out and in a sequenced way: retrieve, reuse, revise, and retain^{20, 21}. During the retrieve phase those cases that are more similar to the problem case are recovered

from the case-base. The recovered cases are adapted to generate a possible solution during the reuse stage. Such solution is reviewed and if it is appropriate a new case is created and stored, during the retain stage, in the memory. Therefore CBR systems update (with every retain step) their case-bases and evolve with their environment.

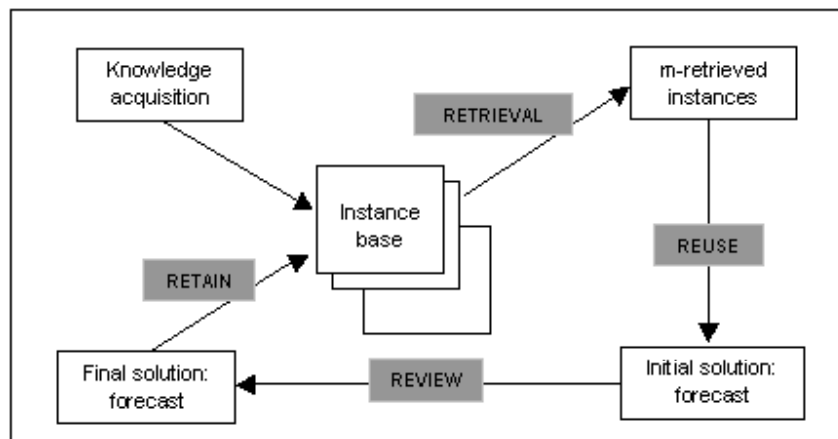


Figure 1: CBR system.

Each of the reasoning steps of a CBR system can be automated, which implies that the whole reasoning process could be automated to a certain extend²². This assumption has carried us to think that agents implemented using CBR systems could be able to reason autonomously and therefore to adapt themselves to environmental changes.

The automation capabilities of CBR systems have lead us to establish a relationship between the cases, the CBR life cycle, and the mental attitudes of the BDI agents. Based on this idea, a model is presented that facilitates the implementation of the BDI agents using the reasoning cycle of a CBR system.

3. CBR System as the Skeleton of a Deliberative Agent

The formalisation presented in this paper takes elements of other^{13,14,15,16,17,18}, and adapts them to the model here presented. It will be presented the relationships that exist between a deliberative agents, with a BDI architecture and a CBR. A direct mapping between the agents and the case-based reasoning system has been identified so the mapping allows a direct and straightforward implementation of the agent, and the agent is capable of learning and adapting to environmental changes. The developed agent, than, may reason with the help of a case-based reasoning system.

To achieve both goals, the structure of the CBR system has been designed around the concept of a case. A problem, a solution and the result that was obtained when the proposed solution was applied usually make a case. Figure 2 shows these components: the problem defines the situation of the environment in a given moment, the solution is the set of states the environment undergoes as a consequence of the actions that carried out inside it, and the result shows the situation of the environment once the problem has been solved.

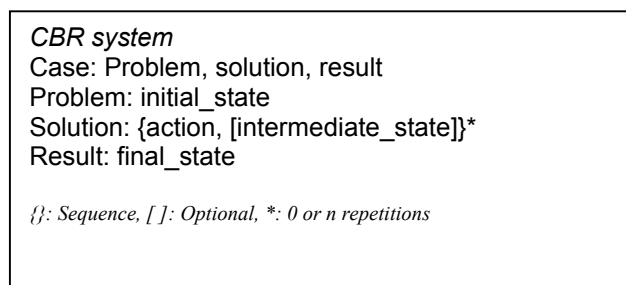


Figure 2: Definition of a case in a case-based reasoning system.

Figure 3 defines what are the beliefs, desires and intentions for a BDI agent. Each change from state to state, after carrying out an action, is considered a *belief* (the agent remembers the

action it carried out in the past when it was in a determined situation and the result it obtained). A belief may also be the objective to reach. The *intentions* are plans of action that the agent is obliged to carry out in order to achieve its objectives²³, so an *intention* is a set (with an order) of *beliefs*. A *desire* will be any of the final states reached in the past (if the agent has to deal with a situation, which is similar to another in the past, it will try to make the result similar to that obtained before).

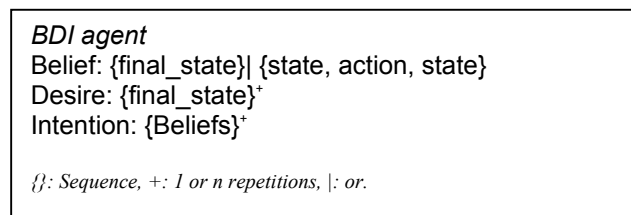


Figure 3: Definition of the mental attitudes of a BDI agent.

The relationship between CBR systems and BDI agents can be established implementing the intention as cases (ordered sequences of actions and states). A case is therefore equal to an intention. The straight relationship between BDI agents and CBR systems can be identified looking at figure 2 and 3.

Using this relationship we can implement agents (conceptual level) using CBR systems (implementation level). Then we are mapping agents into CBR systems. The advantage of this approach is that a problem can be easily conceptualised in terms of agents and then easily implemented in the form of a CBR system. So once the beliefs, desires and intentions of an agent are identified, they can be mapped into a CBR system. In order to show how to implement such agents, a formal notation is introduced to describe the CBR systems and their reasoning process. This notation can be also used to define the beliefs, desires and intention of the agents, due to their correspondence with the elements that make up a CBR system.

The components of a CBR system in this context, are the following:

- Set of case-bases (β): a case-base $B \in \beta$, is a finite set of cases, which are indexed. So, a case-base can be defined as a tuple $(\{c_1, c_2, \dots, c_n\}, \tau)$. Where $\{c_1, c_2, \dots, c_n\}$ are the cases, and τ is the finite set of attributes by which the cases are indexed.
- Case (C): it represents a past experience. A case is represented by an ordered sequence of states. $(c = \{ \text{ini_state}, \{\text{action}, [\text{inter_state}] \}^*, \text{final_state} \}$ or $c = \{\text{final_state}\}$). Each state can be represented by a set of attributes that describe the environment where the CBR system is acting. The states are divided in three different groups:
 1. Set of initial states (ini_state), represents the problems that have to be solved and that constitute the entrance to the CBR system.
 2. Set of intermediate states (inter_state), represents the different states that the environment undergoes before obtaining a desired final state.
 3. Set of final states (final_state), represents the results obtained after carrying out a series of actions starting from a concrete state.

Beside, a case includes:

4. Set of actions (actions), represents the actions applied to a concrete state. It is defined by a noun of action and a set of arguments.
- Finite set of attributes (κ): a state is described by means of a set of attributes.
 - Set of index (I): an index is a set of characteristics τ . $\tau \subset \kappa$.
 - Set of present desires (D): depending on the problem to solve, the final states of some cases would be part of the desires to reach. $d \in D$, $D \equiv \{\text{final_state}\}^+$.
 - A metric (A): a function of similitude determines the degree of equality between two states.

When a CBR system receives a new problem $state_n$, it will be obtained a $final_state$ and the intermediate states $inter_state$, before reaching such $final_state$.

3.1 The Case-based Reasoning Cycle

The four phases of the life cycle of a CBR system are now outlined.

Retrieve:

The cases similar to the new problem $state_n$, (c_1, c_2, \dots, c_k) are retrieved from the case-base B using a similarity metric A_1 . The retrieved cases are c_1, c_2, \dots, c_k .

Reuse:

A first solution $(state_n, \{action_{ni}, \{inter_state_{ni}\}^+, final_state_n)$ is obtained from the retrieved cases and the problem case $(state_n)$ using the metric A_2 . This initial solution is a plan, an ordered sequence of states and actions.

Revise:

Here, it is evaluated the plan $(state_n, \{action_{ni}, \{inter_state_{ni}\}^+, final_state_n)$ obtained in the previous phase. It is checked if $final_state_n$ and the plan developed to obtain it is adequate. The revision can be carried out using simulation techniques²⁴, Belief-revision²⁵, etc. If the revision process concludes that the proposed solution is not acceptable, the plan is sent back to the Reuse stage, as indicated in Figure 1, where a new solution will be proposed.

Retain:

The new plan $(state_n, \{action_{ni}, \{inter_state_{ni}\}^+, final_state_n)$ is indexed and stored in the correspondent case-base B .

As it was stated in the introduction of the paper the aim of this investigation is to develop a methodology for constructing deliberative agents capable of learning and adapting to new situations. To set up an agent using this architecture we need to identify an initial set of beliefs, desires and intentions and include them in the case-base of the agent in the form of cases. Then a number of metrics has to be defined for the retrieval, reuse, revise and retain steps. Once the agent has been initialised it starts the reasoning process and the four steps of the CBR system are run sequentially and continuously until its goal is achieved. During this process its memory changes and new beliefs, desires and intentions could appear.

4 Agent-based System for Sales Support

The construction industry is an information intensive economic sector. This activity, as many others, require the use of a great amount of data, ranging from product data to technical publications, from buildings regulations to best practice guides. This section describes an information system that has been developed for a construction company, D&B Constructions, in which a CBR-BDI agent has been used. From now on, we will refer to the BDI agents implemented as mentioned in the previous section as CBR-BDI agents. This distributed agent based system helps the company to take as much profit as possible from the information published in the Internet and the information that the company holds, and to reuse it as much as possible especially to estimate budgets.

The e-business engineering sales support system incorporates several specialised agents that search for and organise information and data, and several assistant sales support agents. The first prototype of this multiagent system has been implemented in Java using servlets (running in an Apache, Tomcat, jserv, Linux environment). A commercial system is under construction

presently, after an initial successful testing period. The specialised agents are Java applications that run in the company Intranet and the assistant agents run in a portable computer connected to Internet via a mobile phone.

The D&B Constructions deals with medium to small construction problems and it is specialised in installing heating and air conditioning systems in a wide area of the Northwest of Spain. They have a sale force that is growing continuously, which implies that continuously new salesmen are taken on board without much experience in many cases. Until now the salesmen had to visit the clients on demand, had to take notes of their problems and then they had to contact an engineer or an experienced salesman, which had to estimate the work price and personnel and material required to carry on the work. The system here outlined was developed to facilitate the sales force, and in particular to the inexperienced personnel, the estimation of costs, reducing the process bureaucracy.

In the expansion policy of B&D Constructions one of the main points is its incorporation to the new technologies. Several steps will be taken in this direction for developing a web based information system that allows the company to publish information about their activities and that facilitates the communication between the administration, the sales force, the providers and the clients. Figure 4 presents the architecture of the multiagent system. The planning agent is a CBR-BDI agent, implemented with the architecture described in section 3. The planning agent is the only CBR-BDI agent used in this architecture, it estimates the construction cost, and the personnel and material required to carry out a construction project. It also generates reports about clients (or potential clients) using the information stored in the company databases and the one obtained by the internet search agent from the web. The planning agent generates working plans using their incorporated CBR system. The internet agent incorporates a web

search engine that looks continuously for potential clients, information about them, new providers and products. This agent starts looking from a predetermined web address and search for new ones using natural language processing strategies, optimised for the Web, based on the combination of the grammatical characterisation of words and statistical decision techniques²⁶. This agent is monitored and guided by a marketing expert. We are also studying the possibility of implementing this agent using the CBR-BDI model here presented or as proposed in [18]. Assistant agents (they can be as many as salesmen) are interface agents that facilitate the communication between the salesmen and the planning agent, they also hold summarised information about the clients visited by its salesman owner and by the rest of the salesmen.

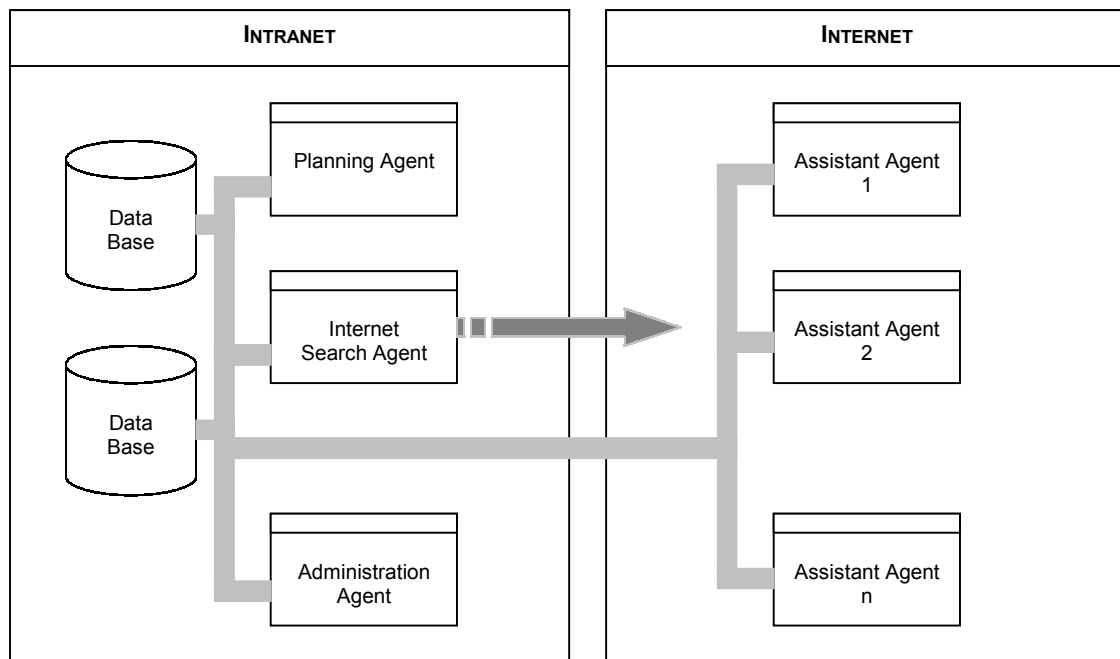


Figure 4: Agent-oriented e-business architecture.

Before a salesman visits a client, he/she interrogates his/her assistant agent providing a description of the client (Name, Address and Activity). The assistant agent compares this data with previous queries and if a match is found, using relaxed K-nearest neighbour algorithms, the data it holds about the client is presented to the salesman²⁷. This information is related to previous building work carried for the client, his financial status, comments about him, noted by the Firm personnel during previous relations with such client, location information and other possible sensible data. This information is valuable especially when an inexperienced salesman starts a negotiation process. If the assistant agent cannot help the salesman or if the salesman demand more information, his assistant agent contacts the planning agent, which search for information about the client in its case-base. This agent also interrogates the internet search agent asking for information about clients. The internet search agent obtains information from the web, analyses and indexes it using natural language processing algorithm optimised for Internet, as mentioned above. Information about potential clients, new materials and providers is sent to the administration agent, which can be interrogated by any of the Construction Company managers, engineers or sales supervisors. They can, then, use this pruned information to target new business. The administration agent is an interface agent that facilitates the interaction between the users (Company managers) and the rest of the elements of the system: agents, databases and even salesmen.

As mentioned before the multiagent²⁸ system has been built using a Java-based library. This library is an extension of the one used to implement the STEB (Simulated Tactical Environmental Bubble) system²⁴. The STEB system was designed to forecast the temperature of ocean waters ahead of ongoing vessels. It is a multiagent system composed of several software agents that co-operate between them and that are operational in different locations:

war/oceanographic vessels and in an Oceanographic Laboratory (Plymouth Marine Laboratory). The agents installed on the vessels use hybrid CBR-ANN system²² to forecast and communicate with the rest of the agents using KQML performatives.

Similarly than for the STEB system, when constructing the e-business multiagent system here presented, a decentralised architecture was selected, in which agents interact between each other when require information or need to share data. The agents communicate with each other using a message passing protocol. Such messages are KQML performatives. The agents of this system collaborate between each other sharing information and working together to achieve a given goal. They use a simple collaboration mechanism. For example, if the Salesman A is associated to the Assistant Agent A, and visits a Client X, the Assistant Agent A has to contact with (send the problem of the Client X, via a performative) the Planning Agent. Then the Planning Agent generates the solution plan, as will be shown in section 4.2, and sends it back to the Assistant Agent.

In a system of these characteristics the data security has to be taken into consideration. A Role-based Access Control with elements that allow the certification of operations has been implemented to guarantee the data security and the information protection²⁴. This security system protects the databases and the information stored in the system from external “agents” or none accredited personnel.

4.1 The Planning Agent

The working mode of the planning agent will be explained to show how the deliberative agents can reason, acquire new knowledge and help the distributed information system to evolve. The other agents of the system do not have a CBR-BDI architecture because they are responsible of

carrying out tasks that do not require reasoning. They are interface agents (assistant and administration ones) or have mechanical tasks to perform such as the internet search engine. Nevertheless, as it has been mentioned before we are studying the possibility of implementing the internet search agent in the form of a CBR-BDI agent, to gain more autonomy and efficiency. Two are the tasks carried out by the planning agent: estimation of the construction cost, the personnel and material required and generation of reports about clients. We will focus in the first task. The second task is carried out following an automatic sequence of queries to the agent case-base and to the system databases. To facilitate the understanding of the problem we have simplified the problem reducing the number of attributes used to describe a building or a house (the implemented system uses 45 attributes) and the number of states that define an intention.

Intentions correspond to working plans, which can be generically seen as ordered set of states and actions:

INTENTION I: INITIAL client state → Action 1 (Material-price; workers required/ ratio hours/men) → client state A → Action 2 (Material-price; workers required/ ratio hours/men) → client state B → Action 3 (Material-price; workers required/ ratio hours/men) → client state C → → Action X (Material-price; workers required/ ratio hours/men) → FINAL client state (Price; Satisfaction level of client; Satisfaction level of Construction Company).

For example:

Intention I1: STATE₁ (BT=h2; Size=225; Heating=no; IM= 17; CY=1973)→

ACTION₁ (Installation of heating pipes: 200 Eur, 3t/2d, 85 m pp+elements, installation kit27)→

STATE₂ (BT=h2; Size=225; Heating=no; IM= 17; CY=1973; HP=H2300-45m)→

ACTION₂ (Installation of radiators: 266 Eur, 2t/1d, 16 radiators +elements, installation kit13)→

STATE₃ (BT=h2; Size=225; Heating=no; IM= 17; CY=1973; HP=H2300-45m, HR=RS16-16-22)→

ACTION₃ (Installation of diesel heating system : 350 Eur, 2t/0.5d, diesel heating+elements, installation kit12)→

STATE₄ (BT=h2; Size=225; Heating=no; IM= 17; CY=1973; HP=H2300-45m, HR=RS16-16-22, HS=B26-600)=

STATE_{FINAL} (BT=h2; Size=225; Heating=D2; IM= 17; CY=1973)

where *Building Type* = *BT*, *Insulating Material* = *IM*, *Construction Year* = *CY*, *Heating Pipe* = *HP*, *Heating Radiators* = *HR*,
Heating System = *HS*

$STATE_1$ indicates that the building is of type *h2*, which means that is a country house with 2 floors, the size is given in square metres. Also indicates that the house does not have any heating system, that the insulating material is of type *I7*, which is a very precarious insulation system and finally it indicates that the house was build in 1973. $STATE_4$ is the final state, which for simplification can be described as in $STATE_{FINAL}$. $ACTION_1$ indicates that the cost of the installation of the heating pipes required is 200 Eur, 3 technicians were required during 2 days, 85 metres of pipes were used together with the elements needed to install them and the installation tool kit used was number 27.

Beliefs included in this intention are:

Believe B1: STATE₁ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973)→

ACTION₁ (Installation of heating pipes: 200 Eur, 3t/3d, 85 m pp+elements, installation kit27)→

STATE₂ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m)

Believe B2: STATE₂ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m)→

ACTION₂ (Installation of radiators: 266 Eur, 2t/1d, 16 radiators +elements, installation kit13)→

STATE₃ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m, HR=RS16-16-22)

Believe B3: STATE₃ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m, HR=RS16-16-22)→

ACTION₃ (Installation of diesel heating system : 350 Eur, 2t/0.5d, diesel heating+elements, installation kit12)→

STATE_{FINAL} (BT=h2; Size=225; Heating=D2; IM= I7; CY=1973)

The desire of the agent with respect to a particular Client is to determine a plan, which $STATE_{FINAL}$ is the one desired by the Client, satisfying his time, price and quality restrictions. For example, if a Client, which property is defined by the $STATE_n$ (*BT=h1; Size=125; Heating=no; IM=*

I2; CY=1991) requires an electric heating system, the goal of the planning agent will be to determine a plan that transform $STATE_n$ into the Client desired state: $STATE_{FINAL}=(BT=h1; Size=125; Heating=E1; IM= I2; CY=1991)$.

Then, once the intentions and beliefs of the agents are identified, the agent will be able to generate plans to achieve its desires, providing that the metrics of each of the stages of the CBR system, that defines the agent reasoning mode, had been defined. The goals of the agent change with each new Client. A brief description of the metrics used by this agent during the reasoning is now presented.

Retrieval:

The retrieval stage is been carried our using a Kernel Method²⁹ that guaranties the retrieval of a number of cases that are related to the problem case. This method facilitates the detection and retrieval of an adequate number of cases. Those intentions/cases, in which there is a state similar to the state at which the client is, $STATE_n$, or the state that the client wants to achieve, $STATE_{FINAL}$, are retrieved.

Reuse:

An initial solution can be obtained by using the sequence of actions carried out in the past, or modifying the sequence of actions, adapting it to the new problem. If, for example, the cases recuperated in the previous phase are the following (a: action):

$$\begin{array}{cccccc}
 a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & \\
 STATE_{k2} \rightarrow STATE_{k21} \rightarrow STATE_{k22} \rightarrow STATE_{k23} \rightarrow STATE_{k24} \rightarrow STATE_{r2} & & & & & \\
 a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & \\
 STATE_{k3} \rightarrow STATE_{k31} \rightarrow STATE_{k32} \rightarrow STATE_{k33} \rightarrow STATE_{k34} \rightarrow STATE_{r3} & & & & & (1)
 \end{array}$$

There can be two possibilities. First, that the environment was, in the past, in a state that is almost identical to the new problem, $STATE_n$, and was obtained the same solution as the one requires now, so it can be carried out the same sequence of actions than in the past. Second, that the recuperated cases are similar to the present state, $STATE_n$ but with some differences. In the second case a sequence of actions, that is a mixture of those that were recuperated in the previous phase, is constructed. To do so, an acyclic directed graph is created, whose first vertex is the new problem/state, and the last vertexes are the final states. The construction of the graph is carried out starting from the new state and applying to it a function of similitude with all the recuperated states; those that are more similar will determine what action will be carried out starting from the new state. This process is repeated with all the cases until a final state is reached.

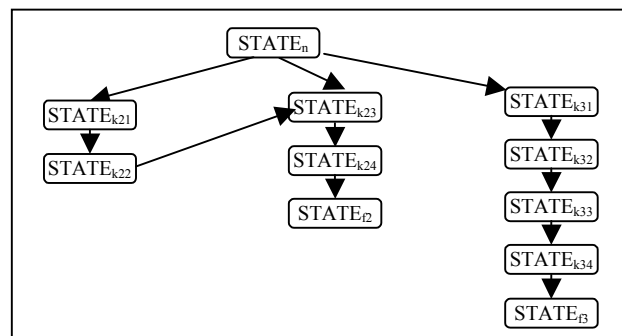


Figure 5: Acyclic graph starting from the recuperated cases in the recuperation phase.

Figure 5 clarifies the exposed process. Starting from the recuperated cases in (1), and after applying a function of similitude it is obtained that the closest states to $STATE_n$ are $STATE_{k2}$,

STATE_{k22} and STATE_{k3}. So from STATE_n the actions a_{21} , a_{23} , a_{31} will be respectively carried out. The results obtained, by means of simplicity, in this case, are supposed to be identical to those recuperated in the previous phase, if they were not the same, the function of similitude would be applied again to the obtained states.

Once the graph has been constructed, the algorithm of Dijkstra is used³⁰ to determine the shortest way (the way that goes through less vertex) taking the new state as the origin. Such path will define the actions that must be carried out from the new state, and so, they will make up the new intention. Then in this case, the shortest way is made up by STATE_n, STATE_{k23}, STATE_{k24}, STATE_{r2}.

Revise:

The revision process is carried out using Belief Revision techniques³¹. A rule-based system is used during this phase, which is updated automatically using a Belief Revision technique that uses Epistemic Entrenchment, as constructive model.

Retain:

After the work has been carried out, the plans are stored in the form of cases. Once a new case is created, it is stored in temporal case-base. A senior salesman accesses this case-base via the administration agent and decides which of these cases/instances should be stored by the CBR-BDI planning agent. Techniques to automate this process are under investigation.

5 Conclusions

For evaluation purposes, during the testing period, the case-base of the planning agent has been fed with 280 cases related to the installation of heating systems. These cases were selected to

cover a wide spectrum of possible installations that the company could carry out (D&B Constructors hold a data base with over 5600 installations from January 1997). The system was interrogated in 30 occasions.

In 23 occasions the estimation differed in less than 4% of the one given by an expert, and in 7 occasions differed in less than 9%. These deviations were caused by one reason, the client required a combination of installations and equipment, which did not appear in any of the 280 cases stored in the agent case-base. These errors could be minimised during the review phase if the agent check out the degree of similarity between the case problem and the retrieved cases. Which respect to this point, strategies are under investigation to identify problems and/or “generated plans” with potential risks. It is expected that the accuracy of the business solution will increase as more cases are introduced in the planning agent memory. Company experts have estimated that the use of this agent-based system could reduce the installation sales cost up to 40% of the actual cost, and the time of the sale up to 50%.

The architecture presented in this paper solves one of the problems of the BDI (deliberative) architectures, which is the lacking of learning capacity. The reasoning cycle of the CBR systems helps the agents to solve problems, facilitate its adaptation to changes in the environment and to identify new possible solutions. Morá *et al.*³² have described the gap that exists between the formalisation and the implementation of BDI agents. What we propose in this article is to define the beliefs and intentions clearly (they don't need to be symbolic or completely logic), and to use them in the life cycle of the CBR system, to obtain a direct implementation of a BDI agent. This paper has shown how single agents can be developed with this technology, and how such agents can be successfully used to construct an efficient agent based system for e-business.

Acknowledgement

This work has been partially supported by the CICYT project SEC2000-0249, the PGIDT00 project MAR30104PR and the project SA039/02 of the JCyL.

Bibliography

1. Wooldridge M. (1999) Intelligent Agents. Multiagent Systems. A modern approach to Distributed Artificial Inteligence. Edited by Gerhard Weiss, pp 27-77.
2. Wooldridge M. and Jennings N. R. (1994) Agent Theories, Architectures, and Languages: A Survey. Procs. ECAI-94 Workshop on Agent Theories, Architectures, and Languages.
3. Jennings N.R. (1992) *On Being Responsible*. In Y. Demazeau and E. Werner, editors, Decentralized A.I. 3. North Holland, Amsterdam, The Netherlands.
4. Rao A. S. and Georgeff M. P. (1991) Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann Publishers, San Mateo, CA.
5. Bratman M.E. (1987) *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, M.A.
6. Kinny D. and Georgeff M. (1991) *Commitment and effectiveness of situated agents*. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI'91), pp 82-88, Sydney, Australia.

7. Cohen P.R. and Levesque H.J. (1990) *Intention is choice with commitment*. Artificial Intelligence, Vol 42, No. 3.
8. Kinny D., Ljungberg M., Rao A.S., Sonenberg E.A., Tidhar G, and Werner E. (1994) *Planned team activity*. In Artificial Social Systems. Lecture Notes in Artificial Intelligence (LNAI-830). Amsterdam, Netherlands. Springer Verlag.
9. Georgeff M.P. and Lansky A.L. (1986) *Procedural knowledge*. In Proceedings of the IEEE Special Issue on Knowledge Representation, Vol. 74, pp 1383-1398.
10. Shoham Y. (1993) Agent-Oriented programming. Artificial Intelligence, Vol 60, No. 1: pp. 51-92.
11. Rao A. S. and Georgeff M. P. (1995) BDI Agents: From Theory to Practice. First International Conference on Multi-Agent Systems (ICMAS-95). San Francisco, USA, June.
12. Caplan L.J. and Schooler C. (1990) *Problem Solving by Reference to Rules or Previous Episodes: The Effects of Organized Training, Analogical Models, and Subsequent Complexity of Experience*. Memory & Cognition, Vol. 18, No. 2, pp. 215-227.
13. Feret M. P. and Glasgow J. I. (1994) Explanation-Aided Diagnosis for Complex Devices, Proceedings of the 12th National Conference on Artificial Intelligence, (AAAI-94), Seattle, USA, August 94.
14. Napoli A, Lieber J and Curien R. (1996) A formal Analysis of Case-based reasoning in an object-based representation Context. In Faltings B and Smith I, editors. 3rd European Workshop and Case-based Reasoning, Lausanne, Lecture Notes in Artificial Intelligence, 1168, pp. 295-308. Springer-Verlag, Berlin, 1996.

15. Glez-Bedia M. and Corchado J. M. (2003) Constructing autonomous distributed systems using CBR-BDI agents. *Innovation in knowledge Engineering*. Faucher C., Jain L. and Ichalkaranje N. (eds.). Advanced Knowledge International. ISBN: 0-9751004-0-8.
16. Bergmann R and Wilke W. (1998) Towards a New Formal Model of Transformational Adaptation in Case-Based Reasoning. *European Conference on Artificial Intelligence (ECAI'98)*, John Wiley and Sons.
17. Wendler J. and Lenz M. (1998) CBR for Dynamic Situation Assessment in an Agent-Oriented Setting. *Proc. AAI-98 Workshop on CBR Integrations*. Madison (USA) 1998.
18. Olivia C., Chang C. F., Enguix C.F. and Ghose A.K. (1999) Case-Based BDI Agents: An Effective Approach for Intelligent Search on the World Wide Web", *AAAI Spring Symposium on Intelligent Agents*, 22-24 March 1999, Stanford University, USA
19. Riesbeck, C. K. and Schank, D. B. 1989. *Inside Case Based Reasoning*. Erlbaum.
20. Glez-Bedia M, Corchado J. M., Corchado E. S. and Fyfe C. (2002) Analytical Model for Constructing Deliberative Agents, *Engineering Intelligent Systems*, Vol 3, pp. 173-185.
21. Watson I. and Marir F. (1994) *Case-Based Reasoning: A Review*. Cambridge University Press, 1994. *The knowledge Engineering Review*. Vol. 9. No. 3.
22. Corchado J. M. and Lees B. (2001) A Hybrid Case-based Model for Forecasting. *Applied Artificial Intelligence*. Vol 15, No. 2, pp. 105-127.
23. Bratman M.E., Israel D., and Pollack M.E. (1988) *Plans and resource-bounded practical reasoning*. *Computational Intelligence*, 4. pp. 349-355.
24. Corchado J. M. and Aiken J. (2002) Hybrid Artificial Intelligence Methods in Oceanographic Forecasting Models. *IEEE SMC Transactions Part C*. Vol 32, No.4.

25. Gärdenfors P. and Rott H. (1995) *Belief Revision*. In Handbook of Logic in Artificial Intelligence and Logic Programming, Vol 4, Chapter 4.2.
26. Corchado J. M., Laza R., Borrajo L., Yañez J. C. and Valiño M. (2003) Increasing the Autonomy of Deliberative Agents with a Case-Based Reasoning System. International Journal of Computational Intelligence and Applications. *Vol. 3, No. 1*, pp.101-119.
27. Watson I and Gadingen D.(1999) A Distributed Case-Based Reasoning Application for Engineering Sales Support. In Proceedings of IJCAI-99. Morgan Kaufmann Publishers Inc. Vol 1, pp 600-605.
28. Weiß G. (1996). Adaptation and learning in multi-agent systems: some remarks and Bibliography. Proceedings IJCAI'95 workshop. Montreal, Canadá. August, 1995.
29. Fyfe C. and Corchado J. M. (2001) Automating the construction of CBR Systems using Kernel Methods. International Journal of Intelligent Systems. Vol 16, No. 4.
30. Schulz F., Wagner D. and Weihe K. (1999) Dijkstra's Algorithm On-line: An Empirical Case Study from Public Railroad Transport. Algorithm Engineering . pp. 110-123.
31. Pavón R., Laza R., Gómez-Rguez. A. and Corchado J. M. (2001) Improving the Revision Stage of a CBR System with Belief Revision Techniques. Computing and Information Systems Journal, Vol 8 No 1.
32. Móra M. C., Lopes J. G., Viccari R. M. and Coelho H., (1998) BDI Models and Systems: Reducing the Gap. ATAL-98.