Construction of BDI Agents from CBR systems[1]

J. M. Corchado* and R. Laza*

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

**Departamento de Informática
University of Vigo
Campus As Lagoas, s/n, 32004, Ourense, Spain
Email: rlaza@uvigo.es

Abstract. This paper shows how deliberative agents can be built by means of a case-based reasoning system. The concept of deliberative agent is introduced and the case-based reasoning model is presented. Once the advantages and disadvantages of such agents have been discussed, it will be shown how to solve some of their inconvenients, especially those related to their implementation and adaptation. The World Wide Web has emerged as one of the most popular vehicle for disseminating and sharing information through computer networks, a distributed agent-based solution for e-business, in which such agents have been used, is also presented and evaluated in this paper.

1 Introduction

In most of the computing systems, all the executed actions are previously planned and encoded by a programmer. However, in our present-day world, where technological evolution is fast and constant, 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 agents [Wo99]. This paper shows how to build deliberative agents, using a case-based reasoning (CBR) system. The proposed method facilitates the automation of their construction and provides them with the capacity of learning and therefore of autonomy.

Agents 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

1 This work has been partially supported by the CICYT projects TEL99-0335-C04-03 and SEC2000-0249. The contribution of Mr. De Andres at D&B Constructions in the collaborative research presented in this paper is gratefully acknowledged.

agents and they can be classified in different ways [WJ94]. 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 [Wo99] [Je92] [RG91].

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 achive; and the *intentions* represent the agents' deliberative state. Intentions are sequences of actions, they can be identified as plans. These mental attitudes determine the agent's behaviour and are critical to attain a proper performance when the information about the problem is scarce [Br87] [KG91]. BDI architecture has the advantage that it is intuitive, it is relatively simple to identify the process of decision-making and how to perform it. Besides, the notions of belief, desires and intentions are easy to understand. On the other hand, its main drawback lies in finding a mechanism that allows its efficient implementation. The formalisation and implementation of BDI agents constitutes the research of many scientists [CL90] [Je92] [Ki94] [RG91] [GL86] [Sh93]. Some of them criticise the necessity of studying multi-modal logic for the formalisation and construction of such agents, because they have not been completely axiomatised and they are not computationally efficient. Rao and Georgeff [RG95] state that the problem lies in the big distance between the powerful logic for BDI systems and practical systems. Another problem is that this type of agents is not able to learn, 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 system can substantially solve the two problems 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 experience [CS90]. If the proper correspondence between the three mental attitudes of BDI agents and the information manipulated by a case-based reasoning system is established, an agent with beliefs, desires, intention and besides with learning capacity will be obtained. Although the relationship between agents and CBR systems have been investigated by other researchers [FG94] [Na96] [MPA99] [BW98] [WL98] [Ol99], 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 introduced in this paper is then presented. Finally the agent and the e-business system are evaluated and some conclusions are exposed.

2 Case-based Reasoning Systems

Case-based reasoning (CBR) is used to solve new problems by adapting solutions that were used to solve previous similar problems [RS89]. To reach this objective, CBR systems 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 in a sequenced way: retrieve, reuse, revise, and retain [AP94] [WM94]. During this process the memory can change and new cases may appear. A sub-phase within the retain phase is included for this reason, in which *Expert's knowledge* is revised.

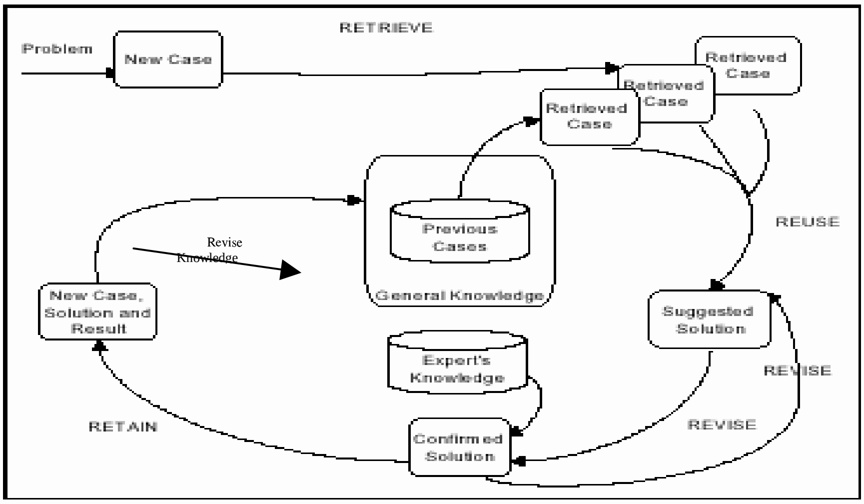

Fig. 1. CBR Cycle of Life.

Each of the reasoning steps of a CBR system can be automated, this implies that the whole reasoning process could be automated to a certain extend [CL01]. This assumption means in out oppinion, 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 led us to establish a relationship among cases, CBR life cycle, and mental attitudes of BDI agents. Based on this idea, a model that facilitates the implementation of the BDI agents using the reasoning cycle of a CBR system is presented.

3 Implementing Deliberative Agents using CBR Systems

This section identifies the relationships established between BDI agents and CBR systems, and shows how an agent can reason with the help of a case-based reasoning system. Our proposal pretends to define a direct mapping from agents to the reasoning model paying special attention to two elements: (i) the mapping between the agents and the reasoning model should allow a direct and straightforward implementation of the agent and (ii) the final agents should be capable of learning and adapting to environmental changes. In the presented model the CBR system is completly integrated in the agents architecture, defering to the above-mentioned works, in which the agents see the CBR system just as a reasoning tool. Our proposal is also concern with the agent's implementation and presents a "formalism" easy to implement, in which the reasoning process is based on the concept of intention. In this model intentions are cases, which have to be retrieved, reuse, revise and retain. This is the main difference between this work and the work presented in [MPA99] [Ol99], in which cases have a more complex structure.

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

Case: Problem, Solution, Result
  Problem: initial_state
  Solution: {action,[intermediate_state]}*
  Result: final state

{ }: Sequence,
[]: Optional,
*: 0 or n repetitions,
|: or

Fig. 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 state is considered a *belief*; a belief may also be the objective to reach. The *intentions* are plans of action that the agent is forced to carry out in order to achieve its objectives [BIP88], so an *intention* is an ordered set of actions; each change from state to state is made after carrying out an action (the agent remembers the action carried out in the past when it was in a determined situation and the result obtained). 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 a past one, it will try to build similar result to the previously obtained).

Belief: state; **Desire**: {final_state}*; **Intention**: {action}*

Fig. 3. Definition of the mental attitudes of a BDI agent.

The relationship between CBR systems and BDI agents can be established implementing a case as a set of beliefs, an intention and a desire which caused the resolution of the problem. The straight relationship between BDI agents and CBR systems can be identified comparing figures 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 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.

3.1 Notation

In order to show how to implement such agents, a formal notation is introduced to describe the CBR systems and their reasoning process. The following section defines the life cycle of a CBR system in terms of the mental attitudes of the BDI agents. Situated in a world or in an environment, an agent senses and effects its environment. The CBR-BDI agent is denoted by a 7-tuple <E, GAL, CM, PAL, EK, O, M>. 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:

E: Environment E described by a set of values for a fixed set of non-null environment variables, denoted as $E = <e_1, e_2, ..., e_n>$ where $e_i \in E$, $i \in \{1,2,...,n\}$, therefore E will describe the current environment. Each variable e_i is a 3_tuple $<name\_var_i, value_i, index_i>$ where $name\_var_i$ is the name of the variable, $value_i$ is a set of values for each variable and $index_i$ is a boolean field, *1* if the variable is in the index and *0* in other case.

GAL: General Actions Library $GAL = <ga_1, ga_2, ..., ga_n>$ where $ga_i \in GAL$, $i \in \{1,2,...,n\}$, ga_i is described as a 3_tuple $<Name_i, InPar_i, OuPar_i>$, where $Name_i$ is the name of the action, $InPar_i$ is a set of input param $InPar_i \subseteq E$ and $OuPar_i$ is a set of output param $OuPar_i \subseteq E$.

CM: Case Memory CM stores sets of previous cases $CM = <c_1, c_2, ..., c_n>$, each case c_i is formulated as a 3-tuple $<B, D, I>$ representing past beliefs, past desires and past intentions.

B: Agent's beliefs are the exact values for each variable of the environment, for any instance of time. $B = <b_1,b_2,...,b_n>$ where each belief is a state, which is a 2_tuple $b_i = <E, valueE>$. $E = <e_1, e_2, ...,e_n>$ where e_i are the variables which describe the environment and $valueE = <v_1,v_2,...,v_n>$ where $v_i \in value_i$, $i \in \{1,2,...,n\}$, it is the exact value for each variable on a certain instant.

D: Desires *D* corresponds to set of objectives, which determine the resolution of the problem. $D = <d_1, d_2, ..., d_n>$, where $d_i \in B$, $D \subset B$.

I: Intention *I* represents a trigger for corresponding plan from the producer actions library PAL. They are previous actions, and have the values of the input parameter and output parameter.

PAL: Producer Actions Library is a collection of actions. Each intention is an ordered sequence, $I = <a_1, a_2, ..., a_n>$ where each change from state to state is produced after carrying out an action, $a_i \in PAL$, $i \in \{1,2,...,n\}$, a_i is a 3_tuple $<Name_i, ValueInputPar_i, ValueOutputPar_i>$ where $Name_i \in ga_i$, ga_i is a general action from general actions library GAL, $ValueInputPar_i$ and $ValueOutputPar_i$ are sets of exact values for input param and output param of ga_i. $ValueInputPar_i$ =

$<v_{i1},v_{i2},...,v_{in}>$ and $ValueOutputPar_i = <v_{i1},v_{i2},...,v_{in}>$ where $v_{ij} \in value_i$, $i,j \in \{1,2,...,n\}$.
EK: Expert Knowledge is composed of a set of default rules associated with case adaptation and case retain process. The procurement of these rules is automated. EK = $<r_1,r_2,...,r_n>$ where r_i is a 2_tupla (Previous, Consistent); Previous = $< v_1,v_2,...,v_r >$ where $v_i \in value_i$, $i \in \{1,2,...,r\}$, it is the exact value for some varibles. And Consistent = $<v_{r+1},...,v_n>$ where $v_i \in value_i$, $i \in \{r+1, r+2,...,n\}$, is the exact value for others variables.
O: It regarded as a set of current goals (O) in a particular belief-world. The objectives are appropriate final states in the environment. $O = <o_1, o_2, ..., o_n>$, where $o_i \in B$ and $i \in \{1,2,...,n\}$, $O \subset B$. This set is null if the environment is not definite.
M: Set of functions of similitude. A function of similitude determines the degree of equality between two states. $M = <m_1, m_2, ..., m_n>$, where $m_i \in M$ and $i \in \{1,2,...,n\}$.

When a CBR system receives a new problem b_{new}, it will be obtained a final state b_n and the intermediate states $<(b_{new},a_1), (b_2,a_2), ..., (b_n,a_n)>$, before reaching the final state.

3.2 The Case-based Reasoning Cycle

The four phases of the life cycle of a CBR system are now outlined. A different metric or technique should be identified for each of the CBR steps depending on the problem to which the agent is applied and the environment in which it evolves.

Retrieve:
The cases similar to the new problem b_{new}, $(c_1,c_2,...,c_k)$ are retrieved from the case-base CM using a similarity metric m_1. The retrieved cases are $c_1,c_2,...,c_k$.

Reuse:
A first solution $<(b_{new},a_1), (b_2,a_2), ..., (b_n,a_n)>$ is obtained from the retrieved cases and the problem b_{new} using the metric m_2. This initial solution is a plan, an ordered sequence of states and actions.

Revise:
In this phase the plan $<(b_{new},a_1), (b_2,a_2), ..., (b_n,a_n)>$ obtained in the previous phase is evaluated. It most be checked if the final state b_n and the plan developed to achive it are adequate. The revision can be carried out using Expert's Knowledge (rules) or simulation techniques [CAR01]. The procurement of rules can be automated using techniques such as rought-sets [Pa01]. 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 $<(b_{new},a_1), (b_2,a_2), ..., (b_n,a_n)>$ is indexed and stored in the correspondent case-base CM. This process implies that knowledge contained in memory changes. So the rules which represent the expert's knowledge may also change. This process of change is carried out using techniques of Belief-revision [GR95].

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 for the retrieval, reuse, revise and retain steps have to be defined. Besides, rules that describe the Expert's knowledge must be established. 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.

4 Case study: The E-business Engineering Sales Support System

The construction industry is an information intensive economic sector. 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. 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. 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 very often 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 price, the personnel and material required to carry out the work. The system here outlined was developed to facilitate the work of the sales force, in particular to the inexperienced personnel, and the estimation of costs, reducing the process bureaucracy.

4.1 The System Architecture

In the expansion policy of B&D Constructions one of the main points is its incorporation to 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 to facilitate the communication among 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, which estimates the construction cost, the personnel and the 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 in a predetermined web address and search for new ones using a natural language processing strategy. This strategy is optimised for the Web and it is based on the combination of the grammatical characterisation of words and statistical decision techniques [Co01]. Assistant agents are interface agents that facilitate the communication between the salesmen and the planning agent, they also hold summarized information about the clients visited by the owner salesman and by the rest of the salesmen.

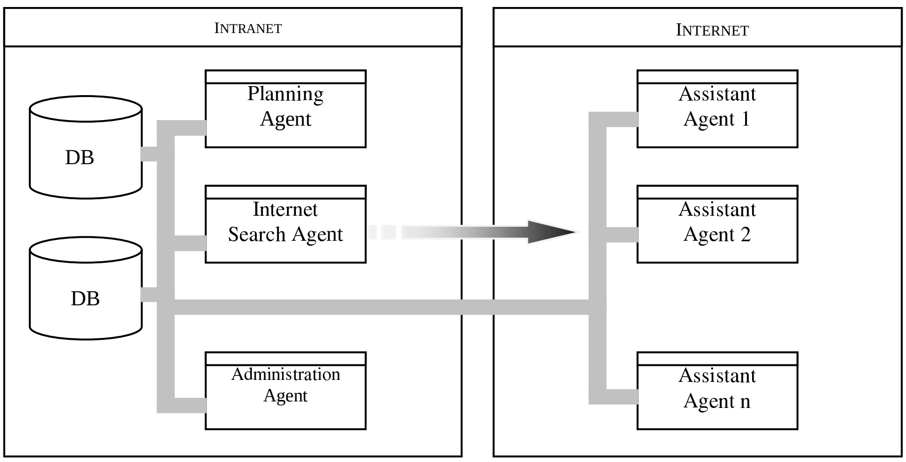

Fig. 4. Agent-oriented e-business architecture.

Before a salesman visits a client, he interrogates his assistant agent for a description of the client. The assistant agent compares this data with previous queries and if a match is found, using relaxed K-nearest neighbor algorithms, the data stored about the client is presented to the salesman [WG99]. 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 that client, location information and other 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 demands more information, his assistant agent contacts the planning agent, which searchs for information about the client in its case-base. This agent also interrogates the internet search agent asking for general information about clients. The internet search agent obtains information from the web, analyses and indexes it using a natural language processing algorithm optimized 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 and the rest of the elements of the system: agents, databases and, even, salesmen.

When constructing the e-business multiagent system here presented, a decentralised architecture was selected, in which agents interact between each other when they 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.

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 protection [CAR01]. This security system protects the databases and the information stored in the system from external “agents” or non accredited personnel.

4.2 A CBR-BDI Planning Agent

Since our intention is to develop a dynamic distributed e-business solution it may be adequate to use agents with adaptation and learning capabilities. The working mode of the planning agent will be explained to show how the agents presented in section 3 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 achaive 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, for example:

Environment variables:
Building Type = BT, Insulating Material = IM, Construction Year = CY, Heating Pipe = HP, Heating Radiators = HR, Heating System = HS
Other variables: Material-price, workers required / ratio hours/men, pp, installation kit.

General Actions:
Action GA1: $ACTION_1$ (Material-price, workers required/ ratio hours/men; pp, installation kit)
Action GA2: $ACTION_2$ (Material-price, workers required/ ratio hours/men; HR, installation kit)
Action GA3: $ACTION_3$ (Material-price, workers required/ ratio hours/men; HP, installation kit)

Producer Actions:
Action A1: $ACTION_1$ (Installation of heating pipes: 200 €, 3t/2d, 85 m pp+elements, installation kit27)
Action A2: $ACTION_2$ (Installation of radiators: 266 €, 2t/1d, 16 radiators +elements, installation kit13)
Action A3: $ACTION_3$ (Installation of diesel heating system : 350 €, 2t/0.5d, diesel heating+elements, installation kit12)

Beliefs:
Belief B1: $STATE_1$ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973)
Belief B2: $STATE_2$ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m)
Belief B3: $STATE_3$ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m, HR=RS16-16-22)
Belief B4: $STATE_{FINAL}$ (BT=h2; Size=225; Heating=D2; IM= I7; CY=1973)

Intention:
$STATE_1$ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973)→
$ACTION_1$ (Installation of heating pipes: 200 €, 3t/2d, 85 m pp+elements, installation kit27)→
$STATE_2$ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m)→
$ACTION_2$ (Installation of radiators: 266 €, 2t/1d, 16 radiators +elements, installation kit13)→
$STATE_3$ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m, HR=RS16-16-22)→
$ACTION_3$ (Installation of diesel heating system : 350 €, 2t/0.5d, diesel heating+elements, installation kit12)→
$STATE_4$ (BT=h2; Size=225; Heating=no; IM= I7; CY=1973; HP=H2300-45m, HR=RS16-16-22, HS=B26-600)= $STATE_{FINAL}$ (BT=h2; Size=225; Heating=D2; IM= I7; CY=1973)

$STATE_1$ 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 by described as in $STATE_{FINAL}$. $ACTION_1$ indicates that the cost of the installation of the heating pipes required is 200 €, 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.

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 Method [FC01] 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.

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 them to the new problem. At this point, two possibilities occurr. First, that the environment was, in the past, in a state that is almost identical to the new problem, b_{new}, and the same solution required now was obtained. In that case, the same sequence of action than in the past can be carried out. In the second case, the recuperated cases are similar to the present state, b_{new} but with some differences. So 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.

Once the graph has been constructed, the algorithm of Dijkstra is used [SWW99] 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.

Revise:
The revision process is carried out using an expert's knowledge, which is represented as rules.

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 a 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. Besides, the rules used in the revise phase should be analysee, this process is carried out using Belief Revision techniques [Pavón *et al.*, 2001]. The rules are updated automatically using a Belief Revision technique that uses Epistemic Entrenchment, as constructive model.

5 Results and Conclusions

The acceptance of this agent-based distributed system, by the Company staff, has been excellent. For evaluation purposes, during the testing period, the case-base of the planning agent has been fed with 320 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 6200 installations from January 1997). The system was interrogated in 35 occasions and its output was qualified as adequate by experienced salesmen.

In 32 occasions the estimation differed in less that 4% of the one given by an expert, and in 3 occasions differed in less than 11%. These deviations were caused by two different reasons, in the first case the client required a combination of installations and equipment, which did not appear in any of the 320 cases stored in the agent case-base. The other two errors were caused by human mistakes, a misinterpretation between the salesman and the client. Therefore from the point of view of the planning agent, these last two errors should not be taken into account. In the other case, the error could be minimised during the review phase. With 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 35% of the actual cost, and the time of the sale up to 50%.

The CBR-BDI architecture presented in this paper solves one of the problems of the BDI 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. New cases are continuously introduced and older ones are eliminated.

Morá [Mo98] 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. The work presented in this paper is the basis of a wider project that aims to construct a tool for developing dynamic agent-based e-business solution in flexible and efficient manner.

Bibliography

[AP94] Aamodt A. and Plaza E. (1994) Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7. No 1, March.

[BW98] Bergmann R and Wilke W. (1998) Towards a New Formal Model of Transformational Adaptation in Case-Based Reasoning. European Conference an Artificial Intelligence (ECAI'98), John Wiley and Sons.

[Br87] Bratman M.E. (1987) Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A.

[BIP88] Bratman M.E., Israel D., and Pollack M.E. (1988) Plans and resource-bounded practical reasoning. Computational Intelligence, 4. pages 349-355.

[CS90] 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, 18(2). pages 215-227.

[CL90] Cohen P.R. and Levesque H.J. (1990) Intention is choice with commitment. Artificial Intelligence, 42(3).

[Co01] Corchado J. M. (2001) CBR Agents in an e-commerce environment. Proceedings of WOOPS-ECOOP, Budapest, June.

[CL01] Corchado J. M. and Lees B. (2001) A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence. Vol 15, no. 2, pp105-127.

[CAR01] Corchado J.M., Aiken J, Rees N. (2001) Artificial Intelligence Models for Oceanographic Forecasting. Plymouth Marine Laboratory. ISBN-0-9519618-4-5.

[FG94] Feret M. P. and Glasgow J. I. (1994) Explanation-Aided Diagnosis for Complex Devices, Proceedings of the 12th National Conference an Artificial Intelligence, (AAAI-94), Seattle, USA, August 94.

[FC01] 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, April 2001.

[GR95] Gärdenfors P. and Rott H. (1995) Belief Revision. In Handbook of Logic in Artificial Intelligence and Logic Programming, volume IV, chapter 4.2.

[GL86] Georgeff M.P. and Lansky A.L. (1986) Procedural knowledge. In Proceedings of the IEEE Special Issue on Knowledge Representation, volume 74. pages 1383-1398.

[Je92] Jennings N.R. (1992) On Being Responsible. In Y. Demazeau and E. Werner, editors, Decentralized A.I. 3. North Holland, Amsterdam, The Netherlands.

[KG91] 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), pages 82-88, Sydney, Australia.

[Ki94] 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.

[MPA99] Martín F. J., Plaza E., Arcos J.L. (1999). Knowledge and experience reuse through communications among competent (peer) agents. International Journal of Software Engineering and Knowledge Engineering, Vol. 9, No. 3, 319-341.

[Mo98] Móra M. C., Lopes J. G., Viccari R. M. and Coelho H., (1998) BDI Models and Systems: Reducing the Gap. ATAL-98.

[NLC96] 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, Lausane, Lecture Notes in Artificial Intelligence, 1168, pp295-308. Springer-Verlag, Berlin, 1996.

[Ol99] 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

[Pa01] 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, Volume 8 No 1, February 2001. ISSN 1352-9404.

[RG91] 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.

[RG95] Rao A. S. and Georgeff M. P. (1995) BDI Agents: From Theory to Practice. First International Conference on Multi-Agent Systems (ICMAS-95). San Franciso, USA, June.

[RS89] Riesbeck, C. K. and Schank, D. B. 1989. Inside Case Based Reasoning. Erlbaum.

[SWW99]Schulz F., Wagner D. and Weihe K. (1999) Dijkstra´s Algorithm On-line: An Empirical Case Study from Public Railroad Transport. Algorithm Engineering . pages 110-123.

[Sh93] Shoham Y. (1993) Agent-Oriented programming. Artificial Intelligence, 60(1): pages 51-92.

[WG99] 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.

[WM94] Watson I. and Marir F. (1994) Case-Based Reasoning: A Review. Cambridge University Press, 1994. The knowledge Engineering Review. Vol. 9. Nº3.

[WL98] Wendler J. and Lenz M. (1998) CBR for Dynamic Situation Assessment in an Agent-Oriented
Setting. Proc. AAAI-98 Workshop on CBR Integrations. Madison (USA) 1998.

[Wo99] Wooldridge M. (1999) Intelligent Agents. Multiagent Systems. A modern approach to Distributed Artificial Inteligence. Edited by Gerhard Weiss. Pages 27-77.

[WJ94] Wooldridge M. and Jennings N. R. (1994) Agent Theories, Architectures, and Languages: A Survey. Procs. ECAI-94 Workshop on Agent Theories, Architectures, and Languages.