Computing and Information Systems **8** (2001) p.33-39

Creation of Deliberative Agents Using a CBR Model

Rosalia Laza Fidalgo and Juan Corchado Rodriguez

This paper shows how to build deliberative agents, with learning capacity, using a Case-based Reasoning model. 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 is shown how to solve some of their problems, especially those related to their implementation and adaptation using a Case-based Reasoning System.

1 INTRODUCTION

This paper shows how to build deliberative agents with BDI architecture (*belief, desire and intention*) using a case-based reasoning model. The proposed methodology solves some of the problems derived from using this type of agent, since it facilitates automation of their construction and provides them with a learning capacity. Besides, it defines a framework that facilitates the adaptation of such agents to new situations.

In most computing systems, all executed actions are previously planned and encoded by a programmer. But, 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 which allow them to decide what to do according to their objectives. Such systems are known as agents [Wooldridge, 1999].

This article shows a methodology that facilitates the construction of agents that are able to work in ever-changing, unpredictable and even open environments; that is to say, in dynamic environments. Thus agents must be autonomous, reactive, pro-active, sociable and have learning capacity. They must be able to respond to events which take place in their environment, to take the initiative according to their goals, to interact with other agents (or humans) and to use past experiences to obtain present goals.

There are different types of agents and they can be classified in different ways [Wooldridge *et al.*, 1994]. One of these types are the so-called deliberative BDI agents, 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 [Wooldridge, 1999].

Agents with BDI architecture have their origins in the *practical reasoning* of traditional philosophy. These agents are supposed to be able to decide at 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 goals [Wooldridge, 1999].

The BDI architecture provides the agents with mental attitudes of beliefs, desires and intentions, which respectively represent the information, the motivation and their deliberative states. 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 [Bratman, 1987; Kinny and Georgeff, 1991].

Formalisation and implementation of BDI agents constitutes the research of many scientists [Cohen and Levesque, 1990; Jennings, 1992; Kinny *et al.*, 1994; Rao and Georgeff, 1991; Georgeff and Lansky, 1986; Mueller *et al.*, 1994; Shoham, 1993]. Some of these researchers question the necessity of studying multi-modal logic for formalisation and construction of such agents, because these logics have not been completely axiomatised and are not computationally efficient. Rao and Georgeff (1995) state that the problem is that there is a big distance between the powerful logic for BDI systems and the practical systems. Another estimation 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 training.

This article shows how a cased-based reasoning system (CBR) can substantially solve the two problems that have been previously mentioned. Case-based training facilitates learning based on previous experiences (cases), and it cannot be imagined without the cognitive process that involves the procurement of a new experience. Among the different disciplines of the cognitive science, cognitive psychology has shown the importance of learning from experience [Caplan *et al.*, 1990; Ross *et al.*, 1990; Schielmann *et al.*, 1989]. 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, the result will be an agent with beliefs, desires, intention and also with learning capacity.

This article analyses BDI agents and case-based reasoning systems, then makes a formal description of the proposed model in which a CBR system is used to operate the mental attitudes of the deliberative agents, and finally presents some conclusions.

2 DELIBERATIVE AGENTS

Deliberative agents with BDI architecture are composed by 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, that is to say, the action plan that they have decided to carry out [Weiss, 1999].

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 *beliefs* represent the knowledge of the agent about the environment at the present moment and in previous moments. The *desires* (objectives) are another essential component of the state of the system, they identify some desired final state. For example a human remembers the missing of a train or an unexpected puncture in his car's tyre because he knows where he is (through his beliefs) and remembers what he wants to get (through his desires) [Kinny *et al.,* 1993]. BDI agents act in the same way, and they compose an action plan for the situation in which they are (through their intentions). But since environmental changes are unpredictable, agents have to decide if they continue with the previous decisions or they need to plan new ones. Which is the right approach? Not necessarily any of them: agents need to be committed to plans and objectives, but they also need to be able to reconsider them when appropriate. These committed plans constitute the agent's *intentions* [Kinny *et al.*, 1993].

This article proposes a novel mechanism of implementing BDI agents that facilitates their training, and which is based on a case-based reasoning system.

3 CASE-BASED REASONING SYSTEMS

The idea which impelled the development of case-based reasoning (CBR) systems is centred in the fact that human beings employ what they have learned in previous experiences to solve present problems [Kolodner 1983a, 1983b; Joh, 1997]. Case-based reasoning systems solve problems through the adaptation of solutions previously given to similar problems [Riesbeck *et al.*, 1989].

The CBR systems analyse and obtain solutions through algorithms of index, recuperation, comparison

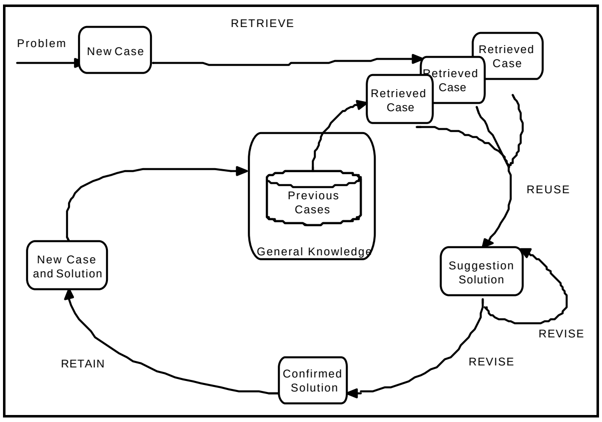

Figure 1: CBR Life Cycle.

techniques and adaptation of problems to a determined situation. To do this, 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 sequence: *retrieve, reuse, revise, and retain* [Kolodner, 1993; Aamodt, 1994; Watson *et al.*, 1997]. The mission of the algorithm of *retrieve* consists of looking for and selecting in the memory of the CBR, the cases that are more similar to the present problem. The recovered cases are *adapted* to generate a possible solution. Such solution is *reviewed* and if it is appropriated a new case is created and *stored* in the memory. A CBR is a system of increasing training since each time a problem is solved, it is possible to create a new case and store in the CBR memory for its further use.

The automation capabilities of CBR systems has led us to establish a relationship between the cases, the CBR cycle of life, and the own mental attitudes of beliefs, desires and intentions of the BDI agents. This way, we put forward a mechanism, which facilitates the implementation of the BDI agents using the reasoning cycle of a CBR system. Besides, this form of implementation allows the adaptation of new situations and provides these agents with a learning capability.

4 FORMALISATION OF A CBR SYSTEM

This section discuses the relationships which can be established between the BDI agents and the case-based reasoning systems, and makes a formal description of the life cycle of a typical CBR system and the mental attitudes of BDI agents.

4.1 BDI agents and CBR systems

The structure of CBR systems has been designed around the concept of a case. A problem, a solution and the result that was obtained once the proposed

CBR	{}: Sequence,
Case: Problem, solution, result	[ ]: Optional,
Problem: initial_state	*: 0 or n repetitions,
Solution: {action, [intermediate_state]}*	+: 1 or n repetitions,
Result: final_state	|: or.

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

BDI
Belief: {final_state}| {state, action, state}
Intention: {Beliefs}$^+$
Desire: {final_state}$^+$

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

solution has been applied usually make a case. Figure 2 shows such 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 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). It also may happen that the agent is carried out in an environment in which it knows the final objective it wants to reach, thus, it is also considered a *desire* (the agent knows the desired final state). The *intentions* are plans of action that the agent is obliged to carry out in order to satisfy its objectives [Bratman *et al.*, 1988], so an *intention* is a set of *beliefs* that have been planned, so the actions fulfilled are sequentially ordered. 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).

Finally, figure 4 shows the relationship between the components of the CBR systems and the attitudes of the BDI agents. From now on, cases and intentions will be considered synonyms and a case will be composed of beliefs.

The following section defines the life cycle of a CBR system in terms of such mental attitudes.

CASE = INTENTION = SET OF BELIEFS

Figure 4: Correspondence between the elements of a BDI agent and a CBR system.

4.2 Definition of a CBR system

To model a BDI agent with reasoning capacity a formal description of a CBR system is done. This description implies a definition of the beliefs, desires and intention 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 of $B \in \beta$, is a finite set of case which is indexed. So, a Case-base can be defined as a tuple $(\{c_1, c_2, \ldots, c_n\}, \tau)$. Where $\{c_1, c_2, \ldots, c_n\}$ are the cases, and τ is the finite set of characteristics by which the cases are indexed.
- Case (c): This represents a past experience. A case is represented by a sequence of states of the environment. ($c = \{$ initial_state , { action x [intermediate_state] }$^+$, final_state} or c={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 <u>initial states</u> (ini_state), represents the problems that have to be solved and that constitute the entrance to the CBR system.
    2. Set of <u>intermediate states</u> (inter_state), represents the different states that the environment undergoes before obtaining a desired final state.
    3. Set of <u>final states</u> (final_state), represents the results obtained after carrying out a series of actions starting from a concrete state.

Besides, a case has:

4. Set of <u>actions</u> (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 resolve, the final states of some cases would be part of the desires to reach. $d \in D$, $D \equiv$ {final_state}$^+$.
- Set of functions of similitude (A): A function of similitude determines the degree of equality between two states. A function of similitude between a problem to be solved st_n and a case

CBR	ENTRANCE	Exit
	$(st_n, B) \in ini\_state \times \beta$	$((act,(st_{inter})^*)^+, st_{fin}, B) \in (actions \times inter\_state \times final\_state \times \beta) \cup \{\perp\}$

Table 1: General representation of a CBR system.

1.	$(c_1,c_2,\ldots,c_k, A) \leftarrow$ Retrieve (st_n,B) with $c_k = \{s_{tk}, \{act_{\_ki}, \{st_{inter\_ki}\}^*\}^+, st_{fin\_k}\}$ k>0 and i>0
2.	$(st_n, \{act_{\_ni}, \{st_{inter\_ni}\}^*\}^+, st_{final\_n}) \leftarrow$ Reuse $(st_n, (c_1,c_2,\ldots,c_k), A)$
3.	$(st_n, \{act_{\_ni}, \{st_{inter\_ni}\}^*\}^+, st_{final\_n}) \leftarrow$ Revision (st_{fin_n})
4.	$(st_n, \{act_{\_ni}, \{st_{inter\_ni}\}^*\}^+, st_{final\_n}, B) \leftarrow$ Learning (c_n,B)

Table 2: Life cycle of a CBR system

RETRIEVE Phase	ENTRANCE	EXIT
	st_n (present state)	Similar states to the present one and actions carried out to reach these states.

Table 3: Retrieve phase.

$c_1=\{st_{ini}, \{act, \{st_{inter}\}^*\}^+, st_{fin}\}$, is applied between $st_n - st_{ini}$ and between $st_n - st_{inter}$.

When a new state/case st_n, that belongs to a Case-base B, is presented to the CBR system, it will be obtained or not a final state st_{fin} and the intermediate states st_{inter}, that the environment underwent before reaching such final state, as it is represented in table 1.

Table 2 describes the retrieve, reuse, revision and training phases that make up its cycle of life, using the terminology just introduced.

4.2.1 Life cycle of a CBR system

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

Retrieve: $(c_1,c_2,\ldots,c_k, A) \leftarrow$ Retrieve (st_n,B)

During this phase those cases similar to the new problem st_n are retrieved from the case-base B using a similarity metric A. The retrieved cases are $c_1,c_2,\ldots,c_k$. During the retrieval step it is identified the case-base in which similar cases to the new problem are stored, so the index of the new case/problem is found. Only those cases that are indexed by the same index that the present case are indexed. Table 3 shows the result of the retrieve phase.

This phase can be divided into two subphases:

- Indexing: First the index by which the problem to solve is indexed has to be found (to do so algorithms such as ID3 (Quinlan, 1979), Neuronal Nets (Corchado *et al.*, 2000), etc., can be used. This article does not deal with automatic indexing).

  i $\leftarrow$ Indexing (stn)

- Selection: then, a similarity transform is applied (for example the Cosine, Sine function) (Laza *et al.*, 1999) to those cases whose index coincides with that of the new case and the most similar are retrieved. Starting from these retrieved cases it can be supposed that from the new case an equal or very similar case will be reached and so the recovered final cases are considered desires to be reached. A previous situation is remembered and the desire is to reach the same objective that was obtained in the past. So, the desires are acquired in a dynamic way, depending on the retrieved cases.

  $(c_1,c_2,\ldots,c_k, W)$ Selection (i, st_n, B)
  where $c_1,c_2,\ldots,c_k$ are cases with similar states to st_n. Each case $c_k = \{st_k, \{act_{\_ki}, \{st_{inter\_ki}\}^*\}^+, st_{finalk}\}$ being $st_{finalk} \equiv d_k$; $d_k \in D$.

Reuse: $(st_n, \{act_{\_ni}, \{st_{inter\_ni}\}^*\}^+, st_{final\_n})$ Reuse $(st_n, (c_1,c_2,\ldots,c_k), A)$

During this phase a first solution $(\{act_{\_ni}, \{st_{inter\_ni}\}^*\}^+, st_{final\_n})$ is obtained from the retrieved cases and the problem case(st_n). The first solution can be obtained using the sequence of actions carried out in the past, or modifying the sequence of actions adapting it to the new problem. Table 4 shows the result obtained in this phase.

If the cases retrieved in the previous phase are the following:

$$est_{k2} \xrightarrow{a21} est_{k21} \xrightarrow{a22} est_{k22} \xrightarrow{a23} est_{k23} \xrightarrow{a24} est_{k24} \xrightarrow{a25} est_{f2}$$

$$est_{k3} \xrightarrow{a31} est_{k31} \xrightarrow{a32} est_{k32} \xrightarrow{a33} est_{k33} \xrightarrow{a34} est_{k34} \xrightarrow{a35} est_{f3} \tag{1}$$

There can be two possibilities. First, that the environment was, in the past, in a state that is almost identical to the new state est_n, so it can be carried out the same sequence of actions than in the past. Second, that the retrieved cases are similar to the present state but with some differences. In the second case a sequence of actions, that is a mixture of those that were retrieved from the previous phase, is constructed. To do so, an acyclic directed graph is created, whose first vertex is the new 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

Reuse phase	Entrance	Exit
	Similar states and actions carried out to reach such states.	Sequence of actions to carry out in order to solve the new problem, intermediate states and final state. $(st_n, \{act_{\_ni}, \{st_{inter\_ni}\}^*\}^+, st_{final\_n})$

Table 4: Reuse phase.

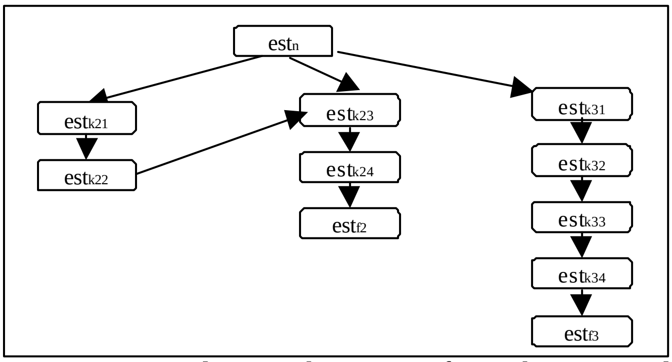

Figure. 5: Acyclic Graph starting from the retrieved cases in the retrieval phase.

it a similarity transform to all the retrieved states; those that are more similar will determine what action will be carried out starting from the new state. This process is repeated with the cases that are obtained until a final state is reached.

Figure 5 clarifies the exposed process. Starting from the retrieved cases in (1), and after applying a similarity transform the result is that the closest states to st_n are st_{k2}, st_{k22} and st_{k3}. So from st_n the actions of 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 retrieved in the previous phase, if they were not the same, the transform would be applied again to the obtained states.

Once the graph has been constructed, the algorithm of Dijkstra is used [Schulz *et al.*, 1998] 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 st_n, st_{k23}, st_{k24}, st_{f2}.

Revision: $(st_n, \{act\_{ni}, \{st_{inter\_ni}\}^*\}^+, st_{final\_n})$ Revision (st_{final_n})

This phase determines if the result obtained in the previous phase is correct or not. That is to say, it checks if the final state of the environment that can be obtained when carrying out the actions proposed in the previous phase is adequate. In order to carry out the revision techniques of simulation of results can be used [Corchado *et al.*, 2000], Belief-revision [Gärdenfors *et al.*, 1995]. Table 5 schematises this process.

Revision phase	Entrance	Exit
	New state, intermediate states and Final states	Revised final state (accepted or not)

Table 5: Revision phase.

Training phase	Entrance	Exit
	Initial_state, actions, intermediate_state, final_state.	New case.

Table 6: Training phase

Retain:

During this phase solved cases and the way in which they are solved (the new state, the intermediate state, the final state and the actions that were carried out to move from one state to the other) are indexed and stored in the correspondent Case-base. Table 6 indicates that the aim of this phase is to store new cases (new beliefs) in the Case-base.

4.3 The reasoning process of CBR-BDI agents

As 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 believes, 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 is 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 believes, desires and intentions could appear.

A tool, called GABDI, has been developed to facilitate the implementation of such agent. This tool also helps the users to understand the reasoning process of the agent and provides information about the different states in which the agent is. GABDI generates a code in Java, using some libraries provided by Toshiba's Bee Gent tool. This tool allows implementing each one of the states an agent will go through (as if a Petri net would be defined with its states and transitions, in this case each state of the Petri net is a sub-class in Java).

5 CONCLUSIONS AND FUTURE WORK

This architecture 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 and Coelho (1998) have described the gap that exists between the formalisation and the implementation of BDI agents. What we propose in this article is defining the beliefs and intentions clearly (they don't need to be symbolic or completely logic) and using them in the life cycle of the CBR system, to obtain a direct implementation of a BDI agent. To automate all the process described in this article a tool has been developed, the GABDI (Generator of Agents BDI), which automatically generates BDI agents.

This article has shown how single agents can be developed with this technology. This initial prototype has been successfully applied to several e-commerce problems. The work presented in this paper is the basis of a wider project that aims to automate the creation of community of agents. Therefore we are defining negotiation and communication protocols for such agents and adapting the CBR model for agents that work with in a community.

Bibliography

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

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

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

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.

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

Corchado J.M., Aiken J, Rees N. (2000) *Artificial Intelligence Models for Oceanographic Forecasting*. Plymouth Marine Laboratory. ISBN-0-9519618-4-5.

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

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.

Georgeff. M.P. and Lansky, A.L. (1987) *Reactive reasoning and planning*. Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87). Seattle, WA.

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

Joh D.Y. (1997) *CBR in a Changing Environment*. Case-based Reasoning Research and Development. ICCBR-97. Providence, IR, USA.

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.

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.

Kinny D. and Georgeff. M.P (1993) *Commitment and effectiveness of situated agents*. Proceedings of Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93) Chambery, France.

Kolodner J. (1983a) *Maintaining organization in a dynamic long-term memory*. Cognitive Science. Vol. 7. pages. 243-280.

Kolodner J. (1983b) *Reconstructive memory, a computer model. Cognitive Science*. Vol. 7. pages. 281-328.

Kolodner J. (1993) *Case-Based Reasoning*. Morgan Kaufmann.

Laza R., Fernández F. and Corchado J.M. (1999) *Sistemas de Razonamiento Basado en Casos para el Soporte a la toma de decisiones*. III Jornadas de Transferencias Tecnología de Inteligencia Artificial (TTIA'99), Murcia.

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

Muller J., Pischel M., and Thiel M. (1994) *A pragmatic approach to modelling autonomous interacting systems: Preliminary report*. In Intelligent Agents: Theories, Architectures, and Languages. Lecture Notes in Artificial Intelligence LNAI 890, Amsterdam, Netherlands. Springer Verlag.

Quinlan J.R. (1979) *Discovering rules from large collections of examples: a case study*. Edinburgh University Press.

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.

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.

Riesbeck C.K., Schank R.C. (1989) *Inside Case-Based Reasoning*. Lawrence Erlbaum Ass. Hillsdale.

Ross B.H., Perkins S.J, and Tenpenny P.L. (1990) *Reminding-based Category Learning*. Cognitive Psychology, 22. pages 460-492.

Schielmann A.D. and Acioly N.M.(1989) *Mathematical Knowledge Developed at Work: The Contribution of Practice Versus the Contribution of Schooling*. Cognition and Instruction, 6(3): pages 185-221.

Schulz Frank, Wagner Dorothea and Weihe Karsten. (1999) *Dijkstra´s Algorithm On-line: An Empirical Case Study from Public Railroad Transport*. Algorithm Engineering . pages 110-123.

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

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

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

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

R. Laza Fidalgo is a Personal Professor at the University of Vigo at Ourense. J. M. Corchado Rodriguez is a Professor at the University of Salamanca.