

# A reasoning model for CBR\_BDI agents using an Adaptable Fuzzy Inference System

R. Laza<sup>1</sup>, R. Pavón<sup>1</sup> y J. M. Corchado<sup>2</sup>

<sup>1</sup>Departamento de Informática  
Universidad de Vigo  
Campus As Lagoas, s/n, 32004, Ourense, Spain  
{[rlaza\\_pavon](mailto:rlaza_pavon@uvigo.es)}@uvigo.es

<sup>2</sup>Departamento de Informática y Automática  
Universidad de Salamanca  
Plaza de la Merced s/n, 37008, Salamanca, Spain  
[corchado@usal.es](mailto:corchado@usal.es)

**Abstract.** This paper proposes to automate the generation of shellfish exploitation plans, which are elaborated by Galician extracting entities. For achieving this objective a CBR-BDI agent will be used. This agent will adapt the exploitation plans to the environmental characteristics of each school of shellfish. This kind of agents develops its activity into changing and dynamic environments, so the reasoning model that they include must be emphasised. The agent reasoning model is guided by the phases of the CBR life cycle, using different technologies for each phase. The use of an adaptative neuro-fuzzy inference system in the reuse phase must be highlighted.

## 1 Introduction

There are different types of agents and they can be classified in different ways [20]. One of these types are the so-called deliberative agents with a BDI architecture, which are characterized for having mental attitudes of Beliefs, Desires and Intentions; besides they have capacity to decide what to do and how to get their objectives according to their attitudes [20] [9] [14] [2].

Formalisation and implementation of BDI agents constitutes the field of research of many scientists [9] [14] [8] [16]. 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 [13] state that the problem lies in the wide 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 must be constantly adding, modifying or eliminating beliefs, desires and intentions. Therefore it would be convenient to include a reasoning mechanism which involves a final apprenticeship.

The developed job shows how to build deliberative agents, using a case-based reasoning system (CBR), that solves the problems quoted previously. In the reasoning

process of these agents, a GCS network (*Growing Cell Structures*) and an ANFIS model (*Adaptive Neuro-Fuzzy Inference Systems*) are utilized. The GCS network is used in the retrieve phase whereas the ANFIS model implements the phase of adaptation.

This paper is structured as follows. In section 2 the reasoning model of CBR-BDI agents is detailed. Section 3 proposes to automate the generation of shellfish exploitation plans, that are elaborated by Galician extracting entities; the results are also analysed in this section. Finally, in section 4 some conclusions are exposed.

## 2 Reasoning Model of CBR-BDI Agents

The relationship between CBR systems and BDI agents can be established implementing a case as a set of beliefs, together with an intention and a desire which caused the resolution of the problem. Using this relationship agents can be implemented (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 [3] [4] [5] [6]. Once the beliefs, desires and intentions of an agent are identified, the reasoning model can be established, in the way presented in this section.

The reasoning cycle of a typical CBR system includes four steps that are cyclically carried out in a sequenced way: retrieve, reuse, revise, and retain [1]. In the cases base, all experiences which can be used by a CBR-BDI agent are stored. Therefore, the first action which must be done is to find groups of similar cases, considering the values taken for the different variables.

In order to obtain such groups a GCS net is used. This kind of net is also used by other authors in the CBR retrieve phase[7]. The information provided by the net includes: a) how many groups are created, b) which cases take part in each group, c) which is the prototype case representing all the cases in the group and d) what is the distance between each case within the group and the prototype case.

For each identified set, a TSK rule is obtained [17]. These rules all together constitute the initial fuzzy inference system. The antecedent of each rule is a combination of variables which describe each case initial belief. They can be represented by a gauss function. In order to obtain the rule consequents, the least square method is used [11].

This initial fuzzy inference system will be used as previous knowledge in the ANFIS model, which will adjust the parameters of both antecedents and consequents, using the hybrid learning method explained in section 2.2. The refinement of these parameters is done using as input patterns the most similar cases retrieved in the previous phase. The result is a new fuzzy inference system, which will estimate the resolution of a new problem.

## 2.1 Retrieve Phase: GCS Network

The type of GCS used in this work is characterized by a two-dimensional space, where the units (cells) are connected and organised in triangles. Each cell in the network is associated with a weight vector,  $w$ , which has the same dimension as the input data. At the beginning of the learning process, the weight vector of each cell is initialised with random values. The basic learning process in a GCS network consists of topology modification and weight vector adaptations [10]. This vector is the prototype case of each cell of the network.

For each training case, the network performs a so-called *learning cycle*, which may result in topology modification and weight vector adaptation. In the first step of each learning cycle, the cell  $c$ , with the smallest distance between its weight vector,  $w_c$ , and the actual input vector,  $x$ , is chosen as the winner cell or best-match cell (see equation (1)).

$$c : \|x - w_c\| \leq \|x - w_i\|; \forall i \in O \quad (1)$$

The second step consists of the adaptation of the weight vectors of the winning cell and their neighbouring cells; see equations (2) and (3). The terms  $\varepsilon_c$  and  $\varepsilon_n$  represent the learning rates for the winner and its neighbours respectively. Both learning rates are constant during learning, and  $\varepsilon_c, \varepsilon_n \in [0, 1]$ .

$$w_c(t+1) = w_c(t) + \varepsilon_c(x - w_c) \quad (2)$$

$$w_n(t+1) = w_n(t) + \varepsilon_n(x - w_n); \forall n \in N_c \quad (3)$$

In the third step of a learning cycle, each cell is assigned a *signal counter*,  $\tau$ , that reflects how often a cell has been chosen as winner (see equations (4) and (5)).

$$\tau_c(t+1) = \tau_c(t) + 1 \quad (4)$$

$$\tau_i(t+1) = \tau_i(t) - \alpha\tau_i(t); i \neq c \quad (5)$$

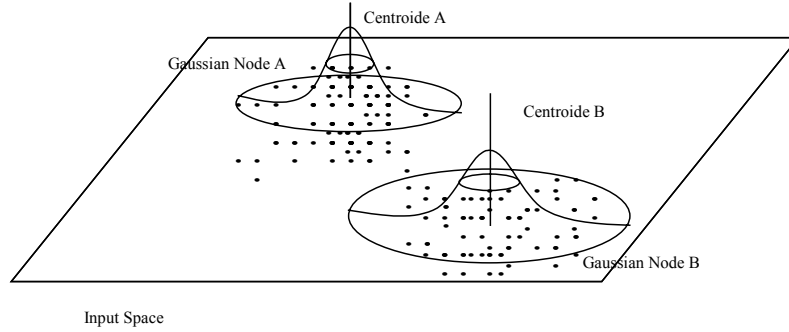
The parameter  $\alpha$  reflects a constant *rate of counter reduction* for the rest of the cells at the current learning cycle. Growing cell structures also modify the overall network structure by inserting new cells into those regions that represent large portions of the input data. The frequency of insertion update is controlled by the parameter  $\lambda$ , which is associated with the number of learning cycles between two cell insertions (see equations (6), (7) and (8)).

$$h_i = \frac{\tau_i}{\sum_j \tau_j}; \forall i, j \in O \quad (6)$$

$$q : h_q \geq h_i; \forall i \in O \quad (7)$$

$$r : \|w_r - w_q\| \geq \|w_p - w_q\|; \forall p \in N_q \quad (8)$$

The GCS network indicates the prototype case of each node, its topology and calculates the scale parameters  $\sigma_j$  of each node [7]. This parameter measures the width in the gauss membership function. It can be seen in the Figure 1. Higher values of  $\sigma$  provide an area more extended of the node dominates in the environment of the centroide.



**Fig. 1.** The point mean cases in the input environment

To calculate the node  $j$ , the prototype cases of its neighbour nodes are selected; then the average of the square-distance between them is calculated [18][19], that is:

$$\sigma_{jk}^2 = \frac{1}{N} \sum_{l=1}^N \|c_l - c_j\|^2 = \frac{1}{N} \sum_{l=1}^N (c_{lk} - c_{jk})^2 \quad (9)$$

where  $K$  is particular variable in cases.

This information is utilized for making the initial fuzzy inference system, which is utilized by the ANFIS model. This fuzzy inference system has a set of TSK rules; each node provides a fuzzy rule. The rules have the form:

$$R_j: \text{if } x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \text{ and } \dots \text{ and } x_M \text{ is } A_{Mj}, \text{ then } y = g_j(x_1, x_2, \dots, x_M) \quad (10)$$

where  $g_j(\cdot)$  is a polynomic function in  $x_i$ .

Each attribute is represented by a gauss function (equation (11)), which takes part of the antecedent of the rule.

$$A_{ij}(x_i) = \exp\left(-\frac{(c_k - x_i)^2}{2\sigma_k^2}\right) \quad (11)$$

where  $c$  is a prototype case and  $\sigma$  the distance.

The next step consists of obtaining the consequents of each TSK rule. The method utilized is least-square [11]. This initial fuzzy inference system is adapted with the ANFIS model.

In this phase the most similar cases to the new problem are retrieved. The problem is determined by a set of variables with particular values, which are used as inputs of the GCS net. Next, the searching for the node to which the new problem belongs is

started, that is, the winner node must be found. This node is obtained by calculating the Euclidean distance between the new problem and every prototype case of each case. The node with the fewer distance will be the winner. All the cases associated to the winner node will be considered the most similar ones, and will be utilised in the following phase.

## 2.2 Reuse Phase: ANFIS Model

One of the first hybrid neuro-fuzzy systems for function approximation was Jang's ANFIS model [11]. ANFIS adjusts only the membership functions of the antecedent and the consequent parameters.

Because ANFIS uses only differentiable functions, it is easy to apply standard learning procedures from neural network theory. For ANFIS a mixture of backpropagation (gradient descent) and least squares estimation (LSE) is used. Backpropagation is used to learn the antecedent parameters, i.e. the membership functions, and LSE is used to determine the coefficients of the linear combinations in the rule's consequents.

A step in the learning procedure has two parts, which are shown in Table 1. In the first part the input patterns are propagated, and the optimal consequent parameters are estimated by an iterative least mean squares procedure, while the antecedent parameters are assumed to be fixed for the current cycle through the training set. In the second part the patterns are propagated again, and in this, epoch backpropagation is used to modify the antecedent parameters, while the consequent parameters remain fixed.

**Table 1.** Two passes in the hybrid learning procedure for ANFIS

	<b>Forward pass</b>	<b>Backward pass</b>
<b>Antecedent parameters</b>	Fixed	Gradient descent
<b>Consequent parameters</b>	Least-square estimator	Fixed
<b>Signals</b>	Node outputs	Error signals

In this phase, the cases retrieved in the previous one are used, that is, the inference system provided by the GCS net. While the result is a fuzzy inference system adapted for solving a particular problem. With this system a result, which will be converted in the desire the CBR-BDI must achieve, is obtained.

Therefore, the next step the CBR-BDI agent must accomplish will be the planing of what actions to do to achieve this desire. The actions carried out for in the retrieved cases are obtained and the following process is done. An acyclical and directed structure whose first vertex is the new problem and the last one is the desire to achieve is created. The construction of this structure is done taking each one of the actions made in the retrieved cases and adapting their parameter values. Once the structure is built, Dijkstra algorithm [15] is used to determine the shortest path, taking as origin the new problem. The path determines the actions which must be done and a new intention is built. This intention reflects the solution to the posed problem.

Summarising, in this phase a sequence of actions starting from a new problem and the result which must be achieved, is proposed; that is, a new case.

### 2.3 Revision and Retain Phases

In this phase the solution obtained in the previous phase is evaluated. The revision can be carried out using Expert's Knowledge (rules) or simulation techniques [5], fuzzy inference system [7] or Belief-Revision techniques [12].

The new case (a problem, a solution and a result) is stored in the cases base. In this phase the produced error between estimated and real result is calculated. If the error is higher than a limit  $\beta$  the GCS network is rebuilt, because this means that a new input space which has not been considered before is being visited. Therefore the network must modify its topology and adapt the weights vector, even using the new stored cases.

## 3 Study Case: Shellfish Exploitation Plans Automation

In Galicia, there are a deep interest in ordering the fishing sector. Due to that, a set of rules for organising the marine resources exploitation were developed. According to those rules, the galician government is in charge of controlling and regulating both the extracting activity of marine resources in Galicia seashore and their commercial transactions which take place in the specific locations devoted to that aim.

In order to practice the extraction of marine resources it is necessary to present to the administration some documents named *shellfish exploitation plans*, which are elaborated for entities interested in exploiting Galician marine resources. Each extracting entity must prepare an exploitation plan for every resource it wants to obtain. The aim of these plans is to achieve the greatest continuous economical profit from marine resources by means of an appropriate planning for the extracting activity.

In Table 2, sections and topics included in an exploitation plan are shown.

**Table 2.** Exploitation plan summary

<b>General data</b>	<b>Goals</b>	<b>Evaluation</b>	<b>Pursuit</b>
Shellfish men's number	Production	Methods	Daily effort
Boats number	Economical	Conclusions	Daily production
Exploitation areas			

<b>Extraction plan</b>	<b>Improvement actions</b>	<b>Financial plan</b>
Probable schedule	Description	Incomes
Dates	Costs	Expenses
Limit (Kg/day)		Investment
Fishing traps		Capitalization
Points of control		
Selling ways		
Surveillance		

The shellfish exploitation plan presented by a extracting entity has values for a complete year and must be approved by the Fishing Authority. Since the moment of its approbation, the shellfish exploitation plan will regulate along the year the capture of the resources that it contains.

Notice that plans elaborated by the extracting entities incorporate general data, like shellfish men's number, selling ways, etc, but they also must include forecast about productive and economical goals, which have to be based on characteristics of extracting entities, environment and conditions of shellfish ecosystems.

Since the information managed by these plans is extensive enough, an automated system that gets, stores and analyses data about marine resources becomes essential. Appropriate tools for collecting data allow the acquisition of useful knowledge for managing marine resources by means of rational criteria. These tools provide help not only to entities in charge of exploitation, assisting them in the process of elaborating new plans, but also to administration, designing better fisheries policies that prevent overexploitation generated by an excessive fishing effort.

Nevertheless, the interest of the current shellfish management system goes beyond a simple statistical study, and pretends to adapt the exploitation plans to the characteristics of each school of shellfish and the necessities of Shellfish men. In this way, each year different management models can be applied and different technical solutions can be proposed in order to help in decisions making.

To obtain this objective, a CBR-BDI agent, like the one described in this work will be used. This agent will generate automatically the shellfish exploitation plan, allowing a reasonable and sustainable exploitation which is a desirable objective in all shellfish sector.

Next, an example of the application of a CBR-BDI agent for automatically generate shellfish exploitation plans for all Galician extracting entities devoted to clam is presented. All the information available in the shellfish exploitation plans belonging this resource and pertaining to previous years is used. This will allow to adjust the productive and economical goals. All the same, as a way of simplifying the example, only an estimation of productive goals (resource kilograms) will be done. If the rest of data were to be obtained, the same process must be followed.

In first place, it is necessary to define the CBR-BDI agent in the terms of the 4-tupla  $\{E, CB, GAL, EK\}$ , where E are the variables which describe the environment, CB is the case base (in terms of beliefs, desires and intentions), GAL is the general actions library and EK identifies the expert's knowledge [4].

The variables which describe the example environment are the typical ones for extracting entities (number of working days, number of shellfish men x number of days), for schools of shellfish like: environmental data (temperature, salty degree in water, PH, oxygen rate, transmittance and fluorescence), the size of the different school of shellfish (area and perimeter) and limits of allowed captures, besides the variable that must be forecast in the exploitation plan. This study is centred in the estimation of the number of marine resource kilograms that are to be recollected.

For building up the CBR-BDI system, a tool, called GABDI (abbreviation in Spanish of *BDI agents Generator*), is available. This tool facilitates the addition of information to the cases base.

In order to define the variables which define the environment, GABDI tool provides a form where the name and the rank of values the variable may take, can be introduced. Next step is to describe the actions which can be made over environment. Table 3 shows the actions which can be done to improve the recollection of a particular marine resource.

**Table 3.** Actions which can be done to improve the production of a marine resource

Action name	Input parameters	Output parameters
Transport	Oxygen	Oxygen, Kilograms
Move	Kilograms	Kilograms
Plantation	Kilograms	Kilograms
Seaweed removal	Oxygen	Oxygen, Kilograms
Plough	Area, Perimeter	Area, Perimeter, Kilograms

Next, the fulfilled intentions must be defined; that is, the actions done previously to solve old problems. Once identified the environmental variables, the actions which can be done and the actions done in the past, it is necessary to create the cases base. In order to achieve this objective, the states (beliefs) and the plan of done actions (intentions) must be indicated.

Now, in the cases base is stored all information available for the agent, so it can apply its reasoning model using the techniques described in section 2. The information managed by the CBR-BDI agent is saved in *csv* format.

### 3.1 Results Obtained

The correct functioning of the model has been proved experimentally, by means of a set of proofs. First proofs were made over 6554 stored cases, particularly, the ones representing the captures of the years 2000 and 2001 of clam. The forecasting was done over 192 situations belonging to year 2002.

The results corresponding to two systems were compared. One system is the initial fuzzy inference system obtained by GCS network (from here on GCS) and the other, the system proposed by the present work (from here on CBR-BDI agent). It must be highlighted that 82,8% of the times, the forecast of the CBR-BDI agent is better than the one provided by GCS system.

In first place, it has been proved if the samples that must be analysed follow a normal distribution. The proofs of normality applied were Z skewness and Z kurtosis. Since the result from these tests was negative, that is, they do not obey the normality hypothesis, a set of non parametrical proofs were applied. These proofs try to determine if a system is better than other analysing the data globally. The statistical techniques used in this latest case were the Sign Test and the rank sum of Wilcoxon for coincident pairs.

The Sign Test is designed specifically for proving hypothesis referents to median of a continuous population. Like the mean, the median is a measure of the centre or distribution position, because of that the Sign Test is also known as proof of position.

Since in the retrieve phase the GCS system provide a initial fuzzy inference system, it can be used to make the forecasting without the necessity of adapting its parameters by means of the ANFIS model. In order to prove that better results are obtained if the parameters are adapted, the two systems were compared: the provided by GCS and the CBR-BDI agent.

Table 4 shows the results after the application of The Sign Test over the two systems previously introduced. Results indicate that the error using the GCS system is bigger than the produced by CBR-BDI agent, with a confidence level of about 95%. Since  $p$  value is very small, it can be assured that in the case of the two extremes proof the null hypothesis is also rejected.



**Table 4.** The Sign Test between the initial fuzzy inference system and the CBR-BDI agent

Test Alternative Hypothesis	Sign Test		
	GCS Error	$\geq$	Agent Error
Difference between pairs	n		
Positive	183		
Negative	9		
Zero	0		
Median difference	3088,466		
95.8% CI	2387,762	to $+\infty$	(exact)
Sign Statistics	183		
1-extremo p	$<0.0001$		(exact)

From the results obtained with the statistical sign test, it can be concluded that the CBR-BDI agent provides better results than the other two systems.

A powerful non parameter technique must compare the whole probability distributions not only the median. This test, which is called the rank sum of Wilcoxon, proofs the null hypothesis, that is, the probability distributions associated to the two populations are equivalent against the alternative hypothesis the probability distribution of a population is moved right (or left) with respect to the other.

In Table 5, the results of the rank sum of Wilcoxon test for two populations, as well as the error produced by GCS system and for the CBR-BDI system, are shown.

**Table 5.** Rank sum of Wilcoxon among GCS system and CBR-BDI agent

Test Alternative hypothesis	Rank sum of Wilcoxon test for coincident pairs		
	GCS Error	$\geq$	Agent Error
Pairs difference	n	Rank sum	Rank median
Positive	183	18377,0	100,42
Negative	9	151,0	16,78
Zero	0		
Median difference	7285,193		
95.0% CI	5267,973	to $+\infty$	(normal aproximation)
Wilcoxon's statistic	18377		
1-extreme p	$<0.0001$		(normal aproximation)

Before doing the tests showed in Table 5, it was made the rank sum of Wilcoxon test for coincident pairs in the two extremes, in which the null hypothesis was rejected. In order to further refine this proof, it was analysed as alternative hypothesis, that the GCS system error were bigger or equal to CBR-BDI agent error, rejecting also the null hypothesis. The final conclusion was that population number 1 is moved to right of population number 2.

The rank sum of Wilcoxon test for coincident pairs reinforces the results obtained, in the sense that the CBR-BDI system provides better results than the other system. So that, it can be concluded that it is necessary to adapt the initial fuzzy inference system obtained by GCS network and that the ANFIS model is the best one to be utilised for that objective.

## 4 Conclusions

This job is part of the objectives of the action of research “Ampliación de sistemas de información geográfica orientado a la gestión de los recursos específicos a los demás recursos marisqueros de Galicia” approved by Xunta of Galicia with code: PGIDTCIMA 02/3. This research is carried out between CIMA (*Centro de Investigaci3n Mariñas*), University of Coruña and University of Vigo.

In this paper, it is showed how a CBR-BDI agent is able to learn and to give solutions to a particular problem. It utilizes a fuzzy inference system, which is adapted to the problem to solve.

The acceptance of this CBR-BDI agent, by the extracting entities, has been excellent. At this moment, it is being tested in different extracting entities. The satisfaction exhibited by the entities devoted to extract marine resources allows to foresee that the system will be completely implanted next year.

At present the elaboration of the shellfish exploitation plans is manual, slow and little reliable. In this action of research it is proposed a simple input data, besides the automatic preparation of the plans with that data.

As a final conclusion, it can be said that the model described in this research paper is capable of adding the partial knowledge provided by each incorporated technology, creating a global knowledge system, based on the case base reasoning method.

## 5 Acknowledgements

This research is supported by the Spanish national project TIC2002-04516-C03-01.

## 6 References

1. Aamodt A. y Plaza E.: Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7. No 1, March. (1994)
2. Bratman M.E., Israel D., y Pollack M.E.: Plans and resource-bounded practical reasoning. *Computational Intelligence* 1988, 4, pages 349-355. (1988)
3. Corchado J. M., Laza R., Borrajo L., Yañez J. C. y Valiño M.: Increasing the Autonomy of Deliberative Agents with a Case-Based Reasoning System. *International Journal of Computational Intelligence and Applications*. ISSN: 1469-0268. (2003)
4. Corchado J.M. y Laza R.: Construction of BDI Agents from CBR systems. 1<sup>st</sup> German Workshop on Experience Management. *Lecture Notes in Informatics Berlin*, 7-8 Marzo. (2002)

5. Corchado J.M., Aiken J. y Rees N.: Artificial Intelligence Models for Oceanographic Forecasting. Plymouth Marine Laboratory. ISBN-0-9519618-4-5. (2000)
6. Corchado J.M., Laza R., Borrajo L., Yáñez J.C., de Luis A. y Glez-Bedia M.: Agent-based Web Engineering. ICWE 2003, Third International Conference on Web Engineering, Oviedo, Asturias, Spain. July 14 -18, (2003)
7. Fdez-Riverola F, Corchado J. M. y Torres J.: An Automated Hybrid CBR System for Forecasting. Advances in Case-Based Reasoning. Craw S. and Preece A. (eds.), Springer. ISBN: 3-540-44109-3. (2002)
8. Georgeff M.P. y Lansky A.L.: Procedural knowledge. In Proceedings of the IEEE Special Issue on Knowledge Representation, volume 74. pages 1383-1398. (1986)
9. Jennings N.R.: On Being Responsible. In Y. Demazeau and E. Werner, editors, Decentralized A.I. 3. North Holland, Amsterdam, The Netherlands (1992)
10. Köhle, M., y Merkl, D.: Visualizing similarities in high dimensional input spaces with a growing and splitting neural network. En Proceedings of International Conference of Artificial Neural Networks, ICANN-96, (pp. 581-586), Bochum, Germany. (1996)
11. Nauck D., Klawonn F. y Kruse R.: Foundations of neuro-fuzzy systems. Publicación Chichester: John Wiley & sons, cop. 1997
12. Pavón R., Laza R., Gómez A. y Corchado J.M.: Automating the Revision phase of a Case-Based Reasoning system using Belief Revision. Soft Computing and Intelligent Systems for Industry. ICSC-NAISO Academic Press. Paisley, Scotland, United Kingdom. (2001)
13. Rao A.S. y Georgeff M.P.: BDI Agents: From Theory to Practice. First International Conference on Multi-Agent Systems (ICMAS-95). San Francisco, USA, June (1995)
14. Rao A.S. y Georgeff M.P.: 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. (1991)
15. Schulz F., Wagner D. y Weihe K.: Dijkstra's Algorithm On-line: An Empirical Case Study from Public Railroad Transport. Algorithm Engineering. pages 110-123. (1999)
16. Shoham Y.: Agent-Oriented programming. Artificial Intelligence. 1993, 60(1): pages 51-92. (1993)
17. Takagi, T., y Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics, 15:116-132. (1985)
18. Warwick, K.: An overview of neural networks in control applications. Neural Networks for Robotic Control, M. Zalzala, Prentice-Hall. (1995)
19. Wasserman, P. D.: Advanced Methods in Neural Computing. Van Nostrand Reinhold. (1993)
20. Wooldridge M.: Intelligent Agents. Multiagent Systems. A modern approach to Distributed Artificial Intelligence. Edited by Gerhard Weiss, 1999. Pages 27-77. (1999)