# Kernel-SIM preprocessing in a variational calculus based planning strategy

E. S. Corchado[1], J. M. Corchado[2], and C. Fyfe[3]
[1]Dept. de Ingeniería Civil, University of Burgos, Esc. Politécnica Superior, Edificio C, C/ Francisco de Vitoria, Burgos, Spain
[2]Departamento de Informática y Automática, University of Salamanca, Plaza de la Merced s/n, 37008, Salamanca, Spain
[3]Computing and Information System Dept. University of Paisley, High Street, Paisley,U.K.

**Abstract:** This paper present a novel technique for constructing autonomous agents using a case-based reasoning system. The reasoning model incorporates a K-SIM algorithm for the identification of the appropriate information to construct the agent plan, in execution time, using variational calculus. The proposed methodology has been used to create several agent based systems. The paper describes the methodology and outlines a case study.

## 1 Introduction

Interesting case-based reasoning (CBR) models have been presented, over the last few years, for constructing deliberative agents. This paper presents a novel architecture that has been developed for constructing deliberative agents which can generate their plans using the framework of a case-based reasoning (CBR) system, that incorporate a Kernel Maximum Likelihood Hebbian Learning Scale Invariant Map (K-SIM) [1] for the identification of the information that later may be used to create a plan using variational calculus [2]. The robust analytical notation introduced in [3] is used to present the proposal.

Agents must be able to respond to events that take place in their environment, take the initiative according to their goals, interact with other agents (even human) and use past experiences to achieve current goals. The deliberative agent with BDI (Belief, Desire and Intention) architecture, uses the three attitudes in order to make decisions on what to do and how to achieve it [4, 5, 6]: their beliefs represent their information state (what the agents know about themselves and their environment); their desires are their motivation state (what they are trying to achieve); and the intentions represent the agents' deliberative state. These mental attitudes determine the deliberative agent's behaviour and are critical if a proper performance is to be produced when information about a problem is scarce [7, 8]. BDI architecture has the advantage that it is intuitive, it is relatively easy to recognize the process of decision-making and how to perform it. Moreover, it is easy to understand the notions of

---

This work has been partially supported by the Spanish MCyT under project TIC2001-4936-E

belief, desires and intentions. For several year we have been working on a mechanism to facilitate the effective implementation of autonomous agents using CBR systems [9]. A robust analytical formalism for the definition of computationally efficient agents, which solves the first of the previously mentioned problems has been identifies and presented in [2, 3]. This paper presents an improved version of our approach in which a Kernel Maximum Likelihood Hebbian Learning Scale Invariant Map (K-SIM) is used for the identification of the cases required to construct the agent plan.

The analytical notation is reviewed and it is outlined how a CBR system is used to operate the mental attitudes of a deliberative BDI agent. Te K-SIM technique is introduced and an agent based system for the e-tourism domain that uses the methodology presented in the paper.

## 2 Constructing agents with case-based reasoning systems

A novel methodology for constructing deliberative agent has been presented in [2, 3], this section outlines it and explains why it has to be improved. This section identifies the relationships that can be established between deliberative agents and CBR systems, and shows how an agent can reason with the help of a case-based reasoning system. Case-based reasoning is used to solve new problems by adapting solutions that were used to solve similar previous problems [9, 10]. The operation of a CBR system involves the adaptation of old solutions to match new experiences, using past cases to explain new situations, using previous experience to formulate new solutions, or reasoning from precedents to interpret a similar situation.

The reasoning cycle of a typical CBR system includes four steps that are cyclically carried out in a sequenced way: retrieve, reuse, revise, and retain [9]. During the retrieval phase, those cases that are most similar to the problem case are recovered from the case-base. The recovered cases are adapted to generate a possible solution during the reuse stage. The solution is then reviewed and, if appropriate, a new case is created and stored during the retention stage, within the memory. Therefore CBR systems update their case-bases and consequently evolve with their environment. Each of the reasoning steps of a CBR system can be automated, which implies that the whole reasoning process could be automated to a certain extent [9,10]. This assumption has led us to the hypothesis that agents implemented using CBR systems could be able to reason autonomously and therefore to adapt themselves to environmental changes. Agents may then use the reasoning cycle of CBR systems to generate their plans.

The proposed model has identified a direct mapping between the agents and the reasoning model has been established, in such a way that the mapping between the agents and the reasoning model should allow a direct implementation of the agent and the final agents should be capable of learning and adapting to environmental changes.

The notation and the relationship between the components that characterise a BDI agent is defined as following. Let $\Theta$ be the set that describes the agent environment. If $T(\Theta)$ is the set of attributes $\{\tau 1, \tau 2, \ldots, \tau n\}$ in which the world's beliefs are expressed, then we define a belief on $\Theta$, that is denoted "e", as an m-tuple of some attributes of T(Θ) denoted by e = (τ1, τ2,…,τm) with m≤ n.

We call set of beliefs on Θ and denote ζ(Θ) to the set:

ζ(Θ) = { e=(τ1, τ2,…,τj ) / where j =(1,2,…, m ≤ n )}

The operator "Λ of accessibility" between m beliefs ($e_1,e_2,e_3,\ldots,e_m$), where we denote: $\Lambda(e_1, e_2, e_3,\ldots,e_m) = (e_1 \wedge e_2 \wedge \ldots \wedge e_m)$ indicates that exists compatibility among the set of beliefs ($e_1, e_2, e_3,\ldots, e_m$). If any of the belief ($e_1, e_2, e_3,\ldots, e_m$) is not accessible, or if there exists a contradiction, it will be denoted by:$\Lambda(e_1,e_2, e_3,\ldots,e_m) = \emptyset$. Moreover, an intention i on Θ is defined as an s-tuple of compatible beliefs, i = ($e_1, e_2,\ldots,e_s$) with s∈ IN and $\Lambda( e_i, e_j )\neq 0$. Then, we call set of intentions on Θ and denote I(Θ)={($e_1, e_2, \ldots, e_k$) where k∈IN}. Now a set of parameters will be associated to the space I(Θ) that characterises the elements of that set. The set of necessary and sufficient variables to describe the system may be obtained experimentally. We call canonical variables of a set I(Θ) any set of linearly independent parameters ℵ =($A_1, A_2,\ldots, A_v$) that characterise the elements $i \in$ I(Θ).

In the same way, a desire d on Θ is defined as a mapping between

$$d : I(\Theta) \longrightarrow \Omega(\aleph)$$
$$i = (e_1 \wedge \ldots \wedge e_r) \rightarrow F(A_1, A_2,\ldots, A_v)$$

where Ω ( ℵ ) is the set of mappings on ℵ .

A desire d may be achieved constructing an intention i using some of the available beliefs, whose output could be evaluated in terms of the desired goals. We denote D(Θ) the set of desires on Θ: D(Θ)={d: I(Θ)→ Ω ( ℵ ) / with I(Θ) set of intentions and Ω ( ℵ ) set of mappings on ℵ }

Now, after presenting our definition of the agent's beliefs, desires and intentions, it is reviewed the proposed analytical formalism for the CBR system. The necessary notation to characterise a CBR system is introduced as follows. Let us consider a problem P, for which it is desired to obtain the solution S(P). The goal of a case-based reasoning system is to associate a solution S(P) to a new problem P, by reusing the solution S(P′) of a memorised problem P′. P is denoted as P=($S_i$,{ $\theta_j$ }, $S_f$ ) with $S_i$=initial state, $S_f$=final state and j=(1,…,m). S(P) is defined as S(P)= { $S_1, \theta_l, S_2, \theta_2,\ldots, \theta_n, S_{n+1}$ }={$S_k, \theta_h$} where k=(1,..,n+1) and h=(1,..,n ≤ m) , $S_1=S_i$ and $S_{n+1}= S_f$.

The state $S_k$ and the operator $\theta_j$ are defined as:

$$S_k = \begin{pmatrix} \{O_r\}_{r=1,\ldots,p} \\ \{R_s\}_{s=1,\ldots,q} \end{pmatrix} \qquad \theta_j : S_k=\begin{pmatrix} \{O_r\} \\ \{R_s\} \end{pmatrix} \longrightarrow \theta_j(S_k)=\begin{pmatrix} \{O'_r\} \\ \{R'_s\} \end{pmatrix}$$

where $\{O_r\}_{r=1,\ldots,p}$ and $\{R_s\}_{s=1,\ldots,q}$ are coordinates in which a state $S_k$ is expressed.

The coordinates type $\{O_r\}_{r=1,\ldots,p}$ are introduced to express the objectives achieved. The coordinates type $\{R_s\}_{s=1,\ldots,q}$ are introduced to express the resources used. Through these definitions, the parameter effectiveness, ℑ, between two states S and S' can be defined, as a vector ℑ (S, S') = ($\Im_x$, $\Im_y$) which takes the form

$$\Im_x = \frac{O_r(S') - O_r(S)}{O_r \max} \qquad \Im_y = \frac{R_s(S) - R_s(S')}{R_s \max}$$

The definition implies that ( $0\leq\Im_x\leq1$ ) and ( $0\leq\Im_y\leq1$ ). In particular, if $S=S_i$ and $S'=S_f$ , it is denoted ℑ ($S_i, S_f$)= ℑ [S(P)] and we call it "effectiveness of a solution". In

This work has been partially supported by the Spanish MCyT under project TIC2001-4936-E

order to evaluate the rate of objectives achieved and resources used, between S and S', it is necessary to normalise every component of $\{O_r\}_{r=(1,\ldots,p)}$, $\{R_s\}_{s=(1,\ldots,q)}$ . Then the expressions that have been defined to sum different objectives are:

If $\{O_r(S)\}= ( O_1, O_2,..., O_P )$ and $\{O_r(S')\}= ( O'_1, O'_2,..., O'_P )$

$$\Im_x = \frac{\sqrt{\left(\frac{O'_1-O_1}{\max O_1}\right)^2+\left(\frac{O'_2-O_2}{\max O_2}\right)^2+\ldots+\left(\frac{O'_p-O_p}{\max O_p}\right)^2}}{\sqrt{\left(\frac{\max O_1}{\max O_1}\right)^2+\left(\frac{\max O_2}{\max O_2}\right)^2+\ldots+\left(\frac{\max O_p}{\max O_p}\right)^2}} = \frac{\sqrt{\left(\frac{O'_1-O_1}{\max O_1}\right)^2+\left(\frac{O'_2-O_2}{\max O_2}\right)^2+\ldots+\left(\frac{O'_p-O_p}{\max O_p}\right)^2}}{\sqrt{p}}$$

As $\{R_s(S)\}= ( R_1, R_2,..., R_q )$ and $\{R_s(S')\}= ( R'_1, R'_2,..., R'_q )$ it is defined

$$\Im_y = \frac{\sqrt{\left(\frac{R_1-R'_1}{\max R_1}\right)^2+\left(\frac{R_2-R'_2}{\max R_2}\right)^2+\ldots+\left(\frac{R_q-R'_q}{\max R_q}\right)^2}}{\sqrt{\left(\frac{\max R_1}{\max R_1}\right)^2+\left(\frac{\max R_2}{\max R_2}\right)^2+\ldots+\left(\frac{\max R_q}{\max R_q}\right)^2}} = \frac{\sqrt{\left(\frac{R_1-R'_1}{\max R_1}\right)^2+\left(\frac{R_2-R'_2}{\max R_2}\right)^2+\ldots+\left(\frac{R_q-R'_q}{\max R_q}\right)^2}}{\sqrt{q}}$$

A new parameter is also introduced - efficiency - that measures how many resources are needed to achieve an objective. Given a target problem P, and a solution S(P), we define $\varsigma[S(P)]= \Im_x / \Im_y$ , as the efficiency of the solution S(P). The definition implies that $\varsigma(S,S') \in (0,\infty)$. The meaning of this new parameter is explained later. In this domain, a case C is a 3-tuple *{P, S(P), ℑ[S(P)]}* where P is a problem description, S(P) the solution of P and ℑ[S(P)] the effectiveness parameter of the solution, and a CBR´s case base CB, denoted as: *CB={$C_k$ / k=(1,...,q) and* $q \in IR$} that is a finite set of cases memorised by the system. Finally the relationship between CBR systems and BDI agents can be established, associating the beliefs, desires and intentions with cases. Using this relationship we can implement agents (conceptual level) using CBR systems (implementation level). So once the beliefs, desires and intentions of an agent are identified, they can be mapped onto a CBR system. First, a mapping is introduced that associates an index to a given case $C_k$.

$$\begin{aligned} &idx: CB \rightarrow I(CB) \\ &C \rightarrow idx(C)= idx\{P, S(P), \Im[S(P)]\}= \{ idx (S_i), idx (S_f) \}= \\ &= \{ [S_i =(O_1,a_1), (O_2,a_2),...,(O_p, a_p), (R_1,b_1), (R_2,b_2),...., (R_q, b_q)], \\ &\quad [S_f=(O'_1,c_1),(O'_2,c_2),..,(O'_p, c_p),(R'_1,d_1),(R'_2,d_2),..,(R'_q,d_q)] \} \\ &\text{with } Oj, R_k \in T(CB), a_i, b_j, c_k, d_l \in IR \text{ and } p, q \in IN \end{aligned}$$

where the set I(CB) is the set of indices of a case base CB that is represented by frames composed of conjunction of attributes of T(CB) and values of the domain. The abstraction realized through the indexing process allows the introduction of an order relation R in the CB that can be used to compare cases. Indices are organized in the form of a Subsumption Hierarchy.

$$(CB, R)=\{ [C_k / k=(1,..,q) \text{ and } q \in IN ], R\}=\{(C_1,..,C_q)/ idx(C_1) \subseteq .. \subseteq idx(C_q)\}$$

Let us say that two cases C and C' $\in$ CB fulfill the relation

$$idx(C) \subseteq idx(C') \quad \text{if} \quad \begin{array}{l} idx(S_I) \subseteq idx(S'_I) \\ idx(S_F) \supseteq idx(S'_F) \end{array}$$

And it is expressed in terms of their components,

$$idx(S_I) \subseteq idx(S'_I) \rightarrow \begin{cases} O_r(S_I) \geq O_r(S'_I) & \forall r = 1, \dots p \\ R_s(S_I) \leq R_s(S'_I) & \forall s = 1, \dots q \end{cases}$$

$$idx(S_F) \supseteq idx(S'_F) \rightarrow \begin{cases} O_r(S_F) \leq O_r(S'_F) & \forall r = 1, \dots p \\ R_s(S_F) \geq R_s(S'_F) & \forall s = 1, \dots q \end{cases}$$

Let us say that S(P') is a possible CBR solution of the target P,

$\forall$ C'= ( P' , S(P'), $\Im$[S(P')] ) / idx(C') $\supseteq$ P

Given a canonical coordinate system $\aleph = (A_1, A_2, \dots, A_v)$ on I(Θ), the set may be reordered, differentiating between:

$\{F_m\} = \{A_{j \text{ with } j \leq v} / A_j \text{ growing}\}$ and $\{G_n\} = \{A_{k \text{ with } k \leq v} / A_k \text{ decreasing}\}$ so,

$\aleph = \{F_m\} \cup \{G_n\}$ and m+n=v

Then, giving an $i \in I(\Theta)$, a functional dependency relationship may be obtained in terms of the attributes $i = i\ [e_1(\tau_1, \tau_2, \dots, \tau_j), e_2(\tau_1, \tau_2, \dots, \tau_k), \dots, e_s(\tau_1, \tau_2, \dots, \tau_q)] =$

$= i(\tau_1, \tau_2, \dots, \tau_n)$ and in terms of its canonical or state variables:

$i = i\ (A_1, A_2, \dots, A_v) = i\ (F_1, F_2, \dots, F_m, G_1, G_2, \dots, G_n)$ which determines a functional relationship of the type $A_j = A_j(\tau_1, \tau_2, \dots, \tau_n)$.

Now the fundamental relationship between the BDI agents and the CBR systems can be introduced. We define "state $\varsigma$ of an intentional process" and we denote as $\varsigma = \{e_1 \wedge e_2 \wedge \dots \wedge e_{s-1} \wedge e_s\}$ to describe any of the situations intermediate to the solution $i = \{e_1 \wedge e_2 \wedge \dots \wedge e_r$, with $r \leq s\}$ that admits a representation over $\aleph$ . Moreover, the solution S(P) for a given problem $P = (S_I, \{\theta_j\}, S_F)$ can be seen as a sequence of states $S_k = (\{O_r\}_{r=1,\dots,p}, \{R_s\}_{s=1,\dots,q})$ interrelated by operators $\{\theta_h\}$.

Given a BDI agent over Θ with a canonical system, $\aleph = (A_1, A_2, \dots, A_v)$ in the set I(Θ) that may be reordered as $\aleph = (F_1, F_2, \dots, F_m, G_1, G_2, \dots, G_n)$, we establish the relationship between the set of parameters:

$$\{F_m\} \longleftrightarrow \{O_r\} \qquad \{G_n\} \longleftrightarrow \{R_s\}$$

The identification criteria may be established among

- the intentional states, $\varsigma_i \in I(\Theta)$, and the CBR states, $S_k \in T(BC)$.
- and a relationship may be established among the agents desires I(Θ) and the effectiveness operator $\Im$[S(P)] of the CBR system.

Then the mathematical formalisation proposed can be used as a common language between agents and CBR system and solves the integration problem. The relationship presented here shows how deliberative agents with a BDI architecture may use the reasoning cycle of a CBR system to generate solutions S(P).

The relationship, presented here, shows how deliberative agents with a BDI architecture may use the reasoning cycle of a CBR system to generate solutions S(P). When the agent needs to solve a problem, it uses its beliefs, desires and intentions to obtain a solution. Previous desires, beliefs and intentions are stored taking the form of cases and are retrieved depending on the current desire. Cases are then adapted to generate a proposed solution, which is the agent action plan.

---

This work has been partially supported by the Spanish MCyT under project TIC2001-4936-E

## 3 Planning in execution time

The proposed agents have the ability to plan their actions, to learn and to evolve with the environment, since they use the reasoning process provided by the CBR systems. CBR systems may be implemented and automated in different ways depending on the problem which must be solved. Several strategies may be implemented in the framework of CBR systems to automate the planning tasks [9, 10]. A variational calculus strategy has been proposed in [2] for the adaptation stage of CBR systems embedded in the agent. We have experimented with several planning strategies and observed that the Variational Calculus Based Planer (VCBP) by [2, 3] provides excellent results and can be compared very favorably with other planning strategies [3]. Variational Calculus allows re-planning at execution time, even in changing environments, where goals are to be achieved successfully in real-time.

Variational calculus allows the agents to plan and replan at execution-time because this formalism is used to model the cases during the reuse phase of the reasoning process to solve a given problem. Assuming that potentially significant changes can be determined after executing a primitive action, it is possible to control the dynamism of the new events of the domain and thus achieve an appropriate reconsideration of the problem [11]. When the plan proposed by the agent is stopped for any reason, variational calculus calculates a new plan. In this case the new initial state is the point at which the initial proposed route has stopped. If it is accepted that the environment may change, it is also necessary to define a reasoning mechanism capable of dealing with such changes by modifying the initial desires and intentions.

Nevertheless the reasoning process may be maintained since the general description problem remain constant. If at $t_0$ the function V(X, Y, Z) takes the form denoted by $V_0$(X,Y,Z), at $t_1$, V is denoted by $V_1$ (X, Y, Z), with the associated surface $\Pi_1$ (X, Y, Z) = 0 on the phase space, upon which it is possible to obtain the optimal curve between two new points, $S_i$ and $S_f$ where $S_i = S_1^{(0)}$ and $S_1^{(0)}$ is the second state of $\psi_0 = \{ S_i = S_0^{(0)}, S_1^{(0)}, \ldots, S_s^{(0)} = S_f \}$ and $S_f$ is the final state or solution state of the global problem. Solving the Euler´s equations, $\boldsymbol{\chi_1 = \chi_1(X, Y, Z)}$ is obtained, which may be used to calculate an expression for $\boldsymbol{\psi_1}$, denoted as $\boldsymbol{\psi_1 = \{ S_i = S_1^{(0)}, S_1^{(1)}, S_2^{(1)}, S_3^{(1)}, S_4^{(1)}, \ldots, S_m^{(1)}, \ldots, S_s^{(1)} = S_f \}}$ and the same can be done for any $\mathbf{t_j}$. From the previous equations, and based on variational calculus tools, an expression can be determined to identify the final solution of the agent. This expression, which represents the agent plan, can be obtained in execution-time and takes the following form:

$$\Psi_{final} = \begin{cases} \Psi_0, \ldots \ldots\; t \in (t_0, t_1) \\ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \\ \Psi_{s-1}, \ldots\; t \in (t_{s-2}, t_{s-1}) \\ \Psi s, \ldots \ldots\; t \in (t_0, t_1) \end{cases}$$

## 3.1 K-SIM data processing strategy

The performance of the variational calculus strategy used in the reuse stage relies on the effectiveness of the retrieval algorithm. The more consistence and adequate the outlines cases, the better the performance of the planning strategy. This paper presents an algorithm, that has been developed for case indexing and retrieval and that has been presented in [1]. Kernel Maximum Likelihood Hebbian Learning Scale Invariant Map (K-SIM) is based on a modification of a new type of topology preserving map that can be used for scale invariant classification [12]. Kernel models were first developed within the context of Support Vector Machines [13]. Support Vector Machines attempt to identify a small number of data points (the support vectors) which are necessary to solve a particular problem to the required accuracy. Kernels have been successfully used in the unsupervised investigation of structure in data sets [11, 14, 15].

Kernel methods map a data set into a feature space using a nonlinear mapping. Then typically a linear operation is performed in the feature space; this is equivalent to performing a nonlinear operation on the original data set. The Scale Invariant Map is an implementation of the negative feedback network to form a topology preserving mapping. A kernel method is applied to an extension of the Scale Invariant Map (SIM) which is based on the application of the Maximum Likelihood Hebbian Learning (MLHL) method [16]. This method automates the organisation of cases and the retrieval stage of case-based reasoning systems. The proposed methodology groups cases with similar structure, identifying clusters automatically in a data set in an unsupervised mode. The features that characterize Kernel models can be used to cluster cases, to identify cases that are similar to a given one and to reuse cases. Then, the VCBP can generate a smoother representation of the problem and identify a more consistent plan.

# 4 Initial evaluation and conclusion

The proposed system has been used to improve an agent based system developed for guiding tourist around a city. The initial system has been presented in [2, 3]. Basically the assistant agent can be contacted via Internet or wireless devices such as mobile phones, PDAs, etc. The initial advising agent has been improved using the methodology presented in the previous section. The agent shares information with other agents that maintain uptodate information about Salamanca, its monuments, restaurants, spectacles, etc. The agent interact with the user and provides them with a plan. The tourist may use a mobile device to contact the agent, and then introduces his/her login and password, and indicates to the agent his/her preferences (monuments to visit, visits duration, time for dinner, amount of money to spend, etc.). The agent then generates a plan for the user according to his/her preferences and sends it back to the visitor.

Once the agent is contacted and knows what are the interest of the tourist, it stars it reasoning process. First identifies what cases should be retrieved to create the model

This work has been partially supported by the Spanish MCyT under project TIC2001-4936-E

of the problem using the K-SIM algorithm and then to identify the plan using the VCBP algorithm. If at any point the tourist decides to change his mind, the agent may run the K-SIM algorithm, and the VCBP planner to modify the initial plan in execution time, taking into consideration the initial constraints together with new ones.

The initial system, was tested from the 1st of June to the 15th of September 2002. The case base was initially filled with information collected from the 1st of February to the 25th of May 2002. Local tourist guides provided the agent with a number of standard routes and distributed among his clients Mobile phones, from which they could contact the agent and inform it about the progress of their plans: routes, times, evaluations, etc. As reported in [2], during this period the agent stored in its memory 540 instances. Which covered a wide range of all the possible options that offers the City of Salamanca. The system was tested during 115 days and the results obtained were very encourages. Three hotels of the City offered the option to their 4216 guests to use the help of the agent or a professional tourist guide, 7% of them decided to use to agent based system and 28% of them used the help of a tourist guide. The rest of the tourists visited the city by themselves. In this initial experiment the agent intentions were related to a one-day route (a maximum of 12). The degree of satisfaction of the tourist that used the help of the agent based tourist guide was very high. The new methodology has tested on bench, and we only have preliminary results, which can not be compared with the previously obtained and reported in [2], never the less we have observed that plans obtained with the improved system are more efficient in terms of the reduction of times spent between visits, identification of more convenient restaurants and monuments to visit with respect to the location and distance. The system will be operational in January 2004 and will be tested for a four month period.

## Acknowledgements

This work has been partially supported by the CICYT projects TEL99-0335-C04-03 and SEC2000-0249 and the project SA039/02 of the JCyL.

## Bibliography

1 Corchado J. M., Corchado E. S., Aiken J., Fyfe C., Fdez-Riverola F. and Glez-Bedia M. (2003) Maximum Likelihood Hebbian Learning Based Retrieval Method for CBR Systems. 5th International Conference on Case-Based Reasoning, Trondheim, Norway, June 23 to 26, 2003

2 Glez-Bedia M. and Corchado J. M. (2002). Variational Calculus Bases Planning for Deliberative Agents. I Workshop on Planning, Scheduling and Temporal Reasoning. Sevilla, Spain, 12 November.

3 Glez-Bedia M., Corchado J. M., Corchado E. S. and Fyfe C. Analytical Model for Constructing Deliberative Agents, Engineering Intelligent Systems, Vol 3: pp. 173-185.

4 Glez-Bedia M. and Corchado J. M. Constructing autonomous distributed systems using CBR-BDI agents. Innovation in knowledge Engineering. R. J. Howlett and L.C. Jain (eds.) Advanced Knowledge International.
5 Jennings N. R. (1992) On Being Responsible. In Y. Demazeau and E. Werner, editors, Decentralized A.I. 3. North Holland, Amsterdam, The Netherlands.
6 Wooldridge M. and Jennings N. R. (1994) Agent Theories, Architectures, and Languages: A Survey. Procs. ECAI-94 Workshop on Agent Theories, Architectures, and Languages.
7 Bratman M.E. (1987) Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A.
8 Kinny D. and Georgeff M. (1991) Commitment and effectiveness of situated agents. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI'91), pp. 82-88, Sydney, Australia.
9 Corchado J. M. and Aiken J. Hybrid Artificial Intelligence Methods in Oceanographic Forecasting Models. IEEE SMC Transactions Part C. Vol. 32, No.4. pp. 307-313.
10 Corchado J. M. and Lees B. (2001) A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence. Vol 15, no. 2, pp105-127.
11 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
12 Fyfe C. (1996) A Scale Invariant Map Network. Computation in Neural Systems, Vol. 7, pp. 269-275.
13 Vapnik V. (1995) The nature of statistical learning theory", Springer Verlag.
14 Scholkopf B., Smola A. and Muller K. R. (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, Vol.10, pp. 1299-1319.
15 Fyfe C., MacDonald D., Lai P. L., Rosipal R. and Charles D. (2000) Unsupervised Learning with Radial Kernels. In Recent Advances in Radial Basis Functions, Editors R. J. Howlett and L. C. Jain.
16 Corchado E. S., MacDonald D. and Fyfe C. (2003) Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery. In Press.

This work has been partially supported by the Spanish MCyT under project TIC2001-4936-E