

Running agents in mobile devices

Javier Bajo¹, Juan M. Corchado², L.F. Castillo³

¹Universidad Pontificia de Salamanca
Compañía 5, 37002, Salamanca, Spain
jbajoje@upsa.es

²Departamento Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
corchado@usal.es

³Dep. Ciencias Computacionales, Universidad Autónoma de Manizales
Antigua Estación del Ferrocarril, Manizales, Colombia
lfcastillo@autonoma.edu.co

Abstract. This paper presents a deliberative architecture based on the concept of CBP-BDI agent. A CBP-BDI agent is a deliberative agent that incorporates a case based planner engine. The work here presented focuses in the development of a multiagent system that has been constructed for the management of some aspects of a shopping mall, specially the interaction with clients; here the aim is to get the portability of a CBP-BDI agent to mobile devices. The system has been tested and this paper presents the obtained results.

1. Introduction

Multiagent systems have become increasingly relevant for developing applications in dynamic, flexible environments, such as the internet, personalized user interfaces, oceanography, control systems or robotic. Agents can be characterized through their capacities such as autonomy, reactivity, pro-activity, social abilities, reasoning, learning and mobility. These capacities can be modelled in various ways, using different methodologies [21]. One of the possibilities is to use Case Based Reasoning (CBR). This paper presents a distributed architecture whose principal characteristic is the use of CBP-BDI agents. These deliberative agents incorporate a reasoning Case Based Planning (CBP) engine, a variant of the CBR system which makes it possible for the agents to learn from initial knowledge, interact autonomously with the environment and users, and allows it to adapt itself to environmental changes.

The aim of this work is to obtain an architecture that allows the development of agents which incorporate CBP reasoning mechanisms, for dynamic environments. To achieve this aim we have concentrated in a specific problem, the management of some aspects of a shopping mall, and we use an architecture that makes it possible to construct agents capable of adapting its knowledge to environmental changes. There are many different architectures for constructing deliberative agents and many of them are based on the BDI model. In the BDI model, the internal structure of an agent and its capacity to choose, is based on mental aptitudes. This has the advantage of

using a natural model and a high level of abstraction. The BDI model uses the agent's beliefs as informational aptitudes, its desires as motivational aptitudes and its intentions as deliberative aptitudes. The method proposed in [7] facilitates the incorporation of CBR systems as a deliberative mechanism within BDI agents, allowing them to learn and adapt themselves, lending them a greater level of autonomy than pure BDI architecture [15].

The system incorporates "lightweight" agents that can live in mobile devices, such as phones, PDAs, etc. [8, 20]. These agents make it possible for a client to interact with the MAS in a very simple way, downloading and installing a personal agent in his mobile phone or PDA. Case-Based Mark-up Language (CBML) [12], a XML based language for representing CBR components, is used in order to allow us to work within mobile devices. So the formal definition of the structure of our cases and similarity measures are completely independent of the application code and CBR components can be exchanged between heterogeneous CBR systems. The agents are adapted to work in mobile devices, so they support wireless communication (Wi-Fi, Bluetooth) which facilitates the portability to a wide range of mobile devices [8].

One of the major problems in the development of an architecture based on multiagent systems is that there are currently no clear standards or well developed methodologies for defining the steps of analysis and design that need to be taken. There are at present a number of methodologies: Gaia [22], AUML [4, 19], MAS-CommonKADS [16], MaSE [13], MESSAGE [14]. For this study, we have decided to opt for Gaia for our MAS. Gaia is a simple methodology that allows us to carry out a preliminary analysis and design with which to confront the problem at a general level. The great advantage is that we can carry out a rapid, broad study. we are able to obtain both a generalized vision of the problem in terms of organization, which helps enormously in the development of such a research project.

In the next section, we will explain the shopping mall problem that has led to most of this research. In the third section we will describe the wireless multiagent system developed. In the fourth section, the case based planning system will be presented. Finally, some preliminary results and the conclusions will be presented.

2. A case study

The Mall has become one of the most prevalent alternative to traditional shopping [2]. A shopping mall is a cluster of independent shops, planned and developed by one or several entities, with a common objective. The size, commercial mixture, common services and complementary activities developed are all in keeping with their surroundings [2]. Every shopping mall has a permanent image and a certain common management. A shopping mall needs to be managed and, the management includes solving incidents or problems in a dynamic environment. As such, a shopping mall can be seen as a large dynamic problem, in which the management required depends on the variability of the products, clients, opinions, etc. Within this framework, the multiagent system technology developed in this project will make it possible to provide better services to the shopping mall clients. Our aim is to develop an open system, capable of incorporating as many agents as necessary, agents that can provide

useful services to the clients not only in this shopping centre, but also in any other environment such as the labor market, educational system, medical care, etc. The system provides mechanisms for easy data consulting. A user will be able to gain access to shopping and sales and leasing time information (entertainment, events, attractions, etc.) by using their mobile phone or PDA. Mechanisms for route planning when a user wants to spend time in the mall will also be available. Moreover, it provides a tool for advertising offers (a commercial manager will be able to make his offers available to the shopping mall clients), or provides a tool to the shopping mall management team in order to contact commercial managers or shopping mall clients, providing an interaction between users interested in the same topics.

3. SM-MAS: Shopping Mall Multiagent System

The option chosen to define an appropriate analysis and design methodology for the problem to be resolved is Gaia [22]. Gaia is a methodology for analysis and design very general and therefore applicable to a very wide range of multiagent systems. It also allows the user to have a wide knowledge of the multiagent systems both at an organizational (social) level and at a detailed level for each agent.

Rol: INCIDENTS MANAGER
<p>Description: Manages the incidents in the SMA. Moreover manages the orders with the suppliers.</p>
<p>Activities and Protocols:</p> <ul style="list-style-type: none"> ▪ ManageSec ▪ ManageCl ▪ ManageCh ▪ ManageRestock ▪ RequestIncid ▪ SupplierSel ▪ RequestUpdateSt ▪ ManageNotices ▪ InformNotice
<p>Permissions:</p> <p>Read</p> <ul style="list-style-type: none"> ▪ DB Incident. <p>Change</p> <ul style="list-style-type: none"> ▪ DB Incident. <p>Generates</p>
<p>Responsibilities:</p> <p>Liveness:</p> <ul style="list-style-type: none"> ▪ MANAGEINCID: (ManageSec ManageCl ManageCh ManageRestock).RequestIncid ▪ SUPPLYPROD: SupplierSel.RequestUpdateSt ▪ SENDADVICE: (ManageNotices.InformNotice)*
<p>Responsibilities:</p> <p>Safety:</p> <ul style="list-style-type: none"> ▪ Successful connection with Incident DB

Fig. 1. Gaia roles model: Incidents Manager role.

Through the Gaia analysis, two models are obtained: the role model and the interaction model. Studying the requirements of the problem we have come to the conclusion that we need nine roles: The Communicator role manages all the communications of a client. The Finder role looks for near devices. The Profile Manager role obtains a client profile. The Store operator is in charge of manage the store: data base operations on stored products. Moreover monitors the products

shortage, in order to prevent desupply. The Promotions manager role controls the shells in each shop, as well as the promotions that every shop offers to its clients. The Clients Manager role deals with the client profiles management and controls the connected clients at a given moment. The Analyst role carries out periodic evaluations on shells, promotions and surveys data trying to provide a good quality service. The Incidents Manager role manages incidents, such as sending advices, or solving a wide range of problems. The Planner role is the most important role in our system. The Planner creates a route printing the most suitable shops, promotions or events to the client profile and available resources at one particular moment. As can be seen in Figure 1, the Incidents Manager role is composed of responsibilities, permissions, activities and protocols [22]. It is authorized to read and change the Incidents DB, and it is responsible for the incidents management, product orders and sending advices. Besides it must maintain a successful connection with the Incidents DB.

As far as interaction model is concerned, the dependences and relations between roles are described. Each interaction in which two roles are involved requires protocols (described in the roles model). In the SMA presented in this work the next protocols have been considered: RequestPromotionsData, SolveConsult, StoreProducts, AlertShortage, OrderSupplier, InformProductsState, InformPromotionsState, SolveIncident, SolveSuggestion, Notify.

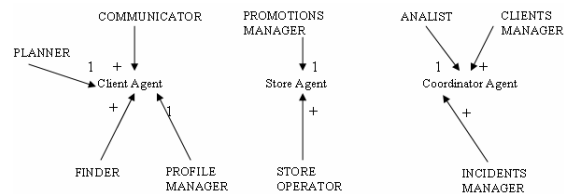


Fig 2. Gaia agent model for the shopping mall problem.

Once the analysis has been finalised, the Gaia design is carried out. Traditional techniques of software engineering are not followed in terms of detailing the analysis to the extent that a direct implementation can be made. Instead, the level of abstraction is reduced so that traditional techniques can be applied. In the design process three models are considered: agents model, services model and acquaintance model [22]. As we can see in Figure 2, the agents model shows the types of agents that are going to appear in the system, as well as the number of instances for each agent type that can be executed within the execution time. For example agent Store plays the Promotions Manager and Store Operator roles.

4. Deliberative behaviour.

The purpose of case-based reasoning (CBR) is to solve new problems by adapting solutions that have been used to solve similar problems in the past. The CBP systems are a variation of the CBR systems, which are based on the plans generation from cases. The deliberative agents, proposed in the framework of this investigation, use this concept to gain autonomy and improve their problem-solving capabilities. As

described in [15], in a CBP-BDI agent, each state is considered as a belief; the objective to be reached may also be a belief. The intentions are plans of actions that the agent has to carry out in order to achieve its objectives [6, 7], 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 specified state, and the subsequent result). 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 achieve a similar result to the previously obtained one). Next, the CBP planner used for the Client Agent to find a route in the shopping mall is presented: Let $E = \{e_0, \dots, e_n\}$ the set of possible interesting places to visit and buy.

$$a_j : E \xrightarrow{e_i} E \quad (1)$$

$\qquad \qquad \qquad \rightarrow \quad a_j(e_i)=e_j$

An Agent plan is the name given to a sequence of actions (1) that, from a current state e_0 , defines the path of states through which the agent passes in order to offer to the client the better path according to each client's characteristics. Below, in (2), the dynamic relationship between the behaviour of the agent and the changes in medium is modelled. The behaviour of agent A can be represented by its action function $a_A(t) \forall t$, defined as a correspondence between one moment in time t and the action selected by the agent,

$$Agent A = \{a_A(t)\}_{t \in T \subseteq N} \quad (2)$$

From the definition of the action function $a_A(t)$ a new relationship that collects the idea of an agent's action plan (3) can be defined,

$$p_A(t_n) = \int_{t_0}^{t_n} a_A(t) dt \quad (3)$$

The variation of the agent plan $p_A(t)$ will be provoked essentially by: The changes that occur in the medium and that force the initial plan to be modified, and the knowledge from the success and failure of the plans that were used in the past, and which are favoured or punished via learning. O indicates the objectives of the agent and O' are the results achieved by the plan. R are the total resources and R' are the resources consumed by the agent. The efficiency of the plan (4) is the relationship between the objectives attained and the resources consumed

$$E_{eff} = \frac{\#(O' \cap O)}{\#R'} \quad (4)$$

The objective is to introduce an architecture for a planning agent that behaves – and selects its actions – by considering the possibility that the changes in the environment block the plans in progress. This agent is called MRPI (most re-plan-able Intention agent) because it continually searches for the plan that can most easily be re-planned in the event of interruption. Given an initial point e_0 , the term planning problem is used to describe the search for a way of reaching a final point $e_i \equiv e^* \in E$ that meets a series of requirements. Given a problem E and a plan p(t) the functions *Ob* and *Rc*

accumulated are constructed from the objectives and costs of the plan (5). For all time points t_i associate two variables are associated:

$$Ob(t_i) = \int_a^{t_i} O(t)dt \quad Rc(t_i) = \int_a^{t_i} R(t)dt \quad (5)$$

This allows us to construct a space representing the environment for planning problems as a vectorial hyper dimensional space where each axis represents the accumulative variable associated with each objective and resource. In the planning space, defined in this way, conform to the following properties:

1. Property 1: The representations of the plans within the planning space are always monotonously growing functions. Given that $Ob(t)$ and $Rc(t)$ are functions defined as positive, function $p(t)$ expressed at these coordinates is constant or growing.
2. Property 2: In the planning space, the straight lines represent plans of constant efficiency. If the representations of the plans are straight lines, the slope of the function is constant, and coincides with the definition of the efficiency of the plan. $\frac{d}{dt} p(t) = cte \Leftrightarrow \lim_{\Delta \rightarrow 0} \frac{\Delta O(t)}{\Delta R(t)} = cte$

In an n-dimensional space, the extension of the straight concept line is called a geodesic curve. In this sense, the notion of geodesic plans can be introduced, defined as those that maintain efficiency at a constant throughout their development. The agent must search for the plan that determines a solution with a series of restrictions $F(O;R) = 0$. In the plans base we seek those plans that are initially compatible with the problem faced by the agent, with the requirements imposed on the solution according to the desires, and in the current state [1]. If all the possible plans $\{p_1, \dots, p_n\}$ are represented within the *planning space*, a *subset of states* that the agent has already attained in the past will be obtained in order to resolve similar problems. With the mesh of points obtained (generally irregular) within the planning space and using interpolation techniques, working hyperplan $h(x)$ (that encapsulates the information on the set of restrictions from restored experiences) can be obtained, from which geodesic plans can be calculated and with which variation calculation is applied. Suppose, for simplicity's sake, that we have a planning space of dimension 3 with coordinates $\{O, R_1, R_2\}$. Between the point e_0 and the objective points $f_s, f = \{e_1, \dots, e_m\}$ and over the interpolation surface $h(x)$, the Euler Theorem [15, 17] guarantees that the expression of the geodesic plans will be obtained by resolving the system of equations in (6), where R_i is the function accumulated R, O is the function of accumulated O and L is the distance function on the hyperplan $h(x)$, $L = \int_h dl$.

In order to obtain all the geodesic plans that, on the surface $h(x)$ and beginning at e_0 , allows us to reach any of the points $e^* \in f_s, f$, a condition of the surrounding must be imposed: the initial point will be $e_0 = (O_0, R_0)$. Once plans have been obtained that will create efficient solutions, the plan around it (along its trajectory) will be calculated by a denser distribution of geodesic plans. The tool that allows to determine this is called the minimum Jacobi field associated with the solution set

[18]. $g_0 : [0,1] \rightarrow S$ be a geodesic over a surface S . Let $h : [0,1] \times [-\varepsilon, \varepsilon] \rightarrow S$ be a variation of g_0 so that for each $t \in (-\varepsilon, \varepsilon)$, the set $\{h_t(s)\}_{s \in [0,1]} : h_t(s) \quad \forall t \in (-\varepsilon, \varepsilon)$ are geodesic in S and they begin at $g_0(0)$, in other words, they conform to $h_t(0) = g_0(0) \quad \forall t \in (-\varepsilon, \varepsilon)$. In these conditions, taking the variations to a differential limit

$$\begin{cases} \frac{\partial L}{\partial R_1} - \frac{d}{ds} \frac{\partial L}{\partial R_1} = 0 \\ \frac{\partial L}{\partial R_2} - \frac{d}{ds} \frac{\partial L}{\partial R_2} = 0 \end{cases} \quad (6)$$

$$\lim_{t \rightarrow 0} \{h_t(s) = g_0(s+t)\} = \lim_{t \rightarrow 0} \{h(s,t)\} = \left. \frac{\partial g_0}{\partial t} \right|_{(s,0)} = \frac{dg_0}{ds} \equiv J_{g_0}(s) \quad (7)$$

The term $J_{g_0}(s)$ is given to the Jacobi Field of the geodesic g_0 for the set $\{g_n(x)\}_{n \in N}$, and in the same way that the definition has been constructed, it is possible to give a measurement for the distribution of the other geodesics of $\{g_n(x)\}_{n \in N}$ around g_0 throughout the trajectory. Given a set of geodesics, some of them are always g^* that, in their environment, have a greater distribution than other geodesics in a neighbouring environment. This is equivalent to saying that it presents a variation in the distribution of geodesics lower than the others and therefore the Jacobi Field associated with $\{g_n(x)\}_{n \in N}$ reaches its lowest value at J_{g^*} . Let's return to the MRPI agent problem that, following the recuperation and variation calculation phase, contains a set of geodesic plans $\{p_1, \dots, p_n\}$. If the p^* is selected with a minimum Jacobi Field value, we can guarantee that in the event of interruption it will have around it a greater number of geodesic plans in order to continue. This suggests that given a problem with certain restrictions $F(O;R)=0$, the geodesic plan p^* with minimum associated Jacobi field associated with the set $\{g_n(x)\}_{n \in N}$ is called the most re-plan-able solution. In this way, the behaviour model G for the MRPI agent is (8).

$$G(e_0, p_1, \dots, p_n) = p^* \Leftrightarrow \exists n \in N / J_{g_n} \equiv J_{g^*} = \underset{n \in N}{\text{Min}} J_{g_n} \quad (8)$$

If the plan p^* is not interrupted, the agent will reach a desired state $e_j \equiv e^* \in f, f \quad j \in \{1, \dots, m\}$. In the learning phase, a weighting $w_f(p)$ is stored. With the updating of weighting $w_f(p^*)$, the planning cycle of the CBP motor is completed. In Figure 3, it is possible to see what happens if p^* is interrupted. Let's suppose that the agent has initiated a plan p^* but at a moment $t > t_0$, the plan is interrupted due to a change in the medium. The geodesic planning meets the conditions of the Bellman Principle of Optimality [5], in other words, each one of the plan's parts is partially geodesic between the selected points. This guarantees that if g_0 is geodesic for interrupted e_0 in t_1 , because e_0 changes to e_1 , and g_1 is geodesic to e_1 that is begun

in the state where g_0 has been interrupted, it follows that: $g = g_0 + g_1$ is geodesic to $e = e_0(t_1 - t_0) + e_1(t_2 - t_1)$

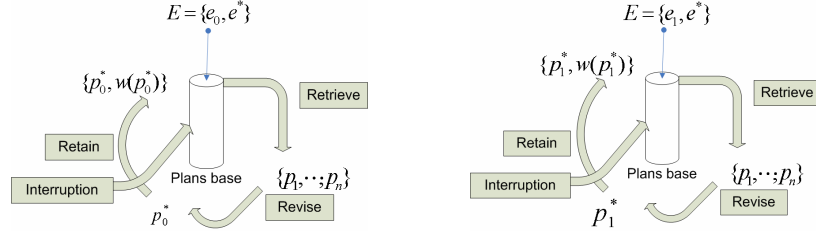


Fig. 3. Model for behaviour $G(t)$.

The dynamic process follows the CBP cycle recurrently: each time a plan finds itself interrupted, it generates from the state reached so far, the surroundings of the plans from the case base and adjusts them to the new problem. With this it calculates the geodesic plans and selects the one which meets the minimum conditions of the associated Jacobi field. In this way the dynamic planning model of the agent $G(t)$ is characterised as it is shown in the Figure 4. A minimum global Jacobi field $J(t)$ also meets Bellman's conditions of optimality [5], in other words, a minimum global Jacobi field, must select minimum Jacobi fields "in pieces" (9).

$$J_{\min}(t) = \{J_{\min}(t_1 - t_0), J_{\min}(t_2 - t_1), \dots, J_{\min}(t_n - t_{n-1})\} \quad (9)$$

If on the one hand, successive Jacobi fields generate one Jacobi field, and on the other hand, minimum Jacobi fields generate a minimum Jacobi field, the MRPI agent that follows a strategy of replanning $G(t)$ as indicated in order to survive a dynamic environment, it generates a global plan $p^*(t)$ that, faced with all possible global plans $\{p_n(t)\}_{n \in N}$, presents a minimum value in its Jacobi field $J_g(t) \equiv J_p(t)$. Up until now, an agent that in a dynamic environment seeks plans that lend it greater capacity for replanning has been formally defined.

5. Results

The previously described system was tested at the Tormes Shopping Mall in the city of Salamanca during 2005. During this period of time, the multiagent system has been tuned and updated, and the first autonomous prototype started to work in October 2005. Although the system is not fully operational and the aim of the project is to construct a research prototype and not a commercial tool, the initial results have been very successful from the technical and scientific point of view. The construction of the distributed system has been relatively easy using previously developed CBR-BDI libraries [3, 9, 10, 11]. Gaia [22] provides an adequate framework for the analysis and design of distributed agent based systems. The formalism defined in [17] facilitates the straight mapping between agent definition and CBR construction.

The fundamental concept when we work with a CBR system is the concept of case, and it is necessary to establish a case definition. A case structure in our problem, managed by the Client agent, is defined using CBML [12] as can be seen in Figure 4. The structure is defined through the feature labels. At execution time the cases are instantiated as shown in Figure 4. The instance contains the user profile and his shells.

```

<case domain="ComercioBluetooth" xsi:noNamespace
<structure>
  <feature name="Age">
    <integer>
      <minInclusive value="4"/>
      <maxInclusive value="100"/>
    </integer>
  </feature>
  <feature name="Gender">
    <symbol>
      <enumeration value="male"/>
      <enumeration value="female"/>
    </symbol>
  </feature>
  <feature name="YearSalary">
    <double>
      <minInclusive value="0.6"/>
      <maxInclusive value="100000"/>
    </double>
  </feature>
  <feature name="Contract">
    <symbol>
      <enumeration value="None"/>
    </symbol>
  </feature>
</structure>
</case>
- <casebase domain="ComercioBluetooth">
- <case name="Customer1">
  <Age>23</Age>
  <Gender>male</Gender>
  <YearSalary>24000</YearSalary>
  <Contract>None</Contract>
  <MaritalStatus>Single</MaritalStatus>
  <InternetUser>true</InternetUser>
  <Amounts>60</Amounts>
  <Nationality>European</Nationality>
  <studies>PreGrade</studies>
  <Shoppingtime>120</Shoppingtime>
- <Buy>
- <article>
  <desc>pants</desc>
  <price>45</price>
  <units>1</units>
</article>
</Buy>
</Case>

```

Fig. 4. CBML Cases structure and case instante.

The system implementation has involved an increase in benefits due to the generation of automatic promotions. The e-commerce techniques [2] have facilitated custom-designed services for the clients. A user can easily find the products he is interested on, spend his leisure time in a more efficient way and make contact with other clients with whom he can share hobbies or opinions. So the degree of client satisfaction has been improved as observed in the surveys. The percentage of the sale of promotional products has grown over the total. The fundamental reason is that clients have instantaneous information about the products the agent thinks they are interested on, and the information is very accurate and customized.

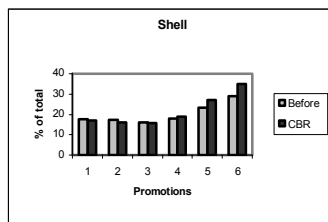


Fig. 5. Shell promotional products and shell total products.

The increase in promotional products sold can be seen in Figure 5 that shows the increased percentage in promotional products sold in comparison with the promotional products sold using traditional commercial techniques carried out the year before. We can observe that at the beginning, the results obtained with the multiagent system were worse than traditional techniques. However, as the system obtained more information about client profiles, products and habits, so the knowledge it obtained became more suitable and it was able to create optimal plans. Moreover the clients also needed some time to get used to the new system.

References

- [1] Aamodt A. and Plaza E. (1994) Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches, AICOM. Vol. 7., March, pp 39-59.
- [2] Adams, F.G. (2003): The E-Business Revolution & the New Economy: E-economics after the Dot-Com Crash. South-Western Educational Pub.
- [3] Bajo J. and Corchado J.M. Evaluation and monitoring of the air-sea interaction using a CBR-Agents approach Proceedings of the 6th Internacional Conference on Case-based Reasoning, ICCBR'05 pp. 50-62. Springer Verlag. (2005)
- [4] Bauer, B. and Huget, M. P. (2003) FIPA Modeling: Agent Class Diagrams.
- [5] Bellman, R.E. (1957). Dynamic Programming. Princeton University Press, Princeton, New Jersey.
- [6] Bratman M.E., Israel D., and Pollack M.E. (1988). Plans and resource-bounded practical reasoning. Computational Intelligence, 4. pages 349-355.
- [7] Bratman, M.E. (1987). Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A.
- [8] Corchado J.M., Corchado E. y Pellicer M.A. (2004). Design of cooperative agents for mobile devices. Proceedings of the International Conference on Cooperative Design, Visualization and Engineering - CDVE2004. Luo Y. (Ed.) LNCS 3190, Springer Verlag. ISSN 0302-9743. ISBN 3-540-23149-8 pag. 205-212.
- [9] Corchado J. M. and Laza R. (2003). Constructing Deliberative Agents with Case-based Reasoning Technology, International Journal of Intelligent Systems. Vol 18, No. 12, December. pp.: 1227-1241
- [10] Corchado J. M. and Lees B. (2001). A Hybrid Case-based Model for Forecasting. Applied Artificial Intelligence. Vol 15, no. 2, pp.105-127.
- [11] Corchado J. M., Pavón J., Corchado E. and Castillo L. F. (2005) Development of CBR-BDI Agents: A Tourist Guide Application. 7th European Conference on Case-based Reasoning 2004. LNCS 3155, Springer Verlag. pp. 547-559.
- [12] Coyle, L., Cunningham, P. & Hayes, C. (2002). Representing Cases for CBR in XML. Proceedings of 7th UKCBR Workshop, Peterhouse, Cambridge, UK.
- [13] DeLoach, S. (2001) Anlysis and Design using MaSE and AgentTool. Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS).
- [14] EURESCOM (2001) MESSAGE: Methodology for engineering systems of software agents. Technical report P907-TII, EURESCOM.
- [15] Glez-Bedia M. and Corchado J. M. (2002) A planning strategy based on variational calculus for deliberative agents. Computing and Information Systems Journal. Vol 10, No 1, 2002. ISBN: 1352-9404, pp. 2-14.
- [16] Iglesias, C., Garijo, M., Gonzalez J.C. and Velasco J. R. (1998) Analysis and Design using MAS-CommonKADS. Intelligent Agents IV LNAI Volume 1365 Springer Verlag.
- [17] Jost, J. and Li-Jost, X. (1998). Calculus of variations. Cambridge University Press, U.K.
- [18] Lee, J.M. (1997). Riemannian Manifolds. An introduction to Curvature. Springer-Verlag, New York, Inc.
- [19] Odell, J. and Huget, M. P. (2003) FIPA Modeling: Interaction Diagrams.
- [20] Sun Microsystems. (2005). Java 2 Micro Edition (J2ME). <http://java.sun.com/j2me/>.
- [21] Wooldridge, M. and Jennings, N. R. (1995) Agent Theories, Architectures, and Languages: a Survey. In: Wooldridge and Jennings, editors, Intelligent Agents, Springer-Verlag, pp. 1-22.
- [22] Wooldridge, M. and Jennings, N. R. and Kinny, D. (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems, 3 (3). pp. 285-312.