Computing and Informatics, Vol. 24, 2005, 427–440

MULTIPLE MOBILE AGENT SYSTEM FRAMEWORK SUITABLE FOR PERVASIVE COMPUTING

Francisco Maciá-Pérez, Juan Manuel García-Chamizo
Antonio Soriano-Payá, Daniel Ruiz-Fernández

Department of Computer Technology and Information
University of Alicante
P. O. 99, E-03080, Alicante, Spain
e-mail: {pmacia, juanma, soriano, druiz}@dtic.ua.es

Manuscript received 24 March 2005; revised 29 June 2005
Communicated by Jozef Kelemen

Abstract. In this article we present a formal framework based on the action and reaction model that allows us to cover the dynamics of multi-agent systems (MAS) made up of mobile software agents suitable for scalable networks. This model is based on the operation of the human nervous centres. At the present time, we are applying it in works related with the control of biological systems and also in those related to the network management. In the case of systems based on mobile agents, the main problem is the different vision the agents have of the world and the impossibility of being aware of and synchronizing all the influences brought by the different agents acting on it. We have compared our proposal with the conventional MAS by solving an extension of the predator-prey problem. The results show the advantages of mobility as the size of the problem grows in a distributed system.

Keywords: Multi-agent system, distributed artificial intelligence, scalable networks, distributed systems, mobile agents

1 INTRODUCTION

In the current communications environment, there is a tendency towards more and more heterogeneous networks. This diversity means that network managers are handling more data and are required to compile huge amounts of information that must be analyzed before undertaking the actual management task. In addition, we also find that present-day network users increasingly expect a reliable and high quality of service. Of course, the ability to provide these guarantees depends on the dynamics of the state of the network, which are closely related to the types of traffic that users generate as part of their communications dialogues. Generally speaking, it requires the implementation of highly sophisticated control and signal techniques [12], which represents the greatest obstacle to network integration.

These are the main arguments determining research into mobile software agents applied to communications network management [1]. However, although formalisms that provide a suitable formulation for specifying multi-agent systems (MAS) can be found in literature, we cannot find formalisms that provide for the peculiarities produced by mobile agents.

Throughout this article, we will describe the background relative to the basic formulation for this type of framework in order to subsequently present our formalism based on the action and reaction model [4]. Once the framework for the performance environment has been established, we will define the agent model that will inhabit it, and, more specifically, focus on the hysteretic agent model proposed by Genesereth and Nilsson [6]. Taking this model as a basis, we will propose the modifications required for mobile software agents. We will subsequently present the model that must define the operation dynamics for a multi-agent system made up of multiple hysteretic software agents. From this formulation, we will propose a set of refinements that will reduce the effect of possible space and time inconsistencies between the agents and the medium, and, at the same time, achieve the generalization of the reaction function. We will propose a series of studies and experiments carried out with this model applied to a predator-prey problem variation, and finally we will present the conclusions drawn from this study and future lines of research arising from it.

2 BACKGROUND

Although we can find several formalisms for the representation and reasoning of concurrent processes [5, 7], they are of an extremely low level so that, generally speaking, they do not allow agents to be specified in mental state terms, nor do they represent actions explicitly in terms of their effects on the world [4].

Unlike the classical conception proposed by AI, in which the manifestation of intelligence is based on logical reasoning [2], MAS take as their starting point the agents' own behavior, from the actions they carry out in the world and the interaction among themselves. An action is, above all, a modification.

Anyway, although this is currently one of the most visible theories in the field of AI [13], it proves to be inadequate for situations in which there may be several agents carrying out different activities at the same time and in which they can find themselves in a situation of conflict.

This problem can be tackled by considering an action as a way of trying to influence the environment, by modifying it according to the agent's goal. However, the consequences of this action need not be reflected in the world according to its intentions; i.e., the actions carried out by agents must be separated from the effect that they really produce on the states of the world. Especially if we consider that, although actions are not produced explicitly on a specific world, its state is not immutable. In fact, the world is clearly in continuous evolution and keeps on changing without us having to assume the existence of external actions [4].

There are many ways of modeling actions and their consequences on the world. We will start out from an extension of the action model as a transformation of a global state, based on influences and reactions to influences. This extension provides a new model, known as the action as a response to influences, proposed by Ferber [3].

In the rest of the article, we will present our formalism constructively, and, whenever possible, based on current models.

3 FORMAL FRAMEWORK

In order to define the complete framework, we have divided its specification into three sections: specification of the agents' environment – their world – specification of the agents themselves, and specification of a system consisting of a world and multiple agents acting on it. The formulation used is based on the proposal for the action and reaction model [6].

3.1 Action and Reaction System

Let us suppose that it is possible to characterize the set $\Sigma = \{\sigma_1, \sigma_2, \ldots\}$ of possible world states. Using the algebraic notation specified by Pednault [10, 11] for ADL (Agent Dynamic Language) language, each world state can be defined by means of the structure $\sigma = \langle D, R, F, C\rangle$, where D is the domain, R is the set of relations, F is the set of functions and C are the constants. If we bear in mind that the domain and constants remain unaltered for all states, each world state σ_i could be defined by means of the structure: $\sigma_i = \langle D, R_i, F_i, C\rangle$.

Let us now suppose that we have a finite set P with all the possible tasks that can be carried out in a certain world. We will call each subset $\xi(P) \subset P$ a plan. Since a plan can be made up of one simple task or of a set of tasks, from now on we will use the terms plan or task without distinction. In the same way, each plan could be made up of all the tasks P, so that both sets would be totally interchangeable in our formulation. However, for generality, we will use the set P. We define the operators which perform the different tasks with a syntax similar to that for STRIP [15] operators, so that each operator will have the form $p = \langle \mathit{name}, \mathit{pre}, \mathit{post}\rangle$, being:

- *name*: This is an expresion with the form $f(x_1, \ldots, x_k)$, in which each x_i is a variable authorized to apper in pre and post formulas.

- *pre*: This represents a Boolean formula that must be true in order to carry out the activity defined in post.
- *post*: This represents a formula that defines the functionality of the task to be carried out together with the influence it aims to have on the world's new state.

Although this model can be regarded as an extension of the state transformation model, as we shall see, its main difference is that it will enable us to get a separate description of the desired objectives and the real effect produced in the environment. It is precisely this difference that enables us to study the execution of simultaneous actions in the same environment. In order to model this situation, we can define the set $\Gamma = \{\gamma_1, \gamma_2, \ldots\}$ for the possible influences or action attempts from the different agents with reference to the current state of the world. This structure is also described as a set of atomic formulas defined with the help of the world states themselves.

In this context, the actions are the result of the combination of the different contributions in the form of influences and the environment's reaction to them. In this way, the execution of a task $p \in P$ modifies the state of the world but we model it as a partially defined application, in which the result is not a new world state but an influence $\gamma \in \Gamma$ on it:

$$Exec : P \times \Sigma \to \Gamma. \tag{1}$$

According to this application, a task $p \in P$ can be executed in Σ, if and only if the application is defined for a specific state $\sigma \in \Sigma$ of the world. We formally express this fact with the predicate

$$\gamma = Exec\,(p, \sigma)\,. \tag{2}$$

The function *Exec* acts in the following way:

$$Exec\,(\langle name, pre, post\rangle\,, \sigma) = \{if \;\; pre\,(\sigma) \to post\;,\;\; else \to \{\}\}\,. \tag{3}$$

Since an influence can be the result of the simultaneity of actions carried out for a specific world state, we can extend the function Exec in order to provide for this fact. To do this, we define the simultaneity operator as $||$. This operator combines simultaneous actions and gathers the different influences produced by each of these in a vector. We now carry out the extension of the function by means of a morphism of the action space, equipped with the simultaneity operator $||$, acting on the set of influence Γ. Formally:

$$Exec : (P, ||) \times \Sigma \to \Gamma. \tag{4}$$

However, since the aim being sought with task execution is the transition from one world state to another, we must define a world reaction function for the different influences. Thus, the laws of the universe will be described by the application

$$React : \Sigma \times \Gamma \to \Sigma. \tag{5}$$

This application will be dependent on each type of environment and will have to be defined for each case. Finally, by using the definitions already proposed, we can describe the environment as a system of actions by means of the structure

$$\langle \Sigma, \Gamma, P, Exec, React \rangle \,. \tag{6}$$

Once the formal framework for the environment based on the action and reaction model has been established, we will present the agents as entities that are capable of influencing the world and distinguishing, from among all its states, those that are of interest for the tasks they must perform.

3.2 Agents

We will begin our definitions by focusing on the agent models proposed by Genesereth and Nilsson [6], pioneers in offering an algebraic representation of their structure and behavior. However, since this definition was developed on the concept of states, it can only be applied to mono-agent systems. To solve this, we will apply the influence and reaction model [3]. In spite of this, this model still fails to contemplate the possibility of multiple mobile software agents, so we will introduce the necessary modifications in order to solve this new problem. These modifications basically concentrate on the definition of an execution function disconnected as far as possible from the environment.

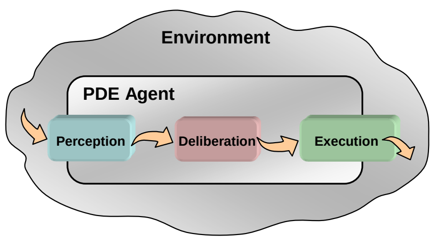

Fig. 1. Agent internal structure: Perception-Deliberation-Execution

As mentioned in the introduction, we started out from the idea of agents as entities that are permanently perceiving, deliberating and executing; that is, an agent made up of three very different parts: perception, deliberation and execution (Figure 1). Each agent is defined by the structure

$$\alpha = \langle Domain, Perception, Deliberation \rangle \tag{7}$$

in which:

Domain refers to the set of elements that make up the structure's domain and represent the perception the agent has of the world in which it is immersed.

Perception refers to the set of functions that enable the significant states of the world to be understood and classified.

Deliberation represents the set of functions that make tasks selection possible.

For an agent, perception represents the quality of being able to classify and distinguish different world states; not only with regard to the environment's most significant characteristics, but also with regard to the actions that are its responsibility. We can regard perception as a function that associates a set of values called perceptions or stimuli – perception when speaking of hysteretic agents and stimuli for tropistic agents – with a set of world states Σ. If we define the set $\Phi_\alpha = \{\phi_1, \phi_2, \ldots\}$ of possible perceptions associated with agent α, the agent's perception function can be defined as

$$Percept_\alpha : \Sigma \to \Phi_\alpha. \tag{8}$$

Finally, the capacity for deliberation remains to be defined. Situated between the agent's input and output, it is the element responsible for its current behavior. It is one of the most complex sections, in which we will define the goals, decision-making and memory faculties, if they have a memory, together with the representation of the world and the concepts used to decide what action to take. According to our definition of this behavior, we can distinguish two very general types of agent: tropistic and hysteretic agents [3]. The first type refers to agents motivated by stimuli that are not perceived by the conscience; so they will be unable to memorize them. The second type possesses behaviors that will be as sophisticated as we want and that use their previous experience to anticipate the future. Assuming that tropistic agents can be a specific case of hysteretic agents, we will focus our attention on the formulation of the latter type.

3.2.1 Hysteretic Agents

An agent of this type is characterized by the fact that it has an internal state that gives it the capacity to memorize and carry out a more valuable decision function than that studied in the case of tropistic agents.

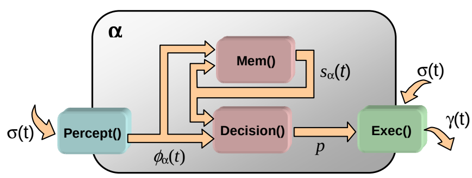

Fig. 2. Graphical representation of a hysteretic agent

If we define S_α as the set of internal states of a certain agent α, a hysteretic agent within the same action system (Equation 6) can be described using the structure

$$\alpha = \langle \Phi_\alpha, S_\alpha, Percept_\alpha, Mem_\alpha, Decision_\alpha \rangle, \tag{9}$$

that is to say, as a set of perceptions Φ_α, a set of internal states S_α and a series of functions that model perception, memorization and decision-making on behalf of the agent (Figure 2). The agent's deliberation capacity is defined in a more complex way by using two different functions: one for memorization and another for decision. The memorization of data consists of going from one internal state to another, so that the memorization function will associate with the agent's internal state and its present perception of the world with a new internal state

$$Mem_\alpha : \Phi_\alpha \times S_\alpha \to S_\alpha. \tag{10}$$

The decision function is responsible for associating a given task to be carried out by the agent from its perception of the world with the internal state in which it is found:

$$Decision_\alpha : \Phi_\alpha \times S_\alpha \to P. \tag{11}$$

In this case, the function that defines the agent's behavior will have to associate a pair made up of a world state and the agent's internal state, with another pair made up of the action produced together with the agent's new internal state:

$$Behave_\alpha : \Sigma \times S_\alpha \to P \times S_\alpha. \tag{12}$$

For example:

$$Behave_\alpha(\sigma, s) = \langle Decision_\alpha(\phi, s), Mem_\alpha(\phi, s) \rangle \; with \; \phi = Percept_\alpha(\sigma). \tag{13}$$

The result of a given behavior for a world state and the agent's internal state will be the result of decision and memorization from the perception that this agent has of the state of the world.

3.2.2 Hysteretic Mobile Agents

A hysteretic mobile agent is, above all, a hysteretic agent. Therefore, the definitions given in the previous section are perfectly valid for them. However, the notion of mobility implies the possibility that the agents may have to act in conditions with a lack of data. In these cases, the notion of autonomy implicit in the agents and which they must exhibit if we want the system to continue to evolve normally, becomes more obvious.

In order to make this task easier, our proposal focuses on including the execution function within the agent's own structure (Figure 3), so that the structure will have the following form:

$$\alpha = \langle \Phi_\alpha, S_\alpha, Percept_\alpha, Mem_\alpha, Decision_\alpha, Exec_\alpha \rangle. \tag{14}$$

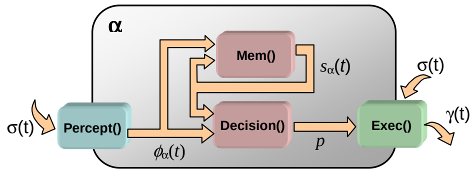

Fig. 3. Graphical representation of a hysteretic mobile agent

Thus, we regard the agent as a true PDE agent, maintaining the previously defined components and adding an execution function. This movement would be merely strategic if it did not separate the agent as far as possible from its environment. To do this, the execution function will operate on the perception the agent has of this environment, instead of operating on a world state. Formally:

$$Exec_\alpha : P \times \Phi_\alpha \to \Gamma. \tag{15}$$

This new function has the same operation mode defined in (Eq. 3). The changes introduced in the agent's structure motivate changes in the environment's structure and above all in the operation dynamics of a multi-agent system. In the following section we will study these implications in more detail.

3.3 Multiple Mobile Agent System

Considering the possibility that there is more than one agent inhabiting the world, i.e., a system based on multiple hysteretic mobile software agents, we can represent it by means of the structure

$$MMAS = \langle AG, \Sigma, \Gamma, P, React \rangle \tag{16}$$

where $AG = \{\alpha_1, \alpha_2, \ldots\}$ represents the system's set of hysteretic software agents and the rest of the elements involved are defined in the same way as in the previous sections.

According to this structure, the system's dynamics is defined by $card\,(AG) + 1$ equations in which the first equation describes the state of the environment according to the time and behavior of each agent and the remaining equations correspond to modifications in their internal state.

$$\begin{aligned}
\sigma(t+1) &= React\left(\sigma(t), \bigcup_{i=1}^{n} Exec_i\left(Decision_i\left(\phi_i(t), s_i(t)\right), \phi_i(t)\right)\right) \\
s_1(t+1) &= Mem_1\left(\phi_1(t), s_1(t)\right) \\
&\vdots \\
s_n(t+1) &= Mem_n\left(\phi_n(t), s_n(t)\right) \\
with \phi_i(t) &= Percept_i\left(\sigma(t)\right)
\end{aligned} \tag{17}$$

Figure 4 shows a graphical representation of a hysteretic MMAS made up of three agents acting in it. Now the problem consists of determining the set of internal states for each agent and describing the decision and memorization functions in such a way that the system's behavior adapts itself to the designer's perspectives and the desired collective phenomena are shown.

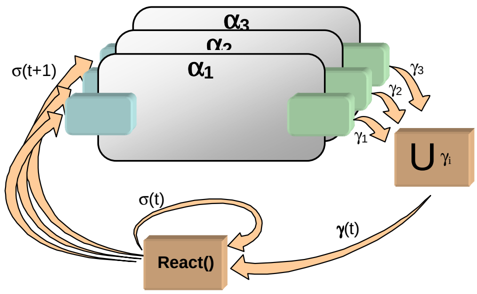

Fig. 4. Graphical representation of a MAS made up of hysteretic agents

Although the operation dynamics of the mobile software agents are given in this formulation, the inherent problems in their mobility are not solved. In the following section, we present a study of the most typical problems and propose a series of strategies to minimize them.

4 MODEL REFINEMENTS

One of the difficulties encountered when putting the formal model described above into practice is that, due to the nature of the problem, we have a specific agent incapable of contributing its influence or, because of a malfunction in one of the agents, the influence it contributes is not the right one or does not reach the world. In order to eliminate, minimize or at least mitigate these problems and systematize the procedure to define the world reaction function (Equation 5), the main one affected by these problems, there are two fundamental points we can touch upon: the set of influences itself or the world reaction function. We will discuss each of these in more detail below.

4.1 Influence Vector Extension

This approach focuses on the problems caused directly by inconsistencies in the agents' different contributions to the state of the world. Due to intrinsic aspects of the mobile agents themselves (inconsistent data), time aspects (synchronization problems) and space aspects (partial data), the consistency of the world reaction function must be questioned when faced with the impossibility of ensuring a sufficiently limited input set. In order to solve this problem, we propose the introduction of a heuristic capable of deducing the suitable influence vector from a vector with possible faults.

The dynamics of the hysteretic multi-agent system in (Equation 18) could be condensed in the following way:

$$\sigma(t+1) = React(\sigma(t), \gamma(t)). \tag{18}$$

We define a heuristic H_Γ capable of deducing a valid influence vector for the world reaction function from an influence vector provided by the system's agents:

$$H_\Gamma : \Sigma \times \Gamma \to \Gamma, \tag{19}$$

for example

$$\gamma' = H_\Gamma(\sigma, \gamma). \tag{20}$$

We can replace the original influence vector in (Equation 18) for the heuristic itself and thus manage to absorb or reduce the different problems of inconsistency, incoherence and ambiguity described above.

$$\sigma(t+1) = React(\sigma(t), H_\Gamma(\sigma(t), \gamma(t))) \tag{21}$$

Although, generally speaking, we could use any classifier as a heuristic, due to the potentially large dispersion that exists among the values for the different vectors and their corresponding states, we have opted for the use of a neural network. This type of tool adapts well to the characteristics of this type of problem as it has a high tolerance to faults and a great facility for modeling non-linear functions [9].

4.2 Reaction Function Extension

The problem here is the specification of the reaction function (Equation 5), responsible for providing the system's present state based on influences from different agents. Now, apart from the problems mentioned in the previous section, we have others that are more closely related to the actual definition of MAS and the reaction function: basically, they can be summed up in the impossibility of detaining the inference mechanism of the world states from the influence vectors. At worst, we might have failed to detect redundancies or very close relationships among the world states that may produce a process of divergence in the system, or we might not even have been able to determine the problems.

For this case, the proposal consists of replacing the world reaction function with a heuristic capable of classifying the input influence vectors on a map of optimum states so that it will subsequently be able to assign one of these states to any possible input influence vector.

5 EXPERIMENTS

In this section, we propose a domain extension for the predator-prey problem based on the Tan work [14] in which a collection of animals that we denominate predators has as objective to capture the biggest number possible of preys. This extension supposes that each node of the network has its own game (ecosystem) and that the predators can communicate to each other, independently of the ecosystem in which they are, as well as move among such ecosystems in case of being necessary. Now, to the problems of lack of information, those originating of the high costs derived from the communications among ecosystems are added, along with those originated by the transfer of predators.

Next we presented the results of several simulations in which particular situations are solved and compared with the results obtained by means of the application of our proposal of mobile agents. In general, the obtained results are very encouraging for our model (Figure 5). The percentage of captures tends to equal itself in both models as it increases the predator density. In the reference to the number of movements – local and global – both models follow similar guidelines.

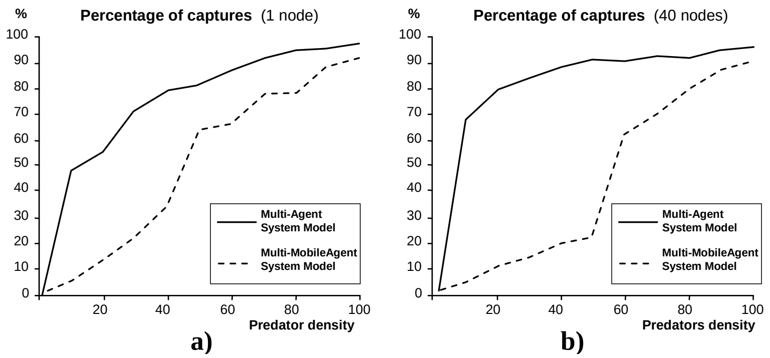

Fig. 5. Comparation of the capture percentages between the conventional model and the proposal of mobile agents applied to the predator-prey problem. a) The game is developed in only one node. b) The game is developed among forty nodes.

Whereas the costs originated by the communication among the different ecosystems practically depend on the number of nodes (Figure 6 a), the local communication increases slightly (Figure 6 b). Considering that the cost caused by the communication among nodes is, with difference, the most elevated of the problem, the model derived from our proposal presents characteristics that make it suitable for systems that must be scalables [8]. Nevertheless, in spite of the promising results, it must be borne in mind that we have chosen a problem that adjusts perfectly to our model because the predators are only able to perceive a very limited window, $s[u, v]$, of its surroundings. It should also be kept in mind that this example reflects many of the problems that are outlined in administration of distributed systems.

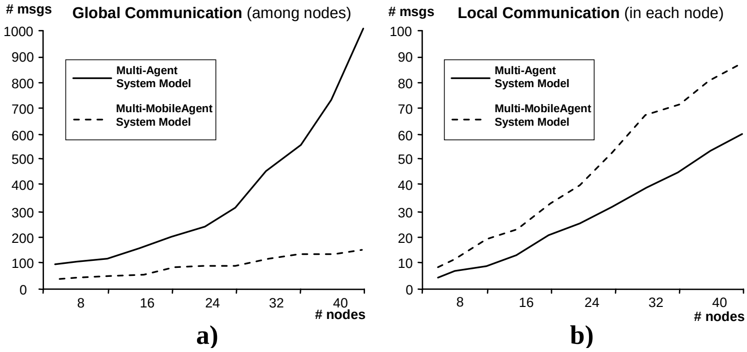

Fig. 6. a) The number of messages originated by the predators among the different ecosystems. b) The number of messages interchanged by the predators in an ecosystem.

6 CONCLUSIONS

In this paper we have presented a formal framework that enables us to cover the operation dynamics of a MAS made up of mobile software agents with the aim of providing an algebraic model that will allow to define systems with these characteristics in a more systematic and reliable way.

The proposed model presents characteristics that allow to drastically reduce the traffic of network by the necessity to constantly update the system state, with which its application in network surroundings favors its scalability. Another of the advantages offered by this formal framework is unification among the system's specification phases, its design and implementation, making the introduction of a declarative language possible.

We are currently developing auxiliary devices that will act as an agent coprocessor, reducing the additional load that this type of platform generates in the central processor. Our projects for the immediate future include the design of intelligent networking devices that incorporate this platform in a natural way, by incorporating capacity to analyze software agents at circuit level.

REFERENCES

[1] Berners-Lee, T.—Hendler, J.—Lassila, O.: The Semantic Web. Scientific American 2001.

[2] Bond, A.—Gasser, L. (eds.): Readings in Distributed Artificial Intelligence. Morgan Kaufman 1998.

[3] Ferber, J.—Mller: Conception de Systèmes Multi-Agents Par Composants Modulaires et Réseaux de Petri. In Proceedings of PRC-IA. Montpellier 1996.

[4] Ferber, J.: Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. Addison-Wesley, England 1999.

[5] FISHER, M.: Towards a Semantics for Concurrent MetateM. Modal and Temporal Logics. Springer-Verlag, Germany 1995.

[6] GENESERETH, M. R.—NILSSON, N. J.: Logical Foundations of Artificial Intelligence. Morgan Kaufman, England 1987.

[7] HOARE, C.: Communicating Sequential Processes. Prentice-Hall International 1986.

[8] LAWRENCE, G. R.: Beyond Moore's Law: Internet Growth Trends. IEEE Computer 2000.

[9] NARENDRA, K. S.: Neural Networks for Control: Theory and Practice. In Proc. IEEE, Vol. 10, 1996, pp. 1385–1397.

[10] PEDNAULT, E.: Formulating Multiagent, Dynamic-World Problems in the Classical Planning Framework. In Reasoning about Actions and Plans, Timberline, M. Georgerff and A. Lansky (eds.), Morgan Kaufman 1986, pp. 425–448.

[11] PEDNAULT, E.: ADL: Exploring the Middle Ground Between STRIPS and the Situation Calculus. In Ronald Brachman, Hector Levesque, and Raymond Reiter (eds.), Proc. of the First Int'l Conf. on Principles of Knowledge Representation and Reasoning 1989, pp. 324–332.

[12] QUENDT, B.: An Agent-Based Resource-Control in a Signalling System for the Open Telecommunication Market. In Proc. of PAAM '97, 1997.

[13] RAO—GEORGEFF: Modelling Rational Agents within a BDI-Architecture. In Proc. of Knowledge Representation and Reasoning KR & R-91, Fikes y Sandewal (eds.), Morgan Kaufmann Publishers 1991, pp. 473–484.

[14] TAN, M.: Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In Proceedings of the Tenth International Conference on Machine Learning, 1993, pp. 330–337.

[15] WALDINGER, R.: Achieving Several Goals Simultaneously. Machine Intelligence, Vol. 8, 1977, pp. 94–136.

Juan Manuel García-Chamizo received his Bachelors degree in Physics from the University of Granada (Spain) in 1980 and his Ph. D. degree in Computer Science from the University of Alicante in 1994. Between 1998 and 2004 he directed the Department of Computer Science and Technology. Currently, he is Professor of Computer Architectures and Director of the Industrial Computing and Computer Networks Research Group of the University of Alicante. He is a reviewer of several international publications and he has coordinated national projects related to computing. His research interests are focused in computer vision, computer architectures, networks and biocybernetics.

Francisco Maciá-Pérez received his B. Sc. degree in Computer Science in 1994 and the Ph. D. degree in Computer Science in 2001, both by the University of Alicante (Spain). He worked as System's Administrator at the same University until 2001. Currently, he is an Assistant Professor and Director of the Department of Computer Science and Technology at the University of Alicante. His research interests are in the area of network management, computer networks and architectures, distributed systems, biocybernetics applications; areas in those that he directs a very diligent research group.

Antonio Soriano-Payá received his Bachelors degree in Computer Science from the University Polytechnic of Valencia (Spain) in 1993 and his Ph. D. degree in Computer Science from the University of Alicante (Spain) in 2001. He is an Assistant Professor in the Department of Computer Science and Technology of the University of Alicante. He coordinates the division of medical bioengineering of the Department. His research interests are focused in biocybernetics, computer architectures and computer networks.

Daniel Ruiz-Fernández received his Bachelors degree in Computer Science from the University of Alicante (Spain) in 1998 and his Ph. D. degree in Computer Science from the same university in 2003. Between 1999 and 2004 he worked as a lecturer at the University of Alicante. Since 2004, he is an Associate Professor in the Department of Computer Science and Technology of the University of Alicante. He is a member of IEEE and Computer Society. His research interests are focused in biocybernetics, computer architectures, distributed computing and computer networks.