An Example of Dynamical Physical Agents

Josep Lluís de la Rosa*, Bianca Innocenti[1]*, Albert Oller**, Albert Figueras* Josep Antoni Ramon*, Israel Muñoz* and Miquel Montaner*

*Institut d'Informàtica i Aplicacions (IiiA)
Universitat de Girona & LEA – SICA
Av. Lluís Santaló, s/n
E-17071 Girona, Catalunya
+34 972 41 84 78
{bianca, peplluis, figueras, jar, imunoz, mmontane}@eia.udg.es

** Dept. d'Enginyeria Elèctrica, Electrònica i Automàtica
Universitat Rovira i Virgili (URV)
Autovia de Salou s/n.
E-43006 Tarragona, Catalunya
+34 977 55 97 04
aoller@etse.urv.es

ABSTRACT

This paper shows the benefits obtained when the dynamic behaviour of the agent's physical body is taken into account. The agent oriented language Agent0, highlighted the need of declaring the capacities of agents in their reasoning. An example of convoying two controlled autonomous mobile robots as agents is shown. The responsibility of avoiding collisions is for the rear agent, but the reliability of sure decisions based on dynamics is of both of them. The deliberative co-operative decisions based on dynamics provide the controllers with safer set points. Finally, some experimental results using the RoboCup real robots are shown.

Keywords

Modelling the behaviour of other agents, autonomous robots, designing agent systems.

1. INTRODUCTION

A real challenge to AI is to come up with solutions to the problems that are solved routinely by humans without any measurements or any computations in a co-operative way.

Let us consider a range of driving automation problems such as: (1) freeway driving with no traffic; (2) freeway driving with light traffic; (3) freeway driving with moderate traffic; (4) freeway driving with heavy traffic; (5) city driving in Helsinki; (6) idem in London; (7) idem in Rome; (8) idem in Istanbul.

The current developments, according to L. Zadeh's opinion, show that automation of (1) is achievable; (2) might be possible, with some qualifications; (3) is not possible today but might be in the future. Beyond (3), the problems are intractable, with no solution in sight.

This paper tries to do a step forwards approaches of higher degree of complexity than (2) by using the football robots technology of RoboCup. It contains the problems of driving or manoeuvring one car, and its non-straightforward extension to multiple cars, problems (2) to (8). The fact is that not only feedback control is necessary for solving these problems, but also the co-operative aspects of AI have to be integrated. In this paper, small robots that have clear dynamic movements will emulate the cars. The robots were developed for MIROSOT (Micro Robot Soccer Tournament) and RoboCup events from 1996 [2] and [4]. There is no lack of generality in this approach since we will stress on the co-operative decisions among autonomous mobile robots by considering the dynamics of emulated vehicles [3] and [9]. Techniques applied to Cupertino use the agent oriented analysis that has to be finally implemented on mobile robots.

This paper in section 2 introduces concepts of physical agents that pretend to represent the situation of embodying one software agent in an autonomous robot. The section 3 completes the notion of physical agent with dynamical knowledge of autonomous vehicles emulated by autonomous robots. Section 4 shows an example of the advantage of using some robot dynamics' knowledge in a case of convoying two autonomous vehicles. Finally, in section 5 some conclusions show open research on the formulation of knowledge about dynamics.

2. PHYSICAL AGENTS

Previous to the physical agents' definition, software agents will be introduced.

Definition 1: Software agents. This term denotes a software-based computer system that has several properties [13] as autonomy, social ability, reactivity, pro-activeness, mobility, rationality, etc.

Physical Agents are software agents that contain the N/S (Numerical/Symbolical) and S/N (Symbolical / Numerical) interface that is typical of real systems, which according to [1] and [8] are constrained by imprecision, uncertainty, changing through time, and others.

One typical implementation of physical agents (but not the unique) is mobile robots, that in current research are progressively more and more autonomous and co-operative. The traditional AI has focused on symbolic paradigms (toy problems) and has not

[1] Bianca Innocenti is a full time PhD student

expended time on real applications. On the other hand, robotics has focused on design and construction of hardware and its control.

For solving current problems in autonomous robots, traditional AI has evolved into perception based and multi-agent approaches. The conjunction of AI and robotics in autonomous mobile robots that solve in an emergent way, complex problems by Cupertino is important, especially when the environment continuously changes because of the movement of the physical agents.

Having a "physical body" according to [1] could be summarised as:

- Sensorial and action capacities are closely related.
- The agent's sensorial and actuation spaces have to be abstracted in the onboard resources (CPU memory, CPU speed, controllers, etc.).
- This abstraction depends on the interactions that this agent experiments in the environment.
- As a result of the abstraction, every agent has a representation model of the environment.
- The interactions among agents and environment-agents are asynchronous, complex, and in concurrency in the real world. The physical agents are continuously running in continuous time.
- On the other hand, the real physical interactions give good data for learning (software agents lack this).

Definition 2: Extension of the agent concept from software to physical agent. This definition consists of definition 1 and also a physical body.

Let us assume, for reasons of performance, that there is no single agent capable of doing a task alone. Every agent has to apply to the others for help. Then, each individual has to decide between the following problems: Which agent am I going to help to? On what terms am I committed to? The decision could be either deliberative or reactive. An interpretation of commitment [11] will be used for the agents. Agents that accept to help (are committed), have to know the implications of these commitments. In other words, whether they could do it or not.

For knowing what is possible or not to be done, *some physical knowledge has to be taken into account*. This means, physical inputs and outputs from the environment have to be mapped in the knowledge base of each agent. This is because agents have to control their physical body by means of proper physical decisions.

3. DYNAMICS OF PHYSICAL AGENTS

Previous analysis is true but now consider systems whose movements can be described by differential equations, that is, can be portrayed by its dynamics. Then the automatic control theory has something to assert.

Definition 3: Extension of the agent concept from physical agent to dynamical physical agent. Complementing Definition 2 with the following new assertion: The knowledge is obtained from dynamics of the physical body, which is represented by a further declarative control and supervision levels [4] [9].

Let us show this new knowledge through the following example that explains the utility of inter-agent negotiation in terms of dynamic behaviour and that improves deceleration decisions. Steps will be as follows:

1. To model and simulate two autonomous vehicles.
2. To implement convoying by a distance controller in the rear vehicle that follows the reference of the first one. The specification is to keep steady a safer distance.
3. To see a conflict: decelerating to 0 m/s. Collisions!
4. To implement a negotiation procedure by means of dynamics included within the capacities inspired from Agent0 language [11].
5. And to avoid collisions using this approach.

A problem arises in the two vehicles convoying when they try to maintain a constant distance between them. The distance oscillates when the vehicles' dynamics are different, even though their static behaviour is equally achievable. As a result, the possibility of collisions increases accordingly to the changes of the guiding vehicle's speed. The responsibility of keeping this distance constant falls on the rear vehicle, which reaches this goal modifying its dynamical behaviour. Since sometimes the vehicles cannot avoid colliding, the up to day solution is to keep some minimum static distance. This facilitates steady state control solutions.

Decisions of changing the convoying speed are not proper under some conditions, although the control of the vehicles has been well designed. The situation is that the guiding vehicle commonly decides to accelerate or decelerate without considering the dynamics of the other vehicle. So, the creation of decision-making procedures that take into account these dynamics is proposed. This brings the systems to co-operate in taking co-ordinate decisions.

The procedure of the autonomous and co-operative decisions is defined by using an agent paradigm. In the agents' state of the art programming languages, the Agent0 language by Shoham [11] is chosen, since the dynamic knowledge of the system could be included within the *capacities* of agents. A capacity is defined as follows:

Capacity: $CAN^{t}_{a}(\varphi)$,

is such that in time **t** agent **a** is able to do φ.

Example: CAN^{5}_{robot} go_to $(0m/s)^{8}$

The robot knows at time 5 that it can decelerate to 0m/s at time 8. This is the same as saying the robot can execute in its current state a trajectory towards the final state in three sample instants.

An immediate version of CAN is ABLE, for example:

$ABLE_{robot}$ go_to $(0m/s)^{5}$ = CAN^{5}_{robot} go_to $(0m/s)^{5}$:

The robot knows at time 5 that it can open decelerate to 0m/s at the same time 5 (now).

The immediate question is how to design and create the base of capacities CAN/ABLE. This paper proposes a first proposal by assigning some determinant information about dynamics of the physical body of agents.

The big interest is to see how these physical knowledge of the dynamical capabilities affect the co-operative behaviour of the whole co-operative world, as [7].

3.1 Modelling and simulation of two autonomous vehicles convoy

There are two types of behaviour for the agents: reactive and deliberative. In the first type the guiding agent (first in the convoy and hereafter agent A) decides to decelerate or accelerate without taking into account the dynamics of the rear vehicle (rear in the convoy, and hereafter agent B). However, agent B is responsible for not colliding. In the second type, agent A modifies, if it is possible, its own dynamics based on agent's B dynamics until they agree on the decelerating time; and again agent B has to keep constant the distance between them.

In the deliberative behaviour, the agent A communicates to the agent B its decision of decelerating at a given time. Agent B simulates its behaviour and answers agent A with its certainty associated to this action. This coefficient lets agent A to decide about the action. If there is no agreement, then agent A proposes to do the same action but in a different execution time, what is reached modifying its dynamic behaviour with a new controller. These steps are repeated until to obtain an agreement.

Transfer functions are used for analysing dynamics. In this first approach, only very ideal systems will be analysed, then first order transfer functions are the proper way to represent dynamics in linear speed (one-dimensional movements). Other higher order transfer functions, non-linearities and other variables (like angular orientation, etc.) will be analysed in future work.

The system closed loop transfer function F(s) is

$$F(s) = \frac{V(s)}{V_{ref}(s)} = \frac{K}{\tau s + 1}$$

Where $V_{ref}(s)$ is a speed step input, K is the static gain, and τ the time constant.

To start easy, agents have proportional controllers that change the position of the single pole of the closed loop system transfer function.

3.2 About controllers and dynamics

Let us start with the simplest feedback solution. The agent B modifies its speed set point proportionally to the separation distance to keep it constant. The time to decelerate t_d is defined to be at most three times the time constant τ of the agent A. This time will be used as a first approach to represent the certainty of executing actions. Agent A will use this coefficient to accept or discard decisions. The certainty coefficient *e* is initially defined as follows:

$$e = 1 - \frac{V_{end} - V_{td}}{V_{end} - V_{ini}}$$

Where:

V_{end} = final speed

V_{ini} = initial speed

V_{td} = speed at the time t_d

Figure 1 shows the temporal response of both systems to a step of 1 m/s. If t_d is chosen as $t_d >> \tau_A$, where τ_A is time constant of agent A, this value *e* will be close to 1 and the action will be always certain. With lower values of t_d , *e* will lead to lower certainties associated to actions.

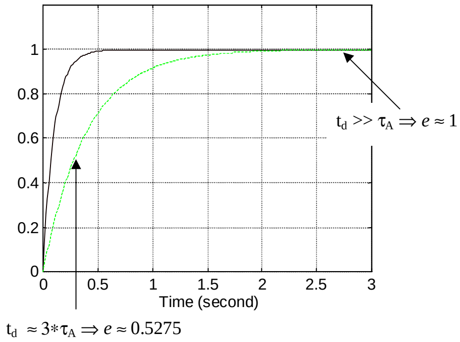

Figure 1: Estimation of certainty according to the time constant of the system, based on the steady state error measure.

Figure 2 shows the whole system in blocks diagram, where $G_A(s)$ is the closed loop transfer function of agent A, $G_B(s)$ is the same for agent B, X_A and X_B linear distance of A and B from the origin, and D the distance to keep between the agents.

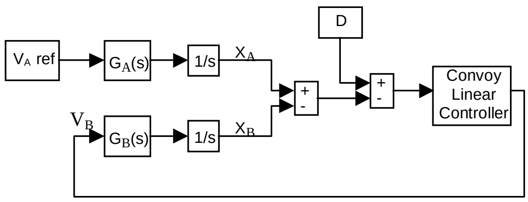

Figure 2: Whole system block diagram.

From the linear point of view, dynamics of the controlled convoying system is as follows: the agent A has its internal control that is apparently independent of agent B. Agent B has inherently more complicated structure since, so far, the responsibility for convoying and avoiding risky situations (e.g. collisions) is up to it. Assuming that agent A dynamics shows a first order closed loop transfer function $G_A(s)$, and that agent B has the same closed loop behaviour $G_B(s)$, then the convoying behaviour of the agent B is *indeed* a second order transfer function. This is due to the dependency of speed set points of agent B on the position controller, as seen in Figure 2. In other words, the variable to be controlled is the set point of speed V_B.

Up to these days there are obvious static specifications to take into account, i.e., to maintain regular distance D between the two autonomous systems $D - (X_A\text{-}X_B) \cong 0$. This can be easily accomplished by the use of a proportional K_P controller (Convoy Linear Controller), that is to say, using the value $K_P \cdot (D - (X_A\text{-}X_B))$ as the signal control V_B. Under these assumptions, if $G_A(s)$ and G_B (s) are first order systems, then $D - (X_A\text{-}X_B)$ *is* a second order system response.

Suppose initial conditions D=5m, $V_A\ (t=0)$=8 m/s and $V_B\ (t=0)$=8 m/s, X_A=5m. (The vehicles keep the safety distance D at 8 m/s regime). Then vehicle A changes the speed to V_A=2 m/s. The result is clearly, a 2nd order behaviour of convoying B, as depicted in the following plot:

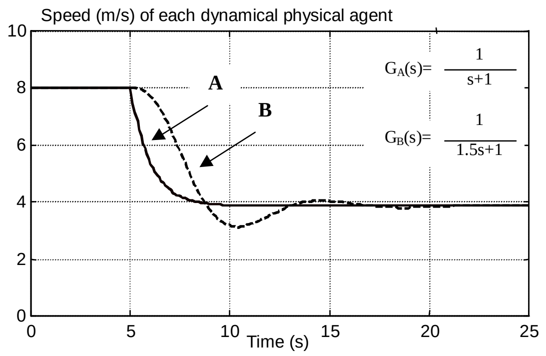

Figure 3: Speed response of mobiles A and B.

This paper tries to point out the fact that in the previous analysis there is no discussion about the possibility of collision between the two mobiles. How can this occurrence be established as a control specification? Next figure shows the evolution of distance between the systems and, as can be seen, at t=6s there is a collision.

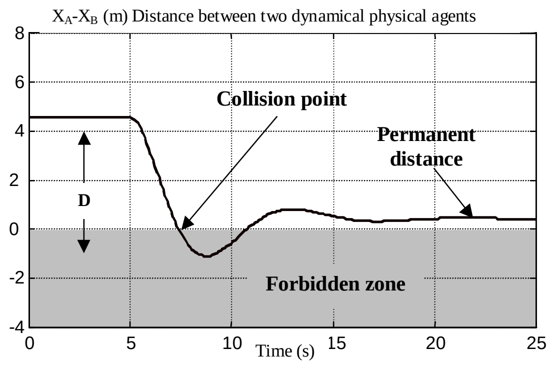

Figure 4: Lower limit of X_A-X_B = *D*.

Due to this 2nd order nature of the convoying system, it can be clearly asserted that static information for deceleration decision is not enough at all. Then, decision based only on static is dangerous in case of dealing with dynamical physical agents.

4. CO-OPERATION MADE BY MEANS OF KNOWLEDGE ABOUT DYNAMICS.

Let us see the theoretic of the previous section with the following real scenario. Two mobile robots (hereafter agents) will describe a rectilinear movement, and in the beginning a security distance separates them. The relative position between the agents is obtained by a camera, which has a global vision of the convoy. For this reason, the distance that the robots can go over is limited

Both agents have two Proportional-Integral (PI) controllers to change their behaviours, different from the previous section, because it was not possible to reach the set point only with a Proportional controller. In the same way, there is a PI controller to keep constant the distance between them. The controllers, as the agents, are located in a host computer. The communication between the agents is made in the main program. A radio transmitter is used to send the calculated set points to the robots.

The G_A(s) block of Figure 2, here is the first robot of the convoy and its controllers. Likewise, G_B(s) is the second robot of the convoy and its controllers. Each robot is a closed loop dynamical system governed by an agent who takes decisions.

The proposal is to include the dynamics of the robots within the capacities (Agent0 language) of the agents. One important reason for doing it is that it is possible to work without knowing the transfer function of the system. In the particular case, and referring to section 3, is not possible to simulate the behaviour of the robots due to the higher time consumed by this operation (total distance is limited by the camera vision field). Consequently and as a first approach, capacities are proposed to be an ordered set with the following information:

< initial_state, final_state, specified_time_for_action, certainty, controller_parameters >

These capacities will be contained in the base of capacities, together with other type of them (social, static, etc.) defined by Agent0.

The variable *specified_time_for_action* t_d is any time considered from the beginning of the action to any time before the system reaches the steady state. The *certainty* is a coefficient obtained by observation, from the dynamics of the system and it is completely dependent of t_d. The *initial_state* e_i is the initial speed; the *final_state* e_f is the final speed; and the *controller_parameters* are the proportional and the integral constants of the PI controller.

The decision is here based on the certainty. The way to take decision is a classical implementation of expert systems that fires the rules that overcome a pre-specified threshold. The agent A determines whether any action has to be done or not. The innovation is that the decision is also based on the dynamics of agent B. For agent B, an action is possible if it can pass from the *initial_state* to *final_state* within the *specfied_time_for_action* time interval that the agent A proposes.

Once both agents agree, that is, the certainty about the decision is high enough, they fire their respective decisions. This negotiation algorithm is as follows:

Negotiation Algorithm

1. Agent A decides to change the speed and sends a request message to Agent B as follows:

request (agent_A, agent_B, initial_speed, final_speed, time);

2. Agent B receives the message and looks in it base of capacities, for one that fulfil the requirements.

IF CAPACITY exist

THEN

inform (Agent_B, Agent_A, final_speed, certainty);

ELSE

inform (Agent_B, Agent_A, final_speed, 0.0);

END

3. Agent A receives the answer:

IF certainty > threshold

THEN

inform (Agent_A, Agent_B, DO_ACTION);

Agent B starts to do the action.

do (agent_A, final_speed, controller_parameters);

ELSE

4. Agent A looks in its base of capacities for one where the final and initial speeds match with the requirements, but using another controller, so the specified time changes.

IF CAPACITY exist

THEN

request (Agent_A, Agent_B, final_speed,time);

5. The algorithm goes to step 2.

ELSE

inform (Agent_B, Agent_A,DO_NOT_ACTION);

6. End of the negotiation.

END

4.1 CAPACITIES

As stated, the capacities are a set of parameters that contains information about the dynamics of the systems. Both robots have a very similar behaviour, so agents have the same controllers and the same base of capacities, shown in Table 1.

Initial Speed cm/s	Final Speed cm/s	Specified time t_d s	Certainty	Proportional constant	Integral constant
20	**0**	**1.6**	**0.96**	**0.45**	**1.0**
20	0	9.4	0.91	0.45	0.3
25	**0**	**1.55**	**0.93**	**0.45**	**1.0**
25	0	8.5	0.93	0.45	0.3
30	**0**	**1.5**	**0.92**	**0.45**	**1.0**
30	0	7.5	0.96	0.45	0.3
35	**0**	**1.45**	**0.91**	**0.45**	**1.0**
35	0	6.0	0.99	0.45	0.3

Table 1: list of capabilities

The time t_d and the certainty associated to it have been obtained applying the different set points to the robots and using the camera to get data.

For the following experiments, agent B has always the PI controller with the integral constant equal to 0.3. Agent A has initially the PI controller with the integral constant equal to 1.0, so agent B is slower than agent A is.

Negotiation in the deliberative case is as follows: agent A requests agent B to decelerate to 0 cm/s in the time t_d corresponding to the different set points. Agent B looks in the base of capacities the t_d necessary to do the action with the current controller. As this time is not the same, agent B responds to agent A that the certainty of decelerating at this time is zero. Then agent A looks in its base, for a new t_d. When it finds it, it requests to agent B the same action but with a different time proposal. Agent B searches again in its base of capacities and answers A with a new certainty. If this coefficient is greater than 0.8, agents A and B agree on decelerating. As a result of the negotiation, agent A has to change the parameters of its PI controller.

Besides these controllers, there is another PI controller, used to keep constant the distance between the agents, that changes the set points of agent B (Convoy Linear Controller block in Figure 2).

In the beginning of the experiments both vehicles are stopped. When the program starts, they begin the acceleration. Five seconds later, agent A decides to stop.

4.2 CASE 1: INITIAL SPEED OF 20 CM/S, DISTANCE 10 CM.

Figure 5 shows the separation distance between the agents for reactive behaviour (dotted line) and deliberative behaviour (continuous line). As it can be seen, both curves are similar until 5 seconds, time when Agent A decides to stop. Few seconds after, the reactive agents have a collision, while in the deliberative case, the distance decreases but never is zero.

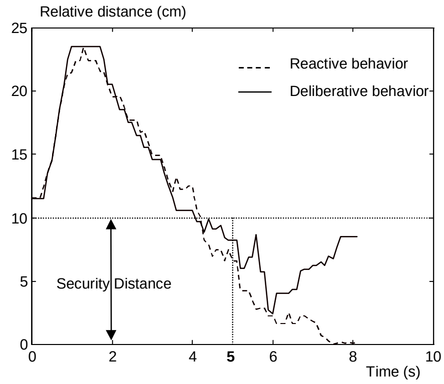

Figure 5: Distance between the agents.

The increase of distance at the beginning is because the agents are stopped when the program starts to run. The distance between the agents never gets the desired set point because the camera limits and more time is needed to reach the steady state. This also makes agent B not to get the desired set point of speed as shown in Figures 6 and 7.

Figure 6 and Figure 7 compare the speed of the agents. In both pictures, agent A reaches the steady state (20 cm/s) but agent B not. This is because of the distance controller (the 10-cm separation distance is not achieved). In Figure 7 a pick can be seen in the speed of the agents. This is due to the change of the controller in agent A.

The same experiments are repeated for other cases with several speeds and analogous results.

As result in this section 4, the deliberative behaviour of both two agents leads to better control of distance D. This is the main advantage of distributing the response of the convoying. This idea is extendable to more than two physical agents. The experiments are done in several speed setpoints to test the generality of the results. Additionally, the capacities have been used as a reasonable alternative to transfer function representation of dynamical systems, because to have such type of models (transfer functions) often is very difficult.

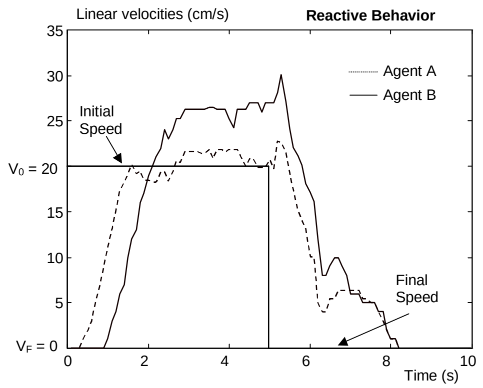

Figure 6: Speed of the agents for the reactive case.

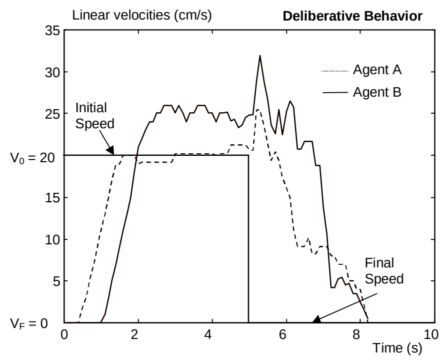

Figure 7: Speed of the agents for the deliberative case.

5. CONCLUSIONS

This work shows that a joint decision based on not only knowledge about statics but also about dynamics of the physical body that will drive to better co-ordinated control. This is because of the application of safer decisions concerning dynamical behaviour of the open or closed loop physical body and other dynamics-related information. As a strong result, some collisions could be avoided.

Capacities seem to be the best way to represent the knowledge about the dynamics of a system without having its transfer function. But it is still difficult to choose the necessary information to include in the capacities.

Finally, there are open researches in how to take advantage at the co-operative level, planning, learning, etc, of this physical introspection. Furthermore, to select one paradigm for implementation of these concepts is not trivial at all, and the application of the agent oriented programming is still open.

6. ACKNOWLEDGEMENT

This work is funded by projects TAP98-0955-C03-02 “Diseño de agentes físicos (DAFNE) / Physical Agents Design” and TAP99-1354-E, "Eurobot-Iberia: red temática en agentes físicos / Thematic Network on physical agents" of the Spanish Research Foundation CICYT.

7. REFERENCES

[1] Asada M., Kuniyoshi Y., et al. *The RoboCup Physical Agent Challenge.* First RoboCup Workshop in the XV IJCAI-97 International Joint Conference on Artificial Intelligence, pp.51-56, 1997.

[2] de la Rosa J. Ll., Oller A., et al., *Soccer Team based on Agent-Oriented Programming, Robotics and Autonomous Systems.* Ed. Elsevier, Vol 21, pp. 167-176, October 1997.

[3] de la Rosa J. Ll., García R., Innocenti B., et al., *Rogi Team Real: Research on Physical Agents.* 3rd Workshop on RoboCup, 16th IJCAI, July 31-August 6, 1999.

[4] Kitano H., Veloso M., et al., *The RoboCup Synthetic Agent Challenge 97.* XV IJCAI-97 International Joint Conference on Artificial Intelligence, Vol 1, pp.24-29, Nagoya, August 1997.

[5] Jennings N., Sycara K., and Wooldridge M. *A Roadmap of Agent Research and Development*. Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers.Vol. 1, n. 1, pp. 7-38, 1998.

[6] Kitano H. and Hondler J. A., *Massively Parallel Artificial Intelligence*. The AAAI Press / The MIT Press pp: 1-52, 1994 .

[7] Mackworth A., The Dynamics of Intelligence: Constraint-Satisfying Hybrid Systems for Perceptual Agents. AAAI Spring Symposium in Hybrid Systems and AI, 1999

[8] Müller J., *The Design of Intelligent Agents: a layered approach.* Lecture Notes in Artificial Intelligence, Vol. 1177, Ed. Springer Verlag, 1996.

[9] Oller A, de la Rosa J.Ll., and del Acebo E., *DPA2: Architecture for Co-operative Dynamical Physical Agents,* MAMAAW ’99, Valencia June 1999

[10] Oller A., J.Ll. de la Rosa, B. Innocenti. Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'99). pp. 311-314. 1999.

[11] Shoham. *Agent-oriented programming,* Artificial Intelligence, vol. 60 (1), pp.51-92, 1993 & Technical Report STAN-CS-1335-90, Computer Science Department, Stanford University, Stanford, CA, 1990.

[12] Steels L., *The Origins of Syntax in Visually Grounded Robotic Agents.* The 15 IJCAI Proceedings, pp: 1632-1641, 1997.

[13] Wooldridge M., Jennings N. *Intelligent Agents: Theory and Practice.* Knowledge Engineering Review, October 1994.