REACTIVE MULTI-AGENT APPROACHES FOR THE CONTROL OF MOBILE ROBOTS

Olivier Simonin

Laboratoire Systèmes et Transports (SeT)
Université de Technologie de Belfort-Montbéliard,
90010 Belfort cedex, FRANCE
http://set.utbm.fr/membres/simonin

Abstract: This paper deals with the control of autonomous mobile robots that have to cooperate to achieve common tasks. The reactive multi-agent approach, which relies on the interaction of simple autonomous entities, is well suited to the control of distributed systems. This approach is presented and its advantages and drawbacks for robotics applications are discussed. Then, two reactive based multi-robot architectures, which extend the reactive multi-agent approach, are presented. The first one relies on the cooperation between heterogeneous reactive robots. The second one consists in extending the reactive architecture by introducing direct communications between agents. Formal specification and verification of these architectures are then discussed.

Keywords: Autonomous mobile robots, reactive agents, cooperation, swarm intelligence

1. INTRODUCTION

This paper presents the Reactive Multi-Agent Systems (Reactive MAS) approach and how it can be used and extended to the control of autonomous mobile robots. The MAS paradigm appeared in the end of 80's in computer science as an extension of both the object oriented programming and the distribution of computation in networks (Ferber, 1999).

Multi-agent Systems are composed of autonomous entities, called agents, that interact in a common environment. They have to perform some tasks without any supervised control (in a cooperative or competitive fashion). The development of MAS is mainly due to its interactions with different scientific domains, in particular with biology. Biology, and especially ethology, inspired the first reactive architectures and several distributed algorithms (Drogoul and Ferber, 1992) (Ferber, 1999). Reactive agents present the features of living entities such as real time response, robustness and ability to exploit the environment. This biological inspiration had also a strong impact on agents organization/cooperation. Indeed, the concept of swarm intelligence (or collective intelligence) relies on self-organizing of agents using their environment as a common memory. Such processes have been observed and analyzed in social insects (Deneubourg *et al.*, 1991) (Bonabeau and Theraulaz, 1994) (Beckers *et al.*, 1994) (Parunak, 1997).

In the same time, the development of mobile autonomous robots was influenced by this reactive and bio-inspired approach (Brooks, 1986) (Arkin, 1987) (Mataric, 1995). During the 90's, several reactive agent architectures were proposed to deal with mobile robotics tasks such as navigation, exploration, box-pushing, foraging (Beni and Wang, 1989) (Kube and Zhang, 1992) (Mataric,

1997) (Drogoul *et al.*, 1998) (Arkin, 1998). However, today, it is still difficult to develop reactive-based and bio-inspired robotics systems for industrial and/or critical applications. In this paper, these different reactive concepts are presented and analyzed from the robotics point of view. Some recent models, extending the reactive approach, are presented to discuss their reuse and their software implementations.

The paper is organized as follows: Section 2 presents the reactive multi-agent approach and analyzes its interest for robotics applications. Section 3 presents a first extension of the approach by defining a cooperative architecture between heterogeneous robots. In section 4, a second extension is presented, which consists to introduce direct communications between reactive agents (called the satisfaction-altruism model). The OZS formalisms is then introduced to deal with specification and reuse of such architectures. Section 5 concludes by drawing open issues in reactive based architecture for robotics.

2. REACTIVE MULTI-AGENT SYSTEMS

2.1 Reactive architecture

This paper considers the so-called "reactive" agents/robots, which react immediately to the sensed information thanks to low-resolution sensors and to the limited number of the possible elementary actions. At the opposite, deliberative agents use high level communications, plans, sensors and representation of the environment, see for instance (Parker, 2000) (Alami, 2005), and (Nana, 2005) for a general presentation of such architectures.

Reactive architectures rely on the stimulus - response principle. It consists in rules defining the actions/behaviors which can be released following the perceived state of the environment. Then the action selection depends on a priority between actions/behaviors. It may be (i) pre-defined as in the subsumption architecture (Brooks, 1986), (ii) dependent of the stimuli strength as in (Arkin, 1987) (Maes, 1991) (iii) dynamically self-adapted by the agent following the environment evolution (Drogoul and Ferber, 1992).

One basic approach to define individual reactive behaviors consists in computing artificial potential fields (APF) from sensor information. Thanks to its capability to act in continuous domains in real-time, this approach has gained popularity during the past decade in the field of autonomous robots and especially in robot motion planning (Barraquand *et al.*, 1992). By assigning repulsive force fields to obstacles and an attractive force field to the desired destination, a robot can follow a collision-free path via the computation of a motion vector from the superposed force fields (Arkin, 1987) (Balch and Arkin, 1995) (Simonin, 2005). Such a technique is extended for cooperation and conflict solving in section 4.

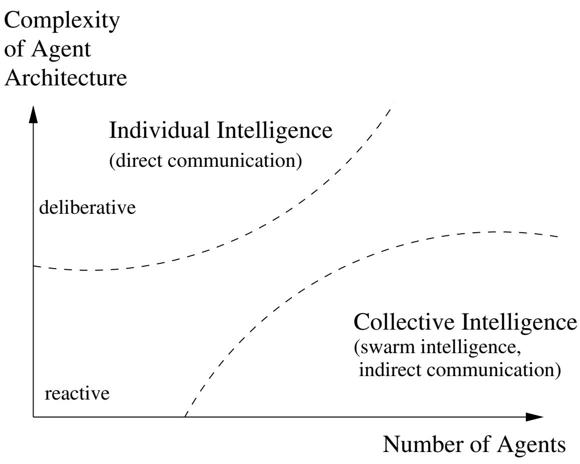

Fig. 1. Agent Architecture and Complexity

Before examining processes of cooperation and communication between reactive agents, we can abstract the main advantages of reactive architectures:

- simplicity (of the individual architecture), implying limited software size and fault tolerant systems
- the approach is relevant for exploration and navigation tasks in unknown and dynamic environments (Nana, 2005).
- low cost for robotic design (few memory, no high level communications, simple sensors, few embedded intelligence)

2.2 Swarm Intelligence

Self-organization exists in many natural systems and especially in insect societies. Such systems are composed of numerous reactive entities that interact in a common environment to solve a global problem. For instance, ants can build tri-dimensional structures or solve complex problems without any global control (Parunak, 1997). Their organization results from the numerous interactions between agents and their environment. This last guides the agent behaviors and the whole system organization (called stigmergy principle) (Parunak and Brueckner, 2001). As shown in fig 1 collective intelligence can emerge only if a sufficient number of agents interact, it is the notion of critical mass. Indeed, reactive agents have no direct communication, their cooperation is based on the interaction with their environment (cf.

(Brooks, 1986) (Arkin, 1998)). For instance, ants drop pheromones to collectively build paths between discovered resources and their nest .

Such an approach has been used to define decentralized algorithms to deal with path finding problems (ant algorithm (Colorni *et al.*, 1991)), collective tasks (such as box-pushing (Kube and Zhang, 1992) (Kube *et al.*, 2005), navigation (Arkin, 1992), foraging with robots), etc. Most of these works are based either on digital pheromones (as inspired by ants) or on artificial potential fields (APF).

The main properties of the swarm approach are:

- robustness in case of agent failures (due to redundancy of agents and their non supervision)
- adaptation to system perturbations / dynamical environments
- self-organization: emergence of solutions facing perturbations

2.3 Interest and drawbacks for application in robotics

Reactive architecture and swarm organization seem well suited to the design of cooperative autonomous robots. As presented in previous section, these models are relevant for cooperative tasks such as spatial coordination. However, considering industrial applications, some drawbacks must be reported:

Drawbacks of reactive architecture:

- strong dependence to perception (quality and nature of percepts)
- sensors perturbations due to environmental conditions (changes)
- internal parameters such as weights for actions selection may be difficult to define (can need a learning process)

Drawbacks of the swarm approach:

- no guarantee of task achievement in a reasonable time (the system can explore a long time the state space before reaching the goal)
- emergence is naturally difficult to exploit/ control
- numerous robots leads to a global high cost even the individual one is low
- few methodologies to design such solution (recently (Simonin and Gechter, 2006) propose an environment-based methodology).
- local techniques, such as the APF one, are limited by local minima, e.g. adding attractive and repulsive fields can produce areas where forces are equilibrated, then agents can be trapped in such places (Barraquand *et al.*, 1992).

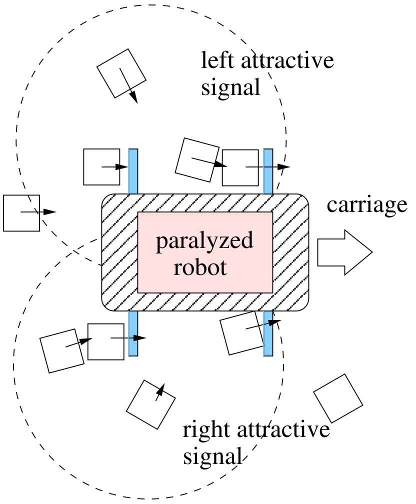

Fig. 2. Principle of the cooperative task

The reactive approach proposes interesting properties for robotics (robustness, adaptability..) but they appear difficult to be controlled. At the opposite, deliberative architectures provide methodological tools and well known technological means. However, if the robots environment is unknown, dynamic and/or hostile, such an approach becomes unsuitable (weak, low.. (Nana, 2005)). Today, the challenge for computer scientists and automatic specialists is to define solutions that combine the both approaches. These ideal architectures can be situated in the middle of the diagram presented in fig. 1, i.e. in the empty area between intelligence and collective intelligence.

Some hybrid architectures have been proposed, but they are generally organized as a deliberative layer dominating a reactive one dedicated to navigation (Connell, 1992) (Arkin and Balch, 1997) (following the so-called tree-layer architectures (Gat, 1998)) . In next sections, two other hybrid approaches are explored. The first one studies the possibility to define a robotic architecture as the cooperation of heterogeneous reactive agents. The second hybrid solution consists in introducing deliberative concepts inside the reactive principle.

3. COOPERATION BETWEEN REACTIVE AND HETEROGENEOUS ROBOTS

3.1 Presentation of the cooperative architecture

In hostile environments and/or under extreme conditions, robots failures may occur. In this case, robots must be able to cooperate in order to help themselves. This section presents an architecture allowing cooperation between autonomous and heterogeneous robots. We assume that robots can communicate only thanks to signal emissions (i.e. local broadcast of simple information) and their

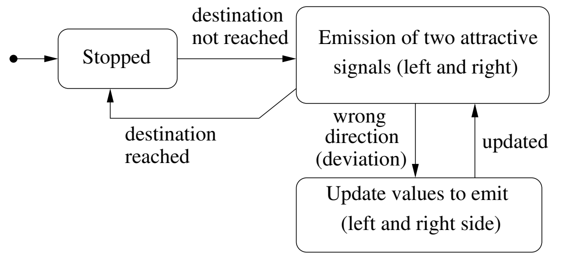

Fig. 3. Behavior of the paralyzed agent (placed on a carriage)

perception are limited to their local environment (within a limited range).

Figure 2 presents the studied system. First, a "paralyzed" robot is placed on a carriage to facilitate its displacement (such a disability can be due to a failure). Second, a set of reactive-based mobile robots evolve in this environment. These robots are just able to perceive close obstacles and to react to signals. The objective for the paralyzed robot is to recruit these simple robots in order they push its carriage and move it on a desired path. To achieve such a goal, we assume that it can emit attractive signals on its both sides, called left and right signals. Each of them covers a limited circular area (see fig. 2). In contrast to the mobile robots, the paralyzed one owns a vision that allows it to determine the path it wants to follow (detection of a white line in experiments, cf. fig. 5). As this robot and its carriage are considered heavy, the displacement can only be obtained if several mobile robots simultaneously push it.

So, this system presents two types of cooperation. An indirect cooperation between the mobile robots that perform the pushing action and a direct one, i.e. using communication, between the paralyzed robot and the mobile ones. The whole system can be seen as a box-pushing task where the box is a robot asking to move along a specific path.

3.2 Robots and behaviors definition

3.2.1. The paralyzed robot As it cannot move, the behavior of this agent is limited to the request of help through the emission of signals. However, to be pushed in a particular direction, the agent can emit differentiate requests on its both sides. The intensity of signals depends on the rotation needed by the agent to move towards the desired direction. For each side, the signal intensity is directly proportional to the direction deviation (see (Simonin and Grunder, 2006) for the numerical model). Consequently, signals intensity changes dynamically along the displacement. This agent's behavior is presented in figure 3.

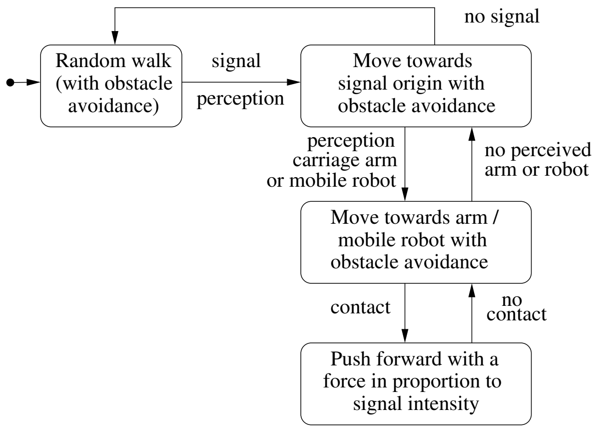

Fig. 4. Reactive behavior of the pushing agents

3.2.2. The pushing robots These reactive robots explore the environment until they perceive attractive signals emitted by the paralyzed one. Their behavior consists then to move closer to the origin of the signal. As a consequence they will arrive near the carriage and may collide it in order to apply a pushing force. To perform these behaviors a pure reactive approach is used, which is based on the combination of influences (or forces). At each time, an agent has to compute two vectors: (i) a goal direction deduced from the signal reception (ii) an obstacle avoidance direction. Agent motion is based on the combination of these two forces. To allow the agents to push on the arms of the carriage, they are equipped with sensors to detect them. This whole behavior is presented in fig. 4.

3.3 Experiments

To experiment the proposed architecture, two different types of robot have been chosen. The paralyzed robot is a Lego Mindstorms. To play the role of pushing robots, Mirosot robot soccer are used (Mirosot, 2006). More details on these robots are presented in (Simonin and Grunder, 2006).

The pushing process around the carriage is composed of different phases. For more clarity, snapshots from a movie of an experiment involving 3 mobile robots and a Mindstorms on the carriage are presented (fig. 5 to fig. 8).

The first step consists in the positioning phase. Robots perform a random walk until they perceive attractive signals (it is the case for the 3 robots in fig. 5). Then, they start to move towards the carriage. An obstacle avoidance procedure is used to ensure collision avoidance between mobile robots. While approaching the carriage, the robots are braked due to the perception of the carriage

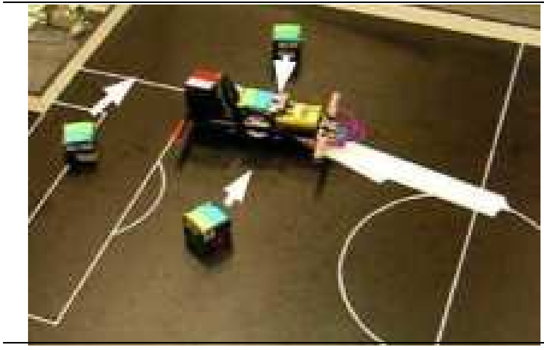

Fig. 5. Placement phase : the Mindstorms attracts the soccer robots

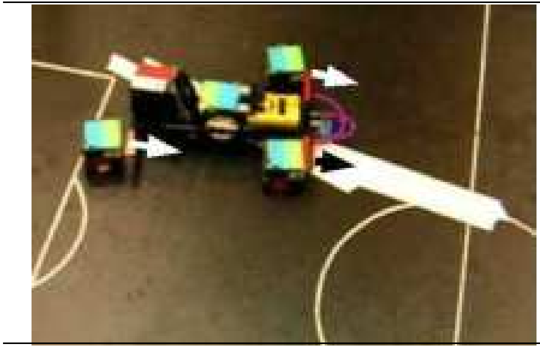

Fig. 6. Pushing phase : the 3 robots push on the arms of the carriage

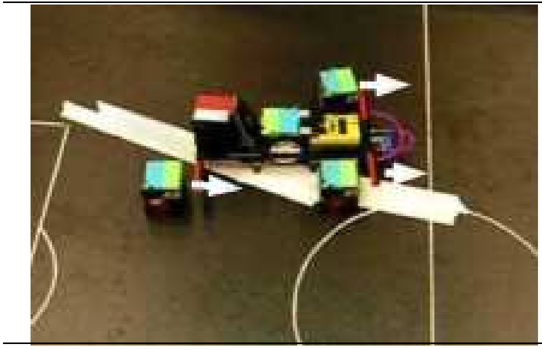

Fig. 7. The carriage takes a wrong way

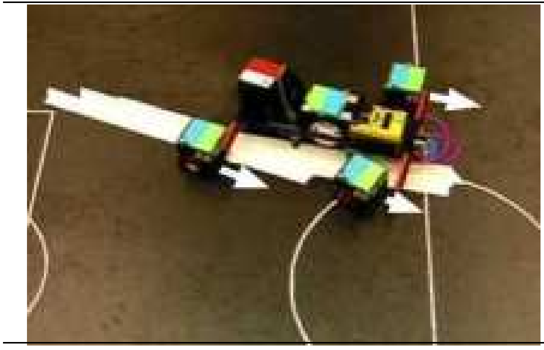

Fig. 8. The carriage has been rotated due to the force applied on its left side

as an obstacle. But the combination of attractive and sliding forces tends to align the robots along the carriage. Moreover when agents detect an arm they change their behavior to simply go towards it (see fig. 6). They start then to push the carriage.

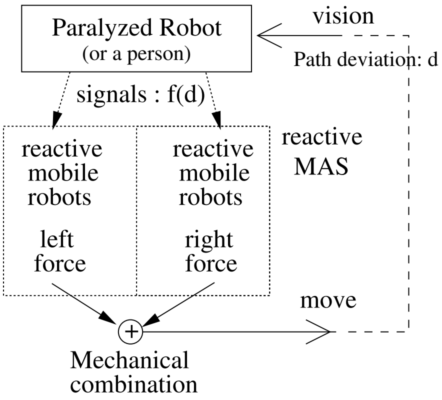

Fig. 9. The reactive MAS becomes an actuator of the paralyzed agent

The second step, for mobile robots, consists to apply a force in proportion to the perceived signals. At the same time, the paralyzed robot has to evaluate its displacement to compute its signals for both sides of the pushing.

Figure 7 shows that the 3 robots have pushed the carriage but its trajectory veer off to the left side of the white line. As a consequence, the Mindstorms robot increases the intensity on its left signal and decreases the right one. Thus, the mobile robot on the left side increases its applied force that rotates the carriage while moving it. When the Mindstorms perceives that its direction is the same as the one of the white line, it sends left and right signals of equal intensity.

3.4 Analysis

Several experiments have shown that the proposed architecture is efficient to perform a cooperative box-pushing task. There exists a first direct cooperation between the mobile robots and the paralysed one (using signals emission). A second type of cooperation appears between mobile robots themselves. They have to simultaneously push the carriage to perform its move. The different forces are naturally combined, defining a collective action. Mindstorm signals ensure that these forces are well distributed between the two sides of the carriage.

Even if the Mindstorms robot and the mobile ones are independent agents, there exists a control loop linking them (represented in fig. 9). Indeed, to move the carriage in the good direction, the paralyzed robot always checks its direction to update the signals it emits. These emitted values depend only on the difference between the current direction of the carriage and the desired direction. They are not dependent on the number of agents pushing on the left and right side of the carriage.

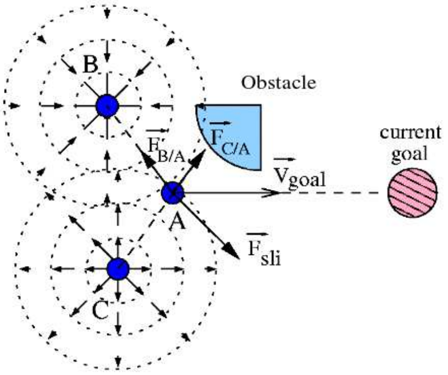

Fig. 10. Signals emitted by agents

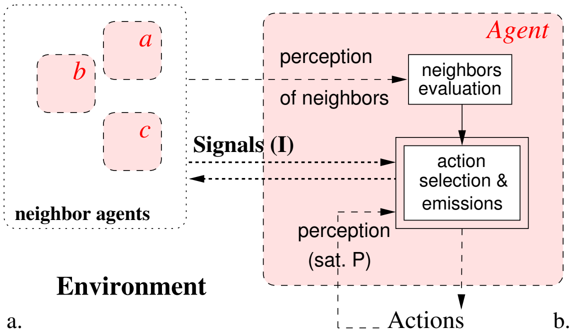

Fig. 11. The Satisfaction-Altruism architecture

Thus the task can be achieved whatever the position and the number of agents involved. Finally, the mobile robots are used by the paralyzed one as a robust actuator allowing it to move.

This cooperative architecture has shown that reactive behaviors/agents can be combined to provide robust and intelligent systems. It is defined as an open multi-agent system, i.e. a system that does not depend on the number of mobile robots (or its variation). Moreover, we can imagine to place different kind of agent on the carriage, such as a deliberative robot or even a human that should be evacuated.

The next section presents an extension of the reactive approach that allows to deal with cooperation in constrained environment.

4. EXTENSION OF THE REACTIVE APPROACH

4.1 The Satisfaction-Altruism Model

The Satisfaction-Altruism model aims at providing means of cooperation and conflict solving to reactive-based agents working in the same environment (Simonin, 2001) (Simonin and Ferber, 2000). This model extends the reactive MAS approach by introducing direct and intentional communications between agents.

The satisfaction-altruism model relies on the extension of the artificial potential fields (APF) approach. The model introduces *new artificial fields* in the environment(Simonin and Ferber, 2000). These fields are *dynamically* and intentionally generated by agents thanks to the emission of attractive and repulsive *signals*. Agents broadcast such signals in order to influence their close neighbours. These artificial fields augment the information existing in the environment to improve cooperation between agents. Figure 10 illustrates that an agent (A) perceives an attractive signal (from agent B) and a repulsive one (from agent C).

The model is based on two modules dedicated to individual and cooperative behaviours. The first one measures the agent satisfaction, i.e. gives in real time an evaluation of the agent task progress. This *individual satisfaction* is defined by a value $P(t) \in [-P_{max}, P_{max}]$ (represented by the actions-perception loop in fig. 11). The second module evaluates a complementary satisfaction which concerns the interactions of the agent with its neighbours (noted perception of neighbours in fig. 11). These satisfactions are used in the action-selection module to decide if the current behavior must be continued or changed (cf. (Simonin and Ferber, 2000)).

These satisfaction measures are also used to enable cooperative interactions. Agents can communicate their satisfactions in order to influence their neighbours. They can locally broadcast attractive or repulsive signals, defined as numerical values: $I(t) \in [-P_{max}, P_{max}]$ (noted signals I in fig. 11). The semantic is the following: positive values for attractions and negative ones for repulsions. A *cooperative behavior* consists in moving in a direction which depends on the sign of the perceived signals: go towards the source if attraction or go away if repulsion. This reaction, named the altruistic behavior, is performed only when

$$|I_{ext}(t)| \geq P(t) \;\wedge\; |I_{ext}(t)| \geq I(t)$$

where I_{ext} is the strongest perceived signal. This condition expresses that an agent reacts to a signal if the signal intensity is greater than the intensity of its own satisfaction. The altruistic reaction is defined as a repulsive or attractive vector (it depends directly of the sign and intensity of the value received). Note this vector can be combined to environment constraints, as obstacles, to improve agents navigation.

Eventually, a process of signal-passing ensures the propagation of the signals. It is useful to perform recruitment tasks or to propagate repulsive forces between blocked agents (see details in (Simonin and Ferber, 2000)).

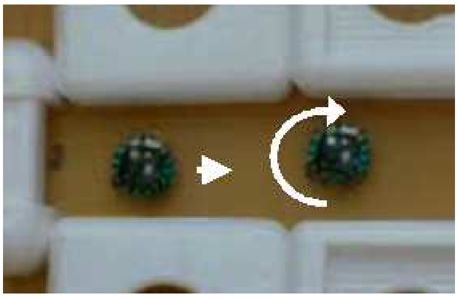

Fig. 12. Two mobile robots in a corridor

This model has been validated on different simulated problems such as foraging, navigation in constrained environments, box-pushing (Chapelle *et al.*, 2002) and with real robots as illustrated in fig. 12 (see details in (Lucidarme *et al.*, 2002)).

4.2 Formal specification of Agent architectures

Reactive-based architectures, such as the satisfaction altruism one, are generally not defined following a specific formalism (it does not exists a norm to express such architectures). Moreover, they are not necessary well specified in order to ensure their reuse. More generally, reactive multi-agent models suffer of a lack of formalism and specification.

(Hilaire *et al.*, 2001) have proposed a formal framework to specify multi-agent systems. It relies on the concepts of role, interaction and organisation. This formal notation composes Object-Z and Statecharts (Harel, 1987), called OZS formalism. This approach allows to represent multi-agent dynamic aspects, tools for specification analysis and mechanisms allowing the refinement of a high level specification into a low level specification which can be easily implemented.

The satisfaction-altruism model has been specified with this formal framework, cf. (Hilaire *et al.*, 2005). This work was useful to express the kernel of the model on one side and the applicative components on the other side. Then, it allowed to animate the model to test the specification on different scenarios.

In particular we have tested that simulated autonomous robots can avoid to be locked in a narrow corridor by applying the satisfaction-altruism model (they can cooperate to escape the corridor). This task was first tested with two robots as shown in fig. 12.

Figure 13 shows an example of such a test. The x axes represents time-points and the y axes represents discretized positions in the corridor for the upper figure and level of satisfaction for each robot for the other figure. The individual behavior consists in exploring the corridor. One can see that levels of satisfaction and trajectories are correlated. Indeed, each time the two robots are locked the satisfaction levels decrease. Here the satisfaction variation is depending on environmental constraints: the more a robot is surrounded by walls the faster its satisfaction falls. Thus the robot locked against a wall is more quickly dissatisfied than the other. Agents emit their dissatisfactions as local signals, then the altruism test becomes true for the less constrained robot. This robot plays the altruist role and changes its direction (it is the case around times 109, 155 and 235 on the figure 13). If a robot is not locked and can explore the corridor following its initial direction its satisfaction level increases.

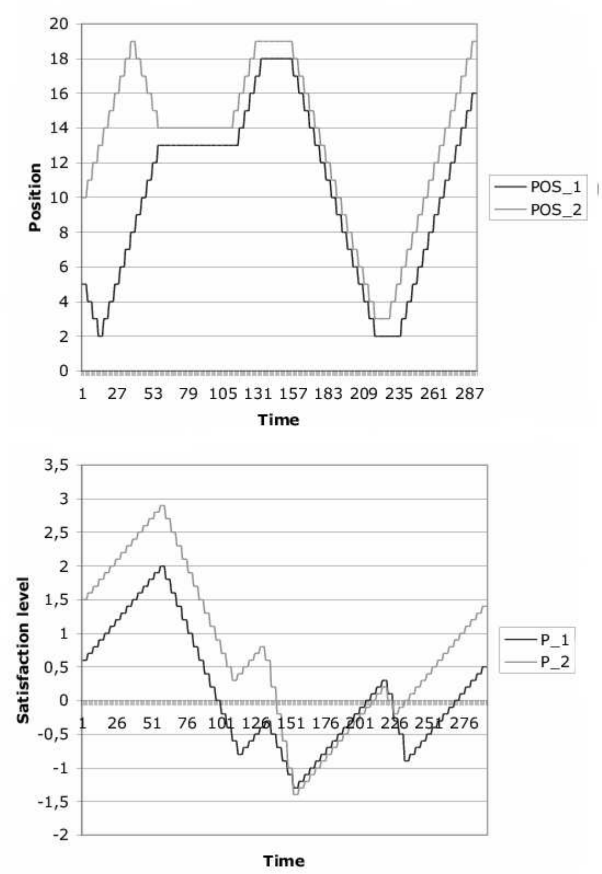

Fig. 13. Robots trajectories and satisfactions

The OZS formalism allows to study a MAS through animation techniques, but it is also useful to verify specification properties. Such a purpose can be done by using software like SAL (de Moura *et al.*, 2004). The SAL model checker has been chosen to enable the verification of the satisfiability of a property. For a instance, a property we have proven using the model checker is specified as follows:

$(\text{left}(r1)=\text{wall} \wedge \text{right}(r1)=\text{robot}) \Rightarrow \quad \diamond\, \text{right}(r1) = \text{empty}$

The formula states that if a robot r1 perceives a wall on its left side and a robot on its right side then eventually the robot at its right will move. The model checker with this property and the transition system produced generate no counter-example. It means that whenever a robot is locked between a wall and another robot it will be freed (see details in (Hilaire *et al.*, 2005)).

To finish with OZS formalism, note that specified systems can be simulated via a programming language, providing breakpoint construction. Breakpoint stops the specification execution when a condition is verified. Possible uses of breakpoints are, for example, configuration tests with predefined interaction scenarios and output of statistics.

5. CONCLUSION

This paper presented reactive architectures and some swarm principles to deal with autonomous multi-agent cooperation. Interests and drawbacks of such an approach for mobile robotics have been discussed. It has been shown that the application of such an approach to robotics needs to extend it in order to improve the system abilities, its control and the tools for their development. First, it has been shown that combining several reactive robots could provide robust and intelligent autonomous systems. Second, introducing direct communication between reactive agents allows to deal with complex cooperative tasks and conflict solving. The formal specification of the satisfaction-altruism model with a tool such as the OZS formalism showed that (reactive) architectures can be formally studied, tested and implemented.

To conclude, we can draw the main challenges that must be tackled to improve the application of reactive-based architecture in robotics:

- to introduce deliberative elements in reactive architectures in order to not limit the agent behavior to perception/reaction mechanisms, but without losing the collective properties
- to specify architectures and algorithms with formal tools in order to prove some individual and collective properties, as emphasized in (Nana, 2005)
- to define some norms and generic libraries for the implementation of reactive architectures
- to study emergent processes in order to understand and to reuse them
- to develop methologies to represent and to design reactive-based systems

REFERENCES

Alami, R. (2005). Multi-robot cooperation : Architectures and paradigms. In: *5e Journees Nationales de la Recherche en Robotique, Guidel, France.* http:// jnrr05.irisa.fr/ document/ jnrr-002-03-alami.pdf. pp. 41–49.

Arkin, R.C. (1987). Motor schema based navigation for a mobile robot: An approach to programming by behavior. *Proceedings of the IEEE Conference on Robotics and Automation* pp. 264–71.

Arkin, R.C. (1992). Cooperation without communication: Multiagent schema-based robot navigation. *International Journal of Robotic Systems* **9(3)**, 351–364.

Arkin, R.C. (1998). *Behavior Based Robotics.* The MIT Press.

Arkin, R.C. and T. Balch (1997). Aura: Principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence* **9**, 175–189.

Balch, T. and R.C. Arkin (1995). Motor schema-based formation control for multiagent robot teams. In: *Proceedings 1995 International Conference on Multiagent Systems.* San Fransisco, CA. pp. 10–16.

Barraquand, J., B. Langlois and J.C. Latombe (1992). Numerical potential field techniques for robot path planning. *IEEE Trans. on Systems, Man and Cybernetics* **22**(2), 224–241.

Beckers, R., O.E. Holland and J.-L. Deneubourg (1994). From local actions to global tasks: stigmergy and collective robotics. In: *Artificial Life IV: Proc. of the 4th Int. Workshop on the synthesis and the simulation of living systems, third edition, MIT Press.*

Beni, G. and J. Wang (1989). Swarm intelligence in cellular robotic systems. In: *Proceedings of the NATO Advanced Workshop on Robots and Biological Systems.* Il Ciocco, Tuscany, Italy.

Bonabeau, E. and G. Theraulaz (1994). *Intelligence Collective.* Editions Hermes.

Brooks, R.A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* **RA-2(1)**, 16–23.

Chapelle, J., O. Simonin and J. Ferber (2002). How situated agents can learn to cooperate by monitoring their neighbors' satisfaction. In: *15th European Conference on Artificial Intelligence.* Lyon, France. pp. 68–72.

Colorni, A., M. Dorigo and V. Maniezzo (1991). Distributed optimization by ant colonies. In: *in proceedings of ECAL91, European conference on artificial life, Paris, Elsevier.* pp. 134–142.

Connell, J.H. (1992). Sss: A hybrid architecture applied to robot navigation. In: *Proc. ICRA.* pp. 2719–2724.

de Moura, L., S. Owre, H. RueB, J. Rushby, N. Shankar, M. Sorea and A. Tiwari (2004). Sal 2. *In Computer-Aided Verification, CAV*

2004, Lecture Notes in Computer Sciences 3114 pp. 496–500.

Deneubourg, J.L., S. Goss, R. Beckers and G. Sandini (1991). Collectively self-solving problems. *Self organization, emergent properties and learning*.

Drogoul, A. and J. Ferber (1992). Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In: *in Proceedings of MAAMAW'92, Viterbo, 1992. (published as Decentralized A.I. 4, Elsevier North-Holland)*.

Drogoul, A., M. Tambe and T. Fukuda (1998). Proceedings of the 1st international workshop on collective robotics (cwr-98). In: *LNAI 1456*. Springer.

Ferber, J. (1999). *Multi-Agent Systems, an introduction to Distributed Artificial Intelligence*. Addison-Wesley.

Gat, E. (1998). On three-layer architectures. In: *Artificial Intelligence and Mobile Robots*. AAAI Press.

Harel, D. (1987). Statecharts: A visual approach to complex systems. *Science of Computer Programming* **8(3)**, 231–274.

Hilaire, V., A. Koukam, P. Gruer and J.P. Muller (2001). Formal specification and prototyping of multi-agent systems. *in Engineering Societies in the Agent's World, LNCS 1972*.

Hilaire, V., O. Simonin, A. Koukam and J. Ferber (2005). A formal framework to design and reuse agent and multiagent models. *Lecture Notes in Computer Sciences 3382* pp. 142–157.

Kube, C. R. and H. Zhang (1992). Collective robotic intelligence. In: *2d International Conference On the Simulation Of Adaptive Behavior (SAB 92)*. MIT Press, Cambridge, MA. Honolulu, HI. pp. 460–68.

Kube, C.R., C.A. Parker, T. Wang and H. Zhang (2005). Biologically inspired collective robotics. *Chapter 15 in Recent Developments in Biologically Inspired Computing*.

Lucidarme, P., O. Simonin and A. Ligeois (2002). Implementation and evaluation of a satisfaction/altruism based architecture for multi-robot systems. In: *International Conference of Robotics and Automation*. Washington. pp. 1007–1012.

Maes, P. (1991). A bottom-up mechanism for behavior selection in an artificial creature. In: *First International Conference On the Simulation Of Adaptive Behavior (SAB 91)*. Paris. pp. 238–246.

Mataric, M. J. (1995). Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems* **16 n 2-4**, 321–331.

Mataric, M.J. (1997). Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence* **9, 2-3**, 323–336.

Mirosot (2006). Mirosot robot. In: *http://www.fira.net/soccer/mirosot/overview.html*.

Nana, L. (2005). Architectures logicielles pour la robotique. In: *in Proceedings of 5e Journees Nationales de la Recherche en Robotique*. pp. 239–248.

Parker, L. (2000). Current state of the art in distributed autonomous mobile robotics. In: *Distributed Autonomous Robotic Systems 4*. L. E. Parker, G. Bekey, and J. Barhen eds.. pp. 3–12.

Parunak, H.V.D. (1997). Go to the ant: Engineering principles from natural agent systems. *Annals of Operations Research* **75**, 69–101.

Parunak, H.V.D. and S. Brueckner (2001). Entropy and self-organization in multi-agent systems. In: *Fifth International Conference on Autonomous Agents*. pp. 124–130.

Simonin, O. (2001). *Le modle satisfaction-altruisme: cooperation et resolution de conflits entre agents situs ractifs: application la robotique*. PhD thesis, Universit Montpellier II.

Simonin, O. (2005). Construction of numerical potential fields with reactive agents. In: *The Fourth International Joint Conference on Autonomous Agents and Multi Agent System*. pp. 1351–1352.

Simonin, O. and F. Gechter (2006). An environment-based principle to design reactive multi-agent systems for problem solving. *LNAI 3830 Lecture Notes in Artificial Intelligence* pp. 32–49.

Simonin, O. and J. Ferber (2000). Modeling self satisfaction and altruism to handle action selection and reactive cooperation. *6th International Conference On the Simulation Of Adaptive Behavior (SAB 2000 volume 2)* pp. 314–323.

Simonin, O. and O. Grunder (2006). A cooperative mobile robots architecture enabling paralyzed robots to move. In: *in Proceedings INCOM'06*.