Agent Oriented Design of a Soccer Robot Team

Anne COLLINOT*, Alexis DROGOUL* and Philippe BENHAMOU**

* LAFORIA - Université Paris VI / CNRS
4 place Jussieu - case 169
F-75252 Paris cedex 05 - FRANCE
{collinot, drogoul}@laforia.ibp.fr

** ONERA - National Institute for Aerospace Research and Studies
29 avenue de la Division Leclerc
F-92322 Chatillon - FRANCE
benhamou@onera.fr

Abstract

The multi-agent paradigm is widely used to provide solutions to a variety of organizational problems related to the collective achievement of one or more tasks. All these problems share a common difficulty of design: how to proceed from a global specification of the collective task to the specification of the local behaviors, which are to be provided to the agents? We have defined the *Cassiopeia* method whose specificity is to articulate the design of a multi-agent system around the notion of organization. This paper reports the use of this method for designing and implementing the organization of a soccer robot team. We show why we chose this application and how we designed it, we discuss its interest and inherent difficulties, in order to clearly express the needs for a design methodology dedicated to DAI.

Introduction

The multi-agent paradigm is widely used to provide solutions to a variety of organizational problems related to the collective achievement of one or more tasks: computer supported cooperative work, flexible workshop or network management, distributed process control, or coordination of patrols of drones (Avouris & Gasser 1992) (Werner & Demazeau 1992) (Demazeau & Muller 1991). All these problems share a common difficulty of design: how to proceed from a global specification of the collective task to the specification of the individual behaviors, which are to be provided to the agents that achieve the task. A problem of organization has to be solved, most of the time in a dynamic fashion, so as to obtain the collective achievement of the considered task. In previous works, we have defined the *Cassiopeia* method (Collinot, Carle & Zeghal 1995) which purpose is to provide a methodological framework to design multi-agent systems (MAS). The underlying principle of *Cassiopeia* is to articulate the design activity around the notion of organization. In this paper, we report the use of the *Cassiopeia* method in the context of a research project (MICROB) on collective phenomena of organization in robot societies.

The paper is organized as follows: in section 2, we present the MICROB project and the application we are concerned with, that is the design of a soccer robot team. We discuss the interests and difficulties that characterize this application, which directly sets the requirements for a methodology dedicated to DAI. In section 3, we give an overwiew of the methodological approach of *Cassiopeia*. In section 4, we report the use of the *Cassiopeia* method for the design and implementation of the soccer robot team. Finally, in section 5, we discuss our current results, based on a simulation of the soccer robot team.

The MICROB Project and the Soccer Robots

The aim of the MICROB project (Making Intelligent Collective Robotics) is to investigate collective phenomena of organization in societies of robots. The project relies on the use of a simple testbed, which enables to conduct experiments that focus on the organizational features of agents.

The MICROB Experimental Testbed

The MICROB mobile robots are made of remote-controlled cars, which thus have no sensor and no on-board decision module (Fig. 1). Each car receives its commands through a remote-control pilot, which transmits the instructions that have been worked out by either a software agent or a human agent. A camera is filming the scene while an image processing system is analyzing the position and the speed of all the robots and objects that are on the field. These data are then placed into a simulated environment within which the software agents are operating to make appropriate decisions that will be sent to the remote-control pilots. Given this testbed, the issue is how to design the software agents, so that the robots they command can exhibit

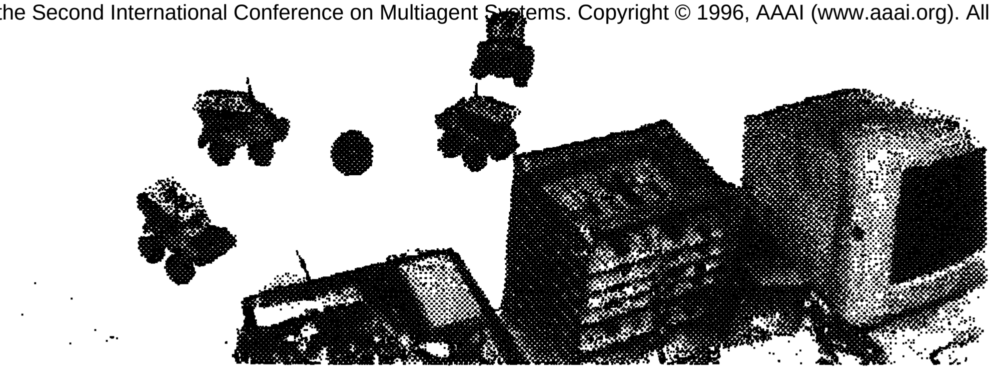

Figure 1. The MICROB Project: the Cars and Computers.

capabilities to organize themselves to collectively achieve a task.

The Application to a Soccer Robot Team

The application to a soccer robot team has been chosen to investigate the design of robots with capabilities to organize themselves to achieve a collective task defined from a common goal. The playground in which the robots are moving is rectangular and comprises a goal at each end. There are eigth cars on the field, which are distributed into two teams. The cars of one team are remote-controlled by software agents while the cars of the opposing team are remote-controlled by human agents. The aim of the designer is that the team of software agents scores goals while defending its side.

Designing a Soccer Robot Team : Interests and Difficulties

The problem is thus to design the sofware agents so that they can play with team spirit. Indeed, they should have capabilities to organize themselves in order to win. The main difficulty is to express locally at the agent level the behaviors allowing to obtain the achievement of the collective task by the team (the MAS). There are two other difficulties: one is related to the dynamics of the game that makes it impossible neither to define in advance the organization of the robots, nor to control the game in a centralized way; the other difficulty derives from the fact that two organizations are facing each other, with one of them unpredictable (the robots team controlled by human agents).

This testbed is also interesting since it covers most of the issues of artificial organizations with respect to analysis, design, implementation, experimentation and validation (Kitano et al. 1995). The soccer robot application thus appeared quite appealing to us in the context of developing the *Cassiopeia* framework, since it enables us to structure the design of a large project in a comprehensive manner, and to evaluate the contribution of *Cassiopeia* more seriously than in the case of a toy application.

Requirements for a Methodology Dedicated to DAI

In most cases, a large project, especially a pluri-disciplinary project, requires the use of a methodology. The main role of such a methodology is to identify the necessary steps that permit to proceed from the project requirements to their fulfillments (i.e. the project life cycle). This approach relies on: (a) the use of an homogeneous terminology, which has a meaning at each step of the cycle and which supports correspondance between steps; (b) the use of *operational* conceptual abstractions: (c) the possibility to backtrack from any step to previous ones without losing what has been done.

In the case of DAI, the project requirements consist in having agents achieving a collective task. To fulfill these requirements, one must design the agents with a specific attention to their capabilities for organizing themselves. The existing methodologies, especially the object oriented methodologies (Rumbaugh et al. 1991) that can be considered because of some similarities such as distribution or locality, provide an interesting basis of analysis: indeed, they enable the distribution of the requirements along both structural and behavioral axes. However, they offer no methodological framework to take the organization issue into account, since it is not considered as an object of analysis in itself. On the other hand, the few DAI works (e.g. (Moulin & Cloutier 1994)) that deal with methodological aspects are not very satisficing, since they indirectly address the organization issue through the use of various negotiation or coordination techniques, which in fact are only particular implementation methods.

The features of the MICROB project stress the need for a methodological framework to integrate the descriptive and operational aspects of the organization as early as the analysis step, both for implementation and documentation reasons. We thus have decided to use the *Cassiopeia* method, which is presented in the next section.

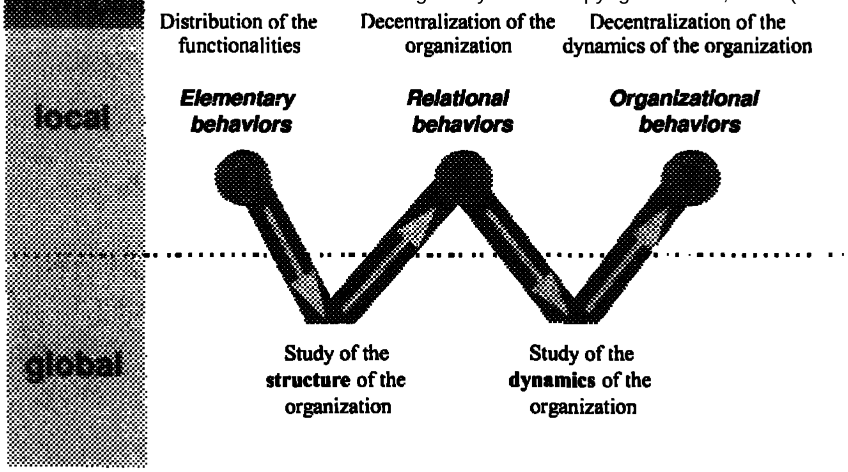

Figure 2. The *Cassiopeia* Method.

The Methodological Approach of *Cassiopeia*

The *Cassiopeia* method is primarily a way to address a type of problem-solving where collective behaviors are put into operation through a set of agents. Although *Cassiopeia* does not offer yet all the components that one could expect to find in a complete method, it provides a methodological framework to better understand and plan the design of a computational organization. According to *Cassiopeia*, a MAS should be designed in terms of agents provided with three levels of behavior: (1) elementary, (2) relational, (3) organizational. This permits the identification of various types of choice made at precise points in the design process. The modification of a MAS is then systematically related to the revision of these choices.

The underlying principle of *Cassiopeia* is that the problem of organizing individuals to achieve collective problem-solving must be addressed as such by the designer and/or the agents. Such an organizational issue arises since *functional dependencies* are inherent to the collective achievement of the considered task: the activation of an individual problem-solving behavior affects and is affected by problem-solving behaviors activated by other agents. The whole set of these dependencies determines the *coupling* of the organizational problem underlying the task achievement. We distinguish two types of coupling: (1) When the individual behaviors are not competing, the coupling is static, that is the organization of the agents remains unchanged and the designer can define it in advance (e.g., the multi-expert systems in which there is exactly one expert per domain of expertise); (2) When some sort of competition is to be managed (individual behaviors exist that are equivalent for a given situation, or a same behavior can be operated by different agents), the coupling is dynamic, that is the organization cannot be defined in advance since it depends on the context. Therefore, the designer can only consider the organizational structures between agents, which will be instantiated within the problem-solving context. This is the case in our application, where the soccer robots should organize themselves dynamically.

The *Cassiopeia* method proceeds from the collective task to the design of the MAS, along three steps that reconcile both local and global views (Fig. 2): (1) definition of the required elementary behaviors in order to define the types of agent; (2) structural description of the organization of the different types of agent; (3) description of the organization dynamics.

Step 1 - Elementary Behaviors

The first step of *Cassiopeia* consists in listing the elementary behaviors that are required for the achievement of the considered collective task. The identification of these behaviors does not come under *Cassiopeia*: it can result from a functional (e.g. (Yourdon 1989)) or object-oriented (e.g. (Rumbaugh et al. 1991)) analysis step. The designer defines different types of agents based on the elementary behaviors.

Step 2 - Relational Behaviors

The second step consists in analyzing the organization structure based on the dependencies between the elementary behaviors. These dependencies define the *coupling graph* underlying the considered collective task. The dependencies between the agents are elaborated based on this coupling graph as follows: first, the coupling graph is projected onto the different types of agents; then the inconsistent dependencies are removed and if necessary, some dependencies are ignored according to the available heuristics of the application domain. This refined projected graph contains the sole dependencies that are supposed relevant to the achievement of the task. Such dependencies between the different types of agents are called *influences*.[1] The refined projected graph is called the *influence graph*. The paths and the elementary circuits of this influence graph define the potentialities of grouping the different types of agents together and therefore, provide a global representation of the organization structure.

Based on this analysis, the designer specifies the *relational behaviors* that enable the agents to identify and handle the influences. In this purpose, he determines the *influence signs*, which are produced by each type of *influencing agents*, and specifies how an *influenced agent* takes into account an influence sign, that is which elementary behaviors the agent activates and in what fashion.

Step 3 - Organizational Behaviors

The third step addresses the dynamics of the organization. It consists in specifying the behaviors that will enable the agents to manage the formation, durability and dissolution of groups.

The relational behaviors may allow the formation of several groups that are redundant with respect to their end. When such a redundancy is useless or costly, it is necessary to determine the criteria that affect the choices to form a group rather than another one. As described in the previous step, when an agent needs to form a group, it produces the relevant influence signs.The occurrence of redundant groups thus indicates a redundancy of means to meet the need of this particular agent, called the *trigger* agent. Such a trigger agent belongs to all potential groups meeting its need and should evaluate them to decide which one is the most appropriate in the current context. The designer should thus (1) identify the trigger agents (according to the grouping potentialities that have been identified in the influence graph) and (2) determine for each of them, the selection methods allowing to control the formation of groups. There exists a variety of techniques for making such choices (Decker 1987), based on: task announcement such as all the techniques deriving from the Contract Net (Smith & Davis 1980); the notion of consensus and negotiation among agents belonging to concurrent groups (Sycara 1989) (Rosenschein & Genesereth 1985); priorities allowing to order the potential groups (Erdmann & Lozano-Pérez 1986); or the use of a supervisor or a hierarchy (Le Pape 1990). These techniques define a first type of organizational behavior: the *group formation behaviors*.

Then, the designer specifies the *commitment signs*, which are produced by the trigger agents to indicate to other agents that a group is formed to meet their need. These signs allow the agents that are not members of the formed groups to control (e.g., to inhibit) their relational behaviors. The designer thus defines for each type of agent a second type of organizational behavior, which take the commitment signs into account to organize the relational behaviors: the *joining behaviors*.

Finally, the choices resulting from the group formation behaviors may need to be revised. A group ceases to exist when it has carried out its commitments; or a group can be replaced by a more efficient one. The designer thus defines for each type of agent a third type of organizational behavior for dissolving a group: the *group dissolution behaviors*.

More details on the *Cassiopeia* method can be found in (Collinot, Carle & Zeghal. 1995).

Using *Cassiopeia* to Design the Soccer Robot Team

In this section, we report how the three steps of the *Cassiopeia* method have been followed to design the agents that command the soccer robot team.

Step 1 - The Elementary Behaviors of the Soccer Robots

Four elementary behaviors are required to play soccer: (1) *shoot* the ball in a given direction (usually that of the opponents' goal); (2) *place* oneself on the playground, waiting for a pass; (3) *block* an opponent's way; (4) *defend* one's side goal against the opponents' attacks. The details of the algorithms that implement these behaviors need not be given in this paper. However, note that the behaviors are *elementary* with respect to the SMA design level, but not with respect to robotics concerns. Indeed, they are at a more abstract level than the standard behaviors of a robot (turn left, accelerate, etc.) and represent a combination of them (e.g. the shooting behavior).

In the current state, the MICROB sofware agents are all provided with the four behaviors. However, they activate only one of them at a given time, which is referred to as their *active elementary behavior*. The agents that shoot, block, defend or place themselves are then respectively called a shooter, blocker, defender or placer. This led us to design four different types of agent. Thus, at a given time, an agent has a particular *role* in its team, which is dynamically determined by the role of the others. This kind of collective control will be held by the relational and organizational behaviors.

Step 2 - The Relational Behaviors of the Soccer Robots

The four soccer behaviors are obviously depending one on another. For instance, the shoot of a robot depends on how the other robots are placed in the playground: one should not make a pass to a misplaced robot. The very structure of the play relies on such dependencies and following the *Cassiopeia* method, we have defined the corresponding coupling graph (Fig. 3). The following dependencies have been identified (an arrow from A to B means that B potentially depends on A):

1. This notion of influence is close to the notion of social dependencies introduced in (Castelfranchi, Miceli & Cesta 1992) and further developed in (Sichman & Demazeau 1995).

d1. Blocking an opponent can help another robot to better place itself.

d2. Defending can help oneself or another robot to better place itself.

d3. It may be necessary to place oneself if another robot wants to shoot.

d4. Defending may allow to catch the ball of the opponent.

d5. Blocking can help oneself or another robot to shoot the ball.

d6. Shooting can help oneself or another robot to shoot (this is the pass).

d7. Defending depends on the other robots' defence strategy.

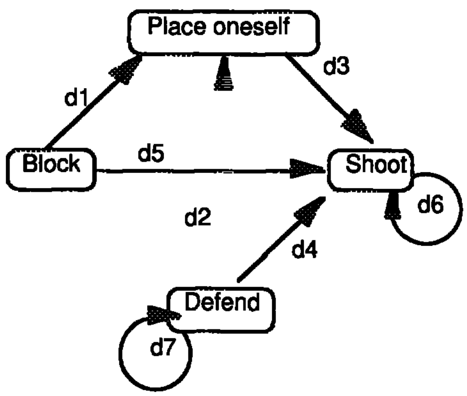

Figure 3. The Coupling Graph of the Soccer Robot Game

In order to define the influences between the soccer behaviors, the coupling graph of Fig. 3 is projected on the four types of soccer agents. Although we considered only the dependencies whose target is an active behavior, the resulting graph is intricate and, following *Cassiopeia*, we made assumptions about the relevance of the dependencies. For example, we decided to ignore the dependencies whose origin is a passive blocking, shoot, or defence (except when their target is an active defence behavior). This choice is clearly discretionary, but it is explicit and we know where it has been done in the design cycle. The resulting influence graph (Fig. 4) is used to determine what influence signs should be produced. Once again, some choices are to be made, in particular regarding the agent communication protocol, that is how the influence signs are communicated. In the current implementation, we chose a simple object-oriented synchronous message passing: an influence sign is a message sent to an agent, its interpretation is the execution of the corresponding method, which returns a value to the sender. In other applications, it could be an asynchronous communication, the diffusion of information in the environment, writing on a blackboard, etc. *Cassiopeia* does not detail this aspect, which is left to the designer.

We now describe the considered influences, along with the associated influence signs:

- A blocker can provide help to a shooter. A shooter is able to send an *help* message, which can be interpreted by any type of agent to return a value representing its capacity to block, based on its perception of the game.
- A shooter may make a pass to itself or another shooter. The shooter sends a *shoot* sign and the receiver of the message returns its capacity to shoot. The shooter then adjusts the direction of the shoot.
- A shooter must take the other robots' positions into account before shooting. The *place* sign is sent by the shooter, and the receiver of the message returns its capacity to shoot and place itself.
- Any agent can influence a defender because of its position on the playground. The *position* sign is produced by the defender, and the receiver of the message returns its position, which is then used by the defender to adjust the direction of its move.

In this current implementation, the relational behaviors are simply handling the messages sending and the associated methods.

Step 3 - The Organizational Behaviors of the Soccer Robots

The behaviors allowing the agents to dynamically organize themselves according to their mutual influences, are based on the contract net mechanism. In this implementation, the *trigger agent* obviously is the manager that proposes the contracts. Based on an analysis of the influence graph (Fig. 4), we decided that the most appropriate type of agent for being the manager is the shooter.

- The *group formation behaviors* are thus exclusively defined for the shooter agents: when an agent becomes a shooter, it first evaluates the capabilites of its team mates to place themselves and block opponents, and then establishes a placing contract and a shooting contract with those which return the best values. Also, a shooter always evaluates the capabilities to shoot of the other robots, and when one of them is better placed than itself, the shooter establishes a *shooting* contract with it.
- The associated *commitment signs* simply are the roles of the robots, which are determined by the contracts they are engaged in. These signs are messages that are sent by the current shooter to the other robots (including itself). By default, any non shooter robot with no contract is a defender.
- The *joining behaviors* of a robot simply consist in behaving according to the role associated with the commitment sign it has received.
- The *dissolution behaviors* are exclusively defined for the shooter robots. They cover two distinct situations. A shooter can cancel any previous contract it has passed

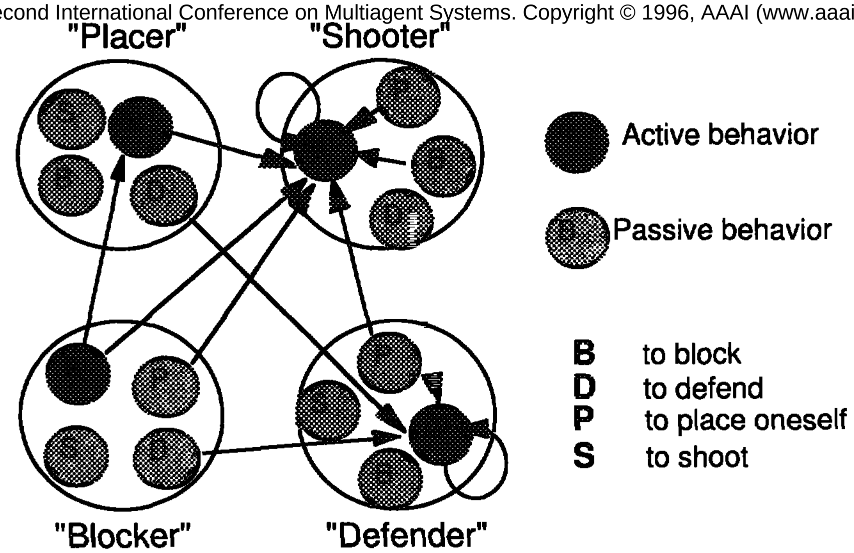

Figure 4. The influence graph for the soccer robot team

with a blocker or a placer if it happens that other robots have a better capability to place themselves or block an opponent. The member of the group that has been rejected automatically becomes a defender. When a shooter passes a shooting contract with another robot, it cancels all the contracts it has passed previously and becomes a defender.

Figure 5 shows an example of the dynamics of group formation. The robot on the upper side is the current shooter. He proposes a placing contract to the other robot. After the first shoot, the capability to shoot of the previous placer happens to be higher than that of the current shooter. The placer then becomes a shooter, and the shooter becomes a defender. The new shooter immediatly proposes a placing contract to the defender, and so on.

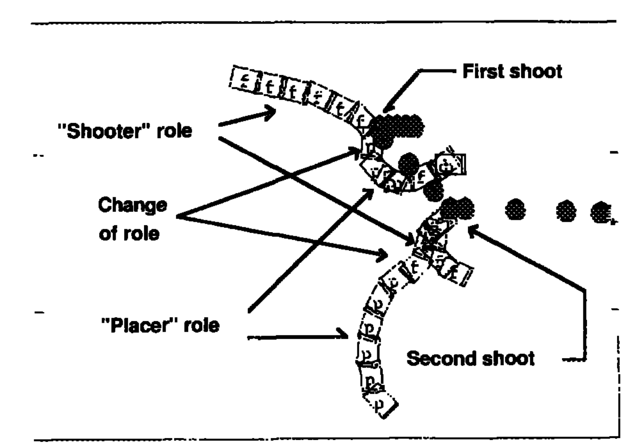

Figure 5. Example of the Dynamics of Group Formation

Preliminary Results and Work in Progress

Although the experimental results need to be proved by physical experiments, we used a simulation of the real environment for a preliminar evaluation of the agents behaviors. The results are very satisficing in terms of both the efficiency of the agents team (faced to a human team playing with the keyboard) and the quality of their play. *Cassiopeia* as a method for designing a MAS (that is before the implementation step) has proven very useful since it has allowed to evaluate various types of agent architecture, without having to revise the analysis choices.

We are currently using *Cassiopeia* to design two other types of soccer robot teams: a reactive robot team and a learning robot team. When designing a reactive robot team, it is obviously not relevant to consider the third step of our method. Indeed, in reactive MAS, organizational behaviors are expected to implicitly emerge from the interactions between agents. The use of *Cassiopeia* is an attempt to clarify this kind of hypothesis. Regarding the design of the learning robot team, we want the agents to learn to play with team spirit, given their elementary behaviors. The relational and organizational behaviors are replaced by two classifier systems (Holland & Reitmann 1978). Clearly, the three levels of behavior of *Cassiopeia* help in determining what should be given and what should be learned.

Conclusion

The major contributions of *Cassiopeia* are the possibilities to: (1) manipulate conceptual abstractions (e.g. influence, group), in an homogeneous way from the analysis step to the implementation step; (2) interface the languages of the robotician and the computer scientist by providing a conceptual level that is neither too sketchy nor too technical: this is certainly due to the use of an operational multi-agent oriented terminology; (3) to clearly distinguish, at design time, between domain-dependent and organization based behaviors.

However, the use of the method has also shown the limits of a simple methodological framework. We need to refine the current terminology, especially when the terms are used by different DAI researchers with different meanings. It is also necessary to integrate *Cassiopeia* as a design method, to existing analysis methods (in particular object-oriented methods), which allow to make the necessary choices related to a domain-dependant structural and behavioral distribution. Finally, it is necessary to determine more precisely how to integrate the techniques developed by the DAI researchers with regard to communication protocols, negotiation or coordination mechanisms, agent architectural components, etc. All these improvements should enable *Cassiopeia* to evolve from a methodological framework to a complete *Agent Oriented Design* method, qualified to provide the DAI community with methodological tools allowing: (1) the development of ambitious multi-agent projects, (2) more fruitful interactions between research, engineering and industry and (3) a more efficient re-use of their works.

Acknowledgements

The MICROB project is conducted by D. Duhaut (Robotics Laboratory of Paris) and A. Drogoul. We thank L. Magnin for designing the simulator and L. Ploix for implementing the agents' behaviors.

References

Avouris, N., and Gasser, L. eds. 1992. *Distributed AI: Theory and Praxis*. Kluwer Academic Publishers.

Castelfranchi, C., Miceli, M., and Cesta, A. 1992. Dependence Relations among Autonomous Agents. In Werner, E., and Demazeau, Y., eds., *Decentralized A.I. 3*. Amsterdam: Elsevier Science Publishers B.V./ North Holland.North Holland. 215-227.

Collinot, A., Carle P., and Zeghal, K. 1995. Cassiopeia: a Method for Designing Computational Organizations. In Proceedings of the First International Workshop on Decentralized Intelligent Multi-Agent Systems, 124-131. Krakow, Poland.

Decker, K. S. 1987. Distributed Problem-Solving techniques: A Survey. *IEEE Transactions on Systems, Man and Cybernetics* 17(5): 730-740.

Demazeau, Y., and Muller, J.-P. eds. 1991. *Decentralized A.I. 2*. Amsterdam: Elsevier Science Publishers B.V./ North Holland.

Erdmann, M., and Lozano-Pérez, T. 1986. On Multiple Moving Objects. In Proceedings of the IEEE International Conference on Robotics and Automation, 1419-1424. San Francisco.

Holland, J.H., and Reitmann, J.S. 1978. Cognitive Systems based on Adaptive Algorithms. In Waterman, D. A., and Hayes-Roth, F., eds., *Pattern Directed Inference Systems*. New-York:Academic Press. 313-329.

Kitano, H., Minoru, A., Kuniyoshi, Y., Noda, I., and Osawa, E. 1995. RoboCup: The Robot World Cup Initiative. In Proceedings of the Workshop on Entertainment and AI/Alife. Montreal, Quebec: International Joint Conference on Artificial Intelligence.

Le Pape, C. 1990. A Combination of Centralized and Distributed Methods for Multi-Agent Planning and Scheduling. In Proceedings of the IEEE International Conference on Robotics and Automation, 488-493. Cincinnati.

Moulin, B., and Cloutier, L. 1994. Collaborative Work based on Multi-Agent Architectures: A Methodological Perspective. In Aminzadeh, F., and Jamshidi, M., eds., Soft Computing: Fuzzy Logic, Neural Networks and Distributed Artificial Intelligence. Prentice Hall. 261-296.

Rosenschein, J., and Genesereth, M. 1985. Deals Among Rational Agents. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 91-99.

Rumbaugh, J., Blaha, M., Eddy, F., Premerlani, W., and Lorensen, W. 1991. *Object Oriented Modeling and Design*. Prentice Hall..

Sichman, J. and Demazeau, Y. 1995. Exploiting Social Reasoning to Deal with Agency Level Inconsistency. In Proceedings of the First International Conference on Multi-Agent Systems, 352-359. San Francisco.

Smith, R. G.; and Davis R. 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem-Solver. *IEEE Transactions on Computers* C29(12): 1104-1113.

Sycara, K. 1989. Multiagent Compromise via Negotiation. In Gasser, L., and Hunhs, M. N., eds., *Distributed Artificial Intelligence II*. San Mateo, California: Morgan Kaufmann. chapter 6, 119-137.

Werner, E., and Demazeau, Y. eds. 1992. *Decentralized A.I. 3*. Amsterdam: Elsevier Science Publishers B.V./ North Holland.North Holland.

Yourdon, E., 1989. *Modern Structured Analysis*. Englewood Cliffs, New Jersey: Yourdon Press.