Autonomous Robots 15, 155–168, 2003

Distributed Coordination in Heterogeneous Multi-Robot Systems

LUCA IOCCHI AND DANIELE NARDI

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198, Roma, Italy

iocchi@dis.uniroma1.it

nardi@dis.uniroma1.it

MAURIZIO PIAGGIO AND ANTONIO SGORBISSA

Dept. of Communication, Computer and System Sciences, Università di Genova, Via Opera Pia 13, 16125 Genova, Italy

piaggio@dist.unige.it

sgorbissa@dist.unige.it

Abstract. Coordination in multi-robot systems is a very active research field in Artificial Intelligence and Robotics, since through coordination one can achieve a more effective execution of the robots' tasks. In this paper we present an approach to distributed coordination of a multi-robot system that is based on dynamic role assignment. The approach relies on the broadcast communication of *utility functions* that define the capability for every robot to perform a task and on the execution of a coordination protocol for dynamic role assignment. The presented method is robust to communication failures and suitable for application in dynamic environments. In addition to experimental results showing the effectiveness of our approach, the method has been successfully implemented within the team of heterogeneous robots Azzurra Robot Team in a very dynamic hostile environment provided by the RoboCup robotic soccer competitions.

Keywords: multi-robot systems, distributed coordination, dynamic task assignment, communication

1. Introduction

Coordination in Multi-Robot Systems (MRS) is nowadays one of the most interesting areas of research in Artificial Intelligence and Robotics (Dudek et al., 1996; Parker, 1998; Kitano et al., 1998). In fact, multi-robot systems are being developed from both a biological and an engineering perspective. In the first case the goal is to simulate the properties of biological systems, while in the second case the goal is to improve the effectiveness of a robotic system both from the viewpoint of the performance in accomplishing certain tasks (Dudek et al., 1996) and in the robustness and reliability of the system (Parker, 1998).

A significant boost to the work on MRS has recently been given also by the Robot competitions and RoboCup, in particular. In fact, the design of MRS is regarded as one of the major scientific challenges to be developed in the RoboCup environment (Kitano et al., 1998). In particular, according to the organization of RoboCup, the real robot leagues as well as the simulated league provide different settings, where the design of MRS is realized according to different hypotheses. The most distinguishing feature, as compared with previous work on MRS, is that the RoboCup environment is highly dynamic and hostile due to the presence of an opponent team.

In this paper we present a distributed approach to coordination in MRS that has been originally developed within the Azzurra Robot Team (ART) of robotic soccer players participating in the RoboCup competitions in the mid-size category (Castelpietra et al., 2000). Nonetheless, the methods and techniques presented here are rather general and suitable for application in other environments where similar underlying assumptions are satisfied.

ART (Nardi et al., 1999) is composed of different robotic players developed in various Italian universities. Each university in the team is responsible for the development of one or more robots which differ in the mechanics (robot base and kickers), in the hardware (processor type and speed) and in the control software (different techniques and approaches for perception, reasoning and motion control) from the robots of the other research groups. Because of this kind of organization, coordination among the ART robots requires not only a distributed coordination protocol, but also a very flexible one, that allows to accommodate the various configurations that can arise by forming teams with players equipped with different basic features. Due to the intrinsic heterogeneity of the team, the dynamic environment in which the robots act, and the complexity of the task (playing soccer), our main focus in the development of such a MRS has been the implementation of a coordination framework suitable for decomposing the complexity of the task to be carried out into different sub-tasks and for dynamically assigning these tasks to the robots in the MRS.

Given the very wide scope of the proposals on MRS and the relatively youth of the research area, a common framework for the work on MRS is difficult to identify. A MRS cannot be simply regarded as a generalization of the single robot case and the proposed approaches need to be more precisely characterized in terms of assumptions about the environment and in terms of the internal system organization. Classifications of the work on MRS has been presented by Dudek et al. (1996), Cao et al. (1997), Iocchi et al. (2001), and Balch et al. (2002). In Dudek et al. (1996) the classification of MRS is focused on the communication and computation aspects, while in Cao et al. (1997) several application domains for MRS are described and many systems are classified according to the task they are realized for. In Iocchi et al. (2001) MRS are analyzed by addressing their deliberation and reactivity capabilities. Other specific goals of the research in MRS include also the issue of explicit versus implicit communication (Balch and Arkin, 1995) or exploring specific coordination strategies for specific problems (Guibas et al., 1999).

In order to properly characterize the present proposal, it is worth highlighting the hypotheses underlying the MRS described by the present work that can be summarized as follows:

- *Communication-based coordination*: the usage of communication among the robots to improve team performances, allowing the robots to acquire more information and to self-organize in a more reliable way.
- *Autonomy in coordination*: the robots are capable to perform their task, possibly in a degraded way, even in case of partial or total lack of communication.
- *Distributed coordination*: the communication capabilities, combined with the autonomy requirement, require that each robot, while interacting with the others, must rely on local control (hence, the system is distributed).
- *Heterogeneity*: the robots are heterogeneous both from hardware and software viewpoints, they can usually perform the same tasks but with different performance.
- *Highly dynamic, hostile environment*: the robots must be able to perform the assigned task in the presence of external and dynamic changes in the environment.

The building blocks of our proposal are a communication layer and a coordination protocol. The former provides for suitable inter-robot information exchange, as well as, proper interoperation within each robot computational model. The latter has been designed to deal both with roles (defender, attacker, etc.) and with strategies (defensive, offensive). While the strategic level may be demanded to an external selection, roles are dynamically assigned (Veloso and Stone, 1998; Stone and Veloso, 1999) to the robots during the game.

In the paper we also address the problem of evaluating the performance of MRS (Balch, 1999, 2000), and are able to show through the analysis of the game logs that, in the operation environment, both the space coverage and the role exchange are effectively accomplished.

The paper is organized as follows. In the next section, we describe the architecture of the MRS, then the communication infrastructure and the coordination protocol. We then present the experimental results on coordination of the ART robotic players, address related work and possible application domains of the approach presented. Finally, we draw some conclusions, specifically discussing both the main features of the proposed approach and some issues that deserve further investigation.

2. System Architecture

Our approach to multi robot coordination is based on a coordination protocol run by all the robots of the team. This is obtained by implementing on every robot of the team a common subsystem that is responsible for communication on a (usually wireless) network and to execute the coordination protocol.

The main features of the proposed approach are that the team may easily include heterogeneous robots and that coordination is distributed, since each robot can take decisions on its own behavior based on its knowledge of the situation and no robot can take direct decisions about the behavior of other robots.

In Fig. 1 the system architecture for multi robot coordination is shown. The coordination module for every robot is organized into two layers: the communication layer that is responsible for exchanging data among the robots in the network, and the coordination protocol that performs a negotiation with the other robots and provides other modules of the robot with information about the actions that the robot must perform according to the team goals.

The robots may also be connected to a monitor that is typically used to visually inspect the behavior of the team and also to send general commands (like start or stop) to all the robots.

An important building block, within such an architecture, is ETHNOS (Expert Tribe in a Hybrid Network Operating System) (Piaggio et al., 2000b) a real-time software architecture and a programming environment dedicated to the development of distributed intelligent robotic applications. In ETHNOS, in order to allocate and guarantee system time for communication activity, the communication infrastructure is fully integrated in the real time software architecture for robot control.

In ETHNOS all the concurrent activities required for the control of a mobile robot are demanded to entities called experts. As an architecture for the development of real-time robotics applications, ETHNOS can schedule three different kind of experts:

1. periodic experts, i.e., real-time tasks which execute at a fixed rate
2. sporadic experts, i.e., real-time tasks which execute in subordination to specific conditions,
3. background experts, i.e., non real-time tasks which execute when the scheduler has no pending requests coming from periodic and sporadic expert to satisfy (for a more exhaustive description of ETHNOS scheduling policy (Piaggio et al., 2000a)).

In the development of the ART MRS ETHNOS has played a twofold role: on some of the robots is used as the underlying operating system, in other robots it simply implements the communication layer. In the former case ETHNOS has been used as a basis for the implementation of different cognitive models on individual robots. In the latter case, the ETHNOS communication facilities ported as a C-library on other control environments, provide a flexible and efficient communication layer suitable for effective implementation of a heterogeneous team of robots.

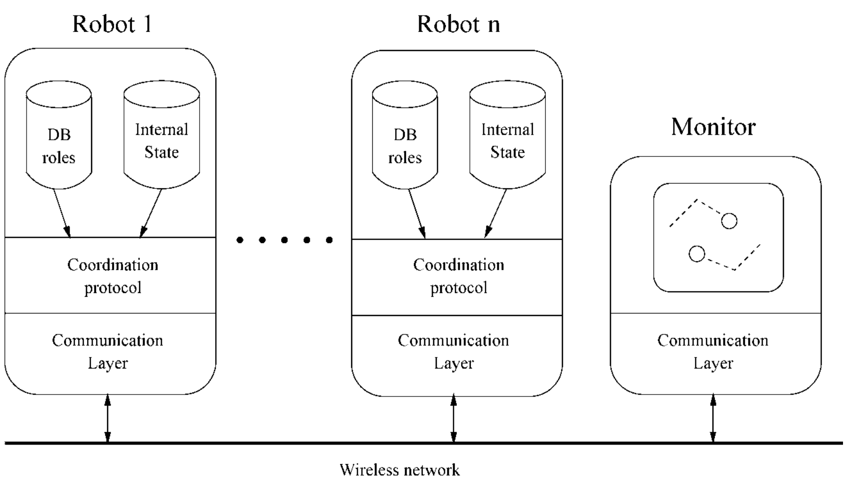

Figure 1. System architecture for robot coordination.

Summarizing, each of the communicating units in the architecture shown in Fig. 1, is based either on the ETHNOS system or on some other control systems enriched with the ETHNOS layer for communication. We have three other kinds of control environments, one of them being the general purpose system SAPHIRA (Konolige et al., 1997), and two specialized systems used in other robotic platforms.

While the features of ETHNOS are described in detail elsewhere (Piaggio et al., 2000b), in the next section we focus on the communication layer that provides the infra-structure for our MRS. The coordination method and the the coordination protocol are described in Section 4.

3. Communication Infra-Structure

As already mentioned, the communication infrastructure of our MRS has been developed by relying on ETHNOS, which implements a suitable inter-expert message-based communication protocol (the EIEP—Expert Information Exchange Protocol (Piaggio and Zaccaria, 1997)) fully integrated with the central system. The EIEP allows the system developer to decouple the different experts in execution, to reach, as close as possible, the limit situation in which the single expert is not aware of what and how many other experts are running. In this way an expert can be added, removed or modified at run-time without altering the other components of the system.

Expert decoupling is achieved by eliminating any static communication link. The EIEP is essentially an efficient implementation of a blackboard (Engelore, 1988) in a network distributed environment. In fact, the EIEP is based on an asynchronous publish-subscribe messaging paradigm. The single expert subscribes to the particular message types, known a priori. When another expert publishes any message of the subscribed types, the subscriber receives it transparently regardless of the particular machine on which they are produced or on which an expert subscribed. This allows EIEP to be used uniformly both to design the control method within the experts of a single robot and to support transparent inter-robot communication. In fact, in ETHNOS the different experts are also allowed to subscribe to communication clubs. It is the responsibility of the system to dynamically and transparently distribute the messages to the appropriate club members.

This general methodology has been used in different applications (i.e., service robotics (Piaggio et al., 2001) and RoboCup (Piaggio et al., 2000b)) and adapted in an application-dependent way by appropriately defining the number and type of clubs used. Specifically, in this RoboCup application, we are allowing the robots to communicate in a single club—the team club—(to which all of them have subscribed) and with an external supervisor (the coach) that monitors the activity of all the robots.

The EIEP also deals with low level network communication. In fact, since in RoboCup (and in general in mobile robotics) network communication is often wireless (i.e., radio link, Wavelan, etc.), due to noise or interference transmission packets are sometimes lost. In this context, both TCP-IP and UDP-IP based communication cannot be used: the former because it is intrinsically not efficient in a noisy environment; the latter because it does not guarantee the arrival of a message, nor any information on whether it has arrived or not. For this reason we have also designed a protocol for this type of applications, called EEUDP (Ethnos Extended UDP), which extends UDP with the required features.

The EEUDP allows for the transmission of messages with different priorities. The minimum priority corresponds to the basic UDP multicast (there is no guarantee on the message arrival) and it is used for data of little importance or data that is frequently updated (for example the robot position in the environment that is periodically published). The maximum priority is similar to TCP because the message is sent until its reception is acknowledged. However, it differs from TCP because it does not guarantee that the order of arrival of the different messages is identical to the order in which they have been sent (irrelevant in ETHNOS applications because every message is independent of the others), which is the cause of the major TCP overhead. Different in-between priorities allow the re-transmission of a message until its reception is acknowledged for different time periods (i.e., 5 ms, 10 ms, etc.).

4. Coordination

The coordination method we present in this section is based on a coordination protocol, that is implemented on every robot of the team and is used by the robots in order to coordinate their activities for accomplishing a global team task.

The approach we adopted is a *formation/role* system similar to the ones described in Veloso and Stone (1998), Stone and Veloso (1999), Parker (2000), Werger and Mataric (2000). A *formation* decomposes the task space defining a set of *roles*. Each robot has the knowledge and capabilities necessary to play any role, therefore robots can switch their roles on the fly, if needed. Notice that, in an heterogeneous team, the implementation choices for each role can be different, thus having possibly different behaviors for a role, depending on the robot.

For instance, in the RoboCup environment, a basic formation could be the one where a robot takes the role of attacking going to get the ball control, another one that of defending and a third one that of supporting the attack. However, other formations are possible depending on the kind of strategy adopted (offensive, defensive) and on the need to handle special situations such as for example the malfunctioning of the goal keeper.

The coordination protocol is used by the robots in order to select the appropriate formation according to the environment conditions and to make a decision on the roles assumed by the robots in the formation.

4.1. *Coordination Protocol*

The coordination protocol is based on broadcast communication of some data by every robot. These data are processed by each robot in order to establish the formation that the team will adopt and the role assigned to the robot. The computation of the coordination protocol is distributed, because each robot must process the information coming from the others in order to identify the team formation and its own role. The protocol is robust because it relies on a little amount of transmitted data.

The coordination protocol is based on the concept of *utility functions*, that are defined off-line before the actual operation of the MRS in the environment. These functions are evaluated periodically during the robot mission and exchanged among the robots. The coordination protocol includes two steps that are periodically executed on-line during the MRS mission: role assignment and formation selection. In our current implementation the protocol is executed every 100 ms (that corresponds to the perception cycle). A less frequent execution of the protocol may lead to less reactivity in role exchanging.

In the following sections we will first describe the definition of the utility functions and then the two steps that define the coordination protocol.

4.2. *Definition of Utility Functions*

A set of *utility functions* is defined in order to provide a quantitative measure of the effectiveness estimated by each robot in assuming a given role within the current formation. A utility function is thus associated to one role and the coordination protocol is based on exchanging the values computed for these functions among the robots and on taking a distributed decision on role assignment from these values.

More formally, a utility function $f_r^i(..)$ for a robot i and a role r is a function that, given the information about the status of the robot, returns a value that indicates how well can this robot play the role r. In other words, a utility function for a role should return higher values when the robot is in a good situation to play the role, and lower values when the robot is in a bad situation to fulfill it.

The definition of the utility functions is an important step in the design and realization of the MRS, and, since they are deeply related to the application domain of the MRS, it is not easy to develop a general methodology for defining them. The designer of the MRS must anyhow take into account two considerations: (i) there are some variables or conditions that characterize the state of the robot that are relevant for the execution of the task associated to a role; (ii) some parameters of the utility functions must be experimentally evaluated, since they also depend from the characteristics of the individual robots (in an heterogeneous team).

Based on the above considerations we have performed the following steps for defining our utility functions:

1. identify the variables that are relevant for the execution of the task associated to a role;
2. define the utility function as a linear combination of these variables;
3. perform a set of systematic experiments in order to determine the coefficients of the utility functions.

For example, the utility function for the role *Attacker* in our RoboCup soccer robots is a linear combination of the following variables: distance to the ball, direction to the ball, distance to the opponent goal, direction to the opponent goal, presence of obstacles in the trajectory from the robot to the ball. The coefficients for this linear combination have been derived for every robot with the experiments described in Section 5. In particular we remark that the experiments with real robots have been fundamental for an effective coordination in our heterogeneous team, since due to the diversity of the robots, the coefficients of the utility functions are different for every robot.

4.3. Role Assignment

Role assignment is the first step of the coordination protocol and it is accomplished through explicit communication of the values of the utility functions (specific for every role), that every robot evaluates given its current local information about the environment. By comparing these values, each member of the team is able to establish the same set of assignments (robot ↔ role) to be immediately adopted.

More specifically, suppose we have n robots $\{R_1, \ldots, R_n\}$ and m roles $\{r_1, \ldots, r_m\}$. The roles are ordered by the MRS designer with respect to importance in the global task to be performed, i.e. assigning r_i has higher priority than assigning r_{i+1}. Moreover, for each role we define a "percentage of role covering" P_j, that denotes how many robots of the team should be assigned to this role. For example, if we have two roles and we want to assign the first role to 60% of the team robots and the second role to the remaining 40% of the robots we have $P_1 = 0.6$ and $P_2 = 0.4$. In this way it is possible to assign a role to a robot if $n > m$. The use of the percentage values P_j may require a little care when rounding them with respect to the total number of robots. A simple solution we adopted for our team is to define P_j as x/n (x being an integer value); while a different choice may be appropriate when a larger number of robots are present.

Let $f_j^i(\ )$ be the value of the utility function, computed by robot R_i for the role r_j and $A(i) = j$ denote that the role r_j is assigned to the robot R_i.

The method for dynamic role assignment requires that each robot R_p computes the following algorithm:

for each role r_j **do**
  compute and broadcast $f_j^p(\ )$;
for each robot R_i $(i \neq p)$
  for each role r_j **do**
    collect $f_j^i(\ )$**;**
$\mathcal{L} = \emptyset$**;** /* Empty the list of assigned robots */
for each role r_j **do**
  for c = 1 to $P_j \times n$ **do**
    begin
      $h = argmax_{(i \notin \mathcal{L})}\{f_j^i(\ )\}$ **;**
      /* the robot R_h has the highest value of $f_j^i(\ )(i \notin \mathcal{L})$ */
      if $h = p$ **then** $A(p) = j$**;** /* the role for R_p is r_j */
      $\mathcal{L} = \mathcal{L} \cup \{h\}$**;**
    end

It is easy to see that every role is assigned to at most $P_j \times n$ robots and that every robot is assigned to only one role. The reason is that at every cycle of the algorithm a different assignment $A(i) = j$ is done: j changes after $P_j \times n$ cycles and robots already included in the set $\mathcal{L}$ of assigned robots cannot be chosen for further assignments.

In particular, the first role (i.e., the one with the highest priority), will be assigned to those robots that have the best utility values for the role r_1, the second role to those among the remaining robots that have the best utility values for the second role, and so on. While in the case of a complete lack of communication all the robots will assume the most important role (r_1).

It is important to notice here that the utility functions we have defined in our approach do not represent a metric evaluation of the performance of the MRS (as in Parker (2000)), but the best way to accomplish every single task (independently from the others). Therefore, even though our algorithm can be considered as a greedy approach to the *Heterogeneous Robot Action Selection Problem (HRASP)* defined in Parker (2000), we remark that our utility functions are not semantically equivalent to the metric evaluation functions. Moreover, while HRASP has been proved to be NP-hard (Parker, 2000) and thus heuristics must be developed to cope with this problem, our objective is not to compute a role assignment that maximize the sum of the utility functions, but to assign a priority to every role and to guarantee that high priority roles are assigned to the robots that are in the best conditions to perform the associated tasks.

This issue is very important in many dynamic and adversarial environments and especially in the RoboCup environment, in which this work has been originally developed. Consider a situation in the RoboCup environment in which two robots coordinate on the two roles: *attacker* and *defender*. For an effective game it is important that the role *attacker* is assigned to the robot that is in the best condition to reach the ball before the opponents. Now consider the following values for the utility functions of these two roles: $f_a^1(..) = 10$, $f_d^1(..) = 6$ for robot 1 and $f_a^2(..) = 6$, $f_d^2(..) = 1$ for robot 2. Our algorithm will assign robot 1 to the role *attacker* and robot 2 to the *defender*, while if we had chosen

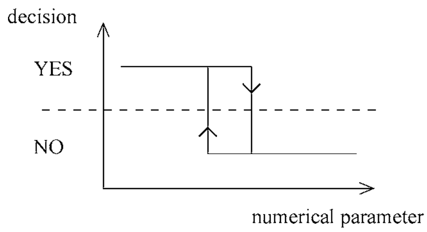

Figure 2. The hysteresis mechanism in decisions.

to maximize the sum of the utility functions we would have the opposite assignment. However, in this scenario, the second choice may be ineffective since it does not take into account the different priorities of the two roles and may lead to a situation in which robot 2 will fail to reach the ball and to act as *attacker*, while robot 1 (that was in the best condition to get the ball and attack) will perform a defensive action possibly loosing the opportunity to control the ball.

In order to obtain an effective application of the above method, an important issue to be dealt with is the stability of decisions with respect to possible oscillations of the numerical parameters on which they depend upon (see also (Hannebauer et al., 1998)).

The method we adopted to stabilize decisions is based on the notion of *hysteresis* (see Fig. 2), which amounts to smoothing the changes in the parameter values. This technique prevents a numerical parameter's oscillation from causing oscillations in high level decisions. In the case of coordination, for instance, if at a certain instant robot R_i covers role r_j, its utility function $f_j^i(\ )$ for role r_j returns a higher value.

Another critical factor for the correct operation of coordination is the capability of each element to realize a sudden difficulty in performing its task. For instance, a robot that is moving toward the ball can get stuck on its way. Once it has realized this circumstance, all its utility functions must return low values so that the the role can be assigned to other robots.

Finally, if all the robots possess the same data (i.e., communications are working correctly), they will compute the same assignment, but in case of a great loss of transmitted data due to interferences, the robots may have slightly inconsistent data. Therefore, there could be roles temporarily assigned to more than one robot or not assigned at all. However, holes in data transmission last in general fractions of a second. So, if we assume that the values of the "utility functions" do not change sharply, the correct use of the hysteresis method guarantees that the roles will be correctly assigned almost always (as shown by the experimental data we have collected during the games and are discussed in the next section).

4.4. Formation Selection

The robots have at their disposal a number of predefined formations and rules to select the formation to adopt, on the basis of the environment configuration. Since each robot status do not necessarily coincide with those of the others (because of possible communication failures or different views of the world), the robots may choose different formations. Therefore, the formation selection algorithm is based on a voting scheme that allows for changing the formation only in presence of the absolute majority of votes.

for each robot R_i of the team **do**
  begin
    collect $voted\_formation[i]$**;**
    $votes[voted\_formation[i]] =$
      $votes[voted\_formation[i]] + 1$**;**
  end
if there is a formation f such that $votes[f] > n/2$
  then
  $selected\_formation = f$**;**

This voting scheme resolves the conflict arising by lack of absolute majority by leaving the formation unchanged. The algorithm also limits significantly the risk of oscillations in formation selection as well as the frequency of changes. Moreover, to ensure the stability of decisions, the formation selection is accomplished at a lower frequency to that of role selection (i.e., once per second).

5. Applications and Experiments

A successful coordination of the team depends on the effectiveness of the coordination protocol and on a suitable calibration of some parameters, such as the coefficients of the utility functions. Calibration of these parameters typically requires a significant experimental work.

In addition, in an heterogeneous team of robots the experimental phase is particularly demanding, since the exchanged information are computed and interpreted differently by each element of the team. For example, consider the evaluation of reachability of the ball: each robot may have a mobile base with different capabilities and a set of behaviors with specific speed characteristics and, that notwithstanding, robots must calculate comparable numerical values.

In this section we describe methods and tools that have been used for developing and evaluating the approach described above and we present the results of our experiments in a competitive and highly dynamic environment such as the one provided by the RoboCup matches.

5.1. Methods and Tools

The experimentation of the coordination protocol must be done in stages which require the use of different tools: a simulator, experimentation without play, experimentation during actual games, and analysis of log files of the games.

The first and easier experimental setting is provided by a simulator. Even though simulation cannot provide a precise characterization of all the aspects that influence the performance of the robot in the real environment, it is very useful both for verifying the correctness of the protocol and for computing a first estimation of the coefficients of the utility functions, that will be refined with subsequent experiments involving real robots.

First experiments with real robots may be done without playing, with steady robots and moving the ball and opponents. At this stage one needs to adjust the discrepancies arising from differences in heterogeneous robots' implementations. In particular, we compare the sensing capabilities of the robots and adjust the coefficients of their utility functions. Notice that while these aspects could be, in principle, resolved through the simulator, this requires a rather complex simulation model that is very difficult to build. Similarly, learning techniques could be used, but they are outside the scope of the present work.

The other experimental phase involving the robots consists in looking at the game and in singling out the failures of the coordination system. For example, a typical task is that of identifying situations where the most suitable player does not move towards the ball (take role *Attacker*) and adjust the parameters to restore the expected behavior.

To this end, an analysis of the log files generated during the games is very useful for identifying misbehaviors of the coordination system. In this respect, we have developed a 3D viewer for experimental data that allows for displaying several information about one or more robots from the log files of real games or portion of them. By analyzing the data collected by the robots during the game through our graphical tool for 3D navigation of these data, we are able to detect several interesting features of coordination as well as unexpected behaviors of the robots, such as unassigned roles, oscillations in role exchange, overlapping roles, etc. Specifically, we have used this tool also to further refine the utility functions.

5.2. The RoboCup Environment

The coordination method presented in the previous section has been implemented within the Azzurra Robot Team, the Italian national team of heterogeneous robotic soccer players, participating in the RoboCup competitions in the Middle-Size League (Nardi et al., 1999).

Coordination in the RoboCup environment is an important issue because of the possibility of implementing effective team strategies for the game. In fact, coordination among the ART players requires not only a distributed coordination protocol, but also a very flexible one, that allows the coach to accommodate the various configurations that can arise by forming teams with different basic features. The performance of the ART team provided substantial evidence that basic coordination among the team players has been successfully achieved. In several situations where two team players were close to the ball, they were able to smoothly switch their roles and managed to get ball possession without obstructing each other; in addition, they have generally occupied the field in a satisfactory way, as shown in the coordination analysis presented in Section 5.4.

The coordination protocol used in this setting is based on a set of formations and a set of roles for every formation. The formation that has been mostly used is the *standard formation* that is described below. Other formations have been considered in order to deal with special situations, like goalkeeper out of the game.

In the *standard formation* we have defined 3 roles for the 3 players: *Attacker*, *Defender*, *Support*. The *utility functions* for these roles are defined as a linear combination of distance from the ball, position and orientation of the robot in the field, and obstacles in the path towards the ball. Since we have 3 roles for 3 robots the *percentage role covering* P_j are set to $1/3$, so that every role is assigned to one robot. Moreover, the roles have priorities and thus if for example one robot is out or does not communicate with the others the first two roles (*Attacker*, *Defender*) are assigned to the remaining two robots, leaving the *Support* role unassigned.

The algorithm for dynamic role assignment has been used for effective role switching between robots during the games and the hysteresis in the utility functions have reduced the possibility of too frequent role switching. The performance has also been evaluated through a set of systematic experiments on the robots and by the data collected during actual games in the official games of RoboCup 1999 and 2000. We have reported the results of the analysis on communication and coordination in the next sections.

5.3. *Communication Analysis*

The reliability of communication with the EIEP has been experimentally verified. ETHNOS system allocates a maximum guaranteed and dedicated time to network communication. Since ETHNOS schedules all tasks in real-time according to the Rate Monotonic scheduling policy, the dedicated time value is computed automatically on the basis of the schedulability analysis so that the real time execution of the whole set of tasks (i.e., user-defined and communication tasks) is guaranteed. Thus, the dedicated time depends on the computational load of the tasks in execution as well as on the processor speed.

In Fig. 3 the diagrams represent different machines (with different processing power) corresponding to two robots (Relé—an AMD-K6 200 MHz—and Homer—a AMD-K6 233 MHz—) and a monitor (an Intel Pentium II 300 MHz), connected using radio ethernet (2 Mbit bandwidth Wavelans). The top line in the diagram indicates the calculated time available each 100 ms for communication purposes. Clearly, this value decreases as the number of tasks in execution increases (and, consequently, the computational load). The bottom line indicates the time spent in communication, which also increases with the number of experts (this is because in this experiment we have assumed that the activity of every expert involves either transmission or reception of messages). In this way it is always possible to determine a priori whether the system is capable of both communicating information and executing in real time.

Finally, it is worth considering the transmission complexity which determines the bandwidth requirements. In general, if we have a system composed of n robots and m roles we have a complexity $O(mn)$ corresponding to the number of utility values that have to be transmitted. In fact, according to ETHNOS broadcast communication strategy, if the number of roles is fixed the complexity linearly increases with respect to the number of robots and viceversa. This typically occurs in many applications in which the number of roles is significantly lower than the number of robots and it is fixed and related to the specific application domain. In addition, the size of messages is limited and the bandwidth required is thus very low.

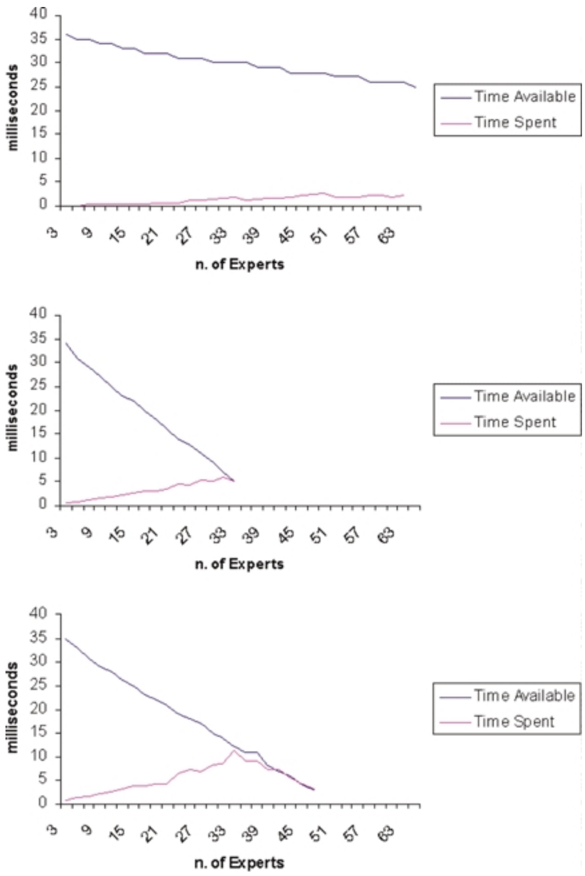

Figure 3. Network communication in ETHNOS. From top to bottom: the monitor, Relé, Homer.

5.4. *Coordination Analysis*

The evaluation of the coordination protocol in this setting has been carried out by analyzing a set of log files acquired during the matches in the European Championship 2000. We present here the results of this

Table 1. Robots' roles.

Match	Attacker (%)	Support (%)	Defender (%)
1	82.9	39.5	98.2
2	84.6	98.0	80.8
3	87.6	38.5	90.0
4	89.5	81.8	96.9
5	93.5	84.1	98.9
Avg.	87.6	68.3	93.0

analysis both for the communication and for the coordination layer.

A quantitative analysis of the coordination protocol has been worked out through the collection of the log files of various games.

Table 1 describes the percentage of time in which a role is assigned to at least one robot. Since the roles are ordered in terms of importance within a formation, some roles are more likely to be assigned. In particular *Attacker* and *Defender* were almost always assigned, while for instance the role *Support* was not assigned when there were only two players in the field. Therefore, with respect to a static assignment of roles, dynamic assignment provides a good distribution of the roles, but with the advantage of selecting the more appropriate robot for every role depending on the current situation of the environment.

Table 2 shows the coverage of the field by the three middlefield robots (i.e., the percentage of time in which at least one robot was in a zone of the field). It is interesting to notice that the field has been properly covered, even if there is not an explicit split of the field in sections assigned to the robots. In particular, the defensive area has been occupied by at least one robot almost at every time.

Another analysis shows that during the game there is an average of one role switch every 10 seconds. Due to occasional loss of transmitted data, we noticed that about 1/10 of the role switches generates roles' oscillation, lasting about 300 ms before stabilization.

Table 2. Robot position in the field.

Match	Forward (%)	Middle (%)	Backward (%)
1	78.8	68.7	97.8
2	65.4	68.8	98.6
3	53.2	54.5	99.3
4	23.5	80.4	93.1
5	64.1	72.1	98.9
Avg.	57.0	68.9	97.5

Finally, by a comparison between the logs and a visual review of the game, we have discovered that one relatively frequent source of failure in coordination arises from situations in which a robot is not actually able to correctly evaluate the utility function for a certain role. This situation arise when there is a problem in the vision system which causes the robot to see the ball very close to it, while it is not. This robot gets the role *Attacker* with the goal of going to the ball, but it will be unsuccessful while blocking the exchange of roles. This behavior shows that the coordinated system is highly influenced by the individual robot failure. In such cases the overall performance of the team is not good, since a robot that is actually in a better position to go will not assume the correct role. The problem has been successfully addressed by decreasing the utility function when the robot does not make progress in its task. This is obtained by a task-dependent evaluation of the progress of the robot in performing the task. For example, the evaluation of the progress of the task associated to our role *Attacker* is given by the computation of the distance to the ball that must decrease in time. If the distance to the ball does not decrease the utility function for the role *Attacker* returns a lower value, allowing other robots to possibly take the role.

6. Related Work

Recently, there has been a growing interest in multi-agent robotic systems, as witnessed by the great number of scientific publications in the field (for a survey see (Dudek et al., 1996; Cao et al., 1997; Iocchi et al., 2001; Balch and Parker, 2002)). Research has mostly been focused on specific aspects (distributed system architectures, communication modalities, coordinated motion, cooperation and competition, etc.), on particular environments (military (Noreils, 1992; Lee et al., 1994), service (Everett and Gage, 1996; Evans, 1995), extra-terrestrial planetary exploration domain (Miller, 1990), etc.).

In this section we identify those aspects of the MRS discussed in the literature that are more closely related to our work. Subsequently, we will address RoboCup related approaches.

From the architectural point of view our approach requires only a common communication layer, whereas significantly different architectural models can be adopted. This is witnessed by the different architectures (behaviors based, hybrid, fuzzy-logic based) of the robots in the ART team. Other approaches impose more rigid constraints. For example in Brooks et al. (1999) and Parker (1993) subsumption-based reactive control is adopted for robot societies consisting of up to 20 agents. ALLIANCE (Parker, 1998) and BLE (Werger and Mataric, 2000) provide coordination among the robots by an inhibition mechanism that is used by one robot to disable the execution of a behavior on other robots. Robots in these systems are designed in a behavior-based architecture. A more general approach is proposed in Arkin et al. (1999) and Balch and Arkin (1998) where the AuRa hybrid architectural model is extended to handle multi robot coordination in hostile environment exploration for urban warfare and for navigation in military formation. Also the MURDOCH architecture (Gerkey and Matarić, 2000) defines a distributed communication system based on a publish/subscribe paradigm that may be implemented on different robotic architectures.

Heterogeneity is a fundamental feature of our approach and in fact the robots in the ART team differ not only in the control architecture, but also in the mechanics, sensors and actuators as well as in the robot shape and appearance. It is important to emphasize that in our approach heterogeneity is a requirement that the robotic system must handle (the robots to be coordinated have been developed independently and beforehand by separate research groups) and that every robot is able to perform every task, but possibly with different performance. Other approaches in literature have typically dealt with heterogeneity in a different way. For example in Simmons et al. (2000) and Parker (2000). heterogeneity is specifically designed and adopted to better accomplish a given task.

Our approach differs from many other proposed systems also in the role of coordination to address the given task. In fact we are not simply interested in finding a spatial distribution of the robots in order to optimize the execution of a cooperative task (as in typical coverage problems (Fontan and Mataric, 1998; 2001), such as mine collecting (Acar et al., 2001) or floor cleaning (Jung and Zelinsky, 2000)); instead, we aim at finding a solution for the more general problem, where robots are given a set of criteria and an arbitration algorithm to negotiate the role that each of them must assume (in our definition, selecting a role means choosing and activating the group of concurrent behaviors that are associated with that specific role). This problem, often related to the presence of a robotic or human opponent, has been considered relevant in military applications for Urban Warfare. In this scenario a group of robots are deployed into an unknown, possibly hostile environment (for example a building occupied by enemy forces) with the purpose of exploring it and report to a base station what they have discovered (as in the DARPA Tactical Mobile Robot domain (Blitch, 1999)).

The problem of dynamic task assignment has been specifically addressed also in some recent works (Werger and Mataric, 2000; Gerkey and Matarić, 2000; Uchibe et al., 2001; Shen and Salemi, 2002). The work in Werger and Mataric (2000) describes an approach to distributed task assignment similar to the one presented here. In order to assign a task every robot collects a value (that can be seen as the result of our utility functions) from all the other robots, the robot with the maximum value can send an inhibition signal to the other robots in order to disable their execution of the behavior associated to this task. As already mentioned before, this mechanism is based on a Port-Arbitrated Behavior-Based architecture. In the MURDOCH system (Gerkey and Matarić, 2000) tasks are organized in a structure similar to a Hierarchical Task Network and task allocation is computed by a sequence of "one-round auctions". When a task is to be executed by the MRS an auction mechanism allows for deciding which robot is in the best condition to perform the task by means of the evaluation of some *metric functions*, that are similar to our utility functions. A difference with respect to our system is that we evaluate our utility functions periodically during the mission, so that if one robot happens to be in a better condition to do a task a role switching is performed. The work in Uchibe et al. (2001) addresses the issue of "module conflict", that is needed for avoiding that two or more robots execute incompatible tasks. The authors define the concept of *module*, that is slightly different from our definition of *role*. While their modules are associated to the basic tasks that can be executed by the robots, our roles define high-level tasks that must be accomplished by the robots (possibly executing simpler subtasks). Therefore, the association between robots and roles is one-to-one in our approach, while it is many-to-many in Uchibe et al. (2001). The module conflict resolution mechanism in Uchibe et al. (2001), that is used for dynamic task assignment, is based on the evaluation of a correlation function that defines incompatibility among the modules. This function is defined by the user for every pair of modules. Finally, in Shen and Salemi (2002) tasks are organized in Distributed Organizational Task Networks (DOTN) and task allocation is performed by a heuristic search in the space of possible assignments between robots and DOTN. Task assignment is not based on the evaluation of a function for measuring the quality of task allocation, but on minimizing the dependencies among the tasks assigned to different robots, so that each robot can perform its tasks in a more independent way.

Among the proposals for robot coordination developed in the RoboCup environment, we may find both centralized and distributed approaches. Centralized ones are mostly adopted in the Small-size League where a global reconstruction of the environment is possible (because of the availability of a global vision system) and when a team of robots can reliably reconstruct global information about the environment from local sensors. For example the CS Freiburg team in the Middle-size League (Gutmann et al., 1999) makes use of very precise measurements given by laser scanners to reconstruct a global representation of the field and a centralized coordination is realized by dynamic selection of the robot that acts as the leader. However, possible communication failures, that are very common in the RoboCup environment, as well as the general difficulty of reconstructing a global reliable view of the environment, require full autonomy on each robot and distributed approaches are usually preferred in this context. A distributed approach that is not based on explicit communication is described in Ferraresso et al. (2000), in which two robots are able to exchange the ball by using direct communication of their roles and a field-vector based collision avoidance that takes into account in a proper way the role of the robots. Other behavior based reactive MRS are presented in Brendenfeld and Kobialka (2000) and Carpin et al. (2000). Our approach exhibits the following two features: on one hand, it exploits explicit communication among the robots for exchanging information about the status of the environment in order to achieve a distributed agreement on the actions to be performed; on the other hand, it is robust to possible communication failures, since in these cases the system degrades its performance, but it still keeps on the execution of the task.

7. Conclusions

In this article we have described an approach to distributed coordination of a team of heterogeneous mobile robots, whose main features are: explicit communication among the robots, autonomy in coordination and heterogeneity of robots with similar capabilities. The approach was originally developed for the RoboCup environment, and it is thus suitable for highly dynamic and hostile environments.

The successful application of the method has proved its effectiveness in a challenging scenario for MRS coordination. In fact, the coordination method described here has been a critical factor in the performance of the ART robot team for the Stockolm 1999 and Amsterdam 2000 RoboCup competitions, where the ART has classified in the second place for both of these competitions. ART soccer team was a highly heterogeneous team, since different players were designed by different research groups exploring different hardware and software solutions. The proposed system architecture allowed to easily integrate together the results of different researchers across different groups and to easily implement protocols for inter-robot coordination (by hiding to the programmers the details of communication). It is important to emphasize that in our approach heterogeneity is a requirement that the robotic system must handle (the robots to be coordinated have been developed independently and beforehand by separate research groups). Moreover, as a difference with other heterogeneous MRS (Simmons et al., 2000; Parker, 2000), in our case all the robots are able to perform all the tasks, but with different performance that is taken into account in the coordination protocol. We believe that the availability of robots with different features and designed by different manufacturers is a realistic scenario in the next future.

Another important contribution of this article is in the definition of methods and tools for an experimental evaluation of the performance of the proposed coordination method. In particular the analysis of communication packets has proved the effectiveness and reliability of the EIEP for robots' information exchanging, while the analysis on the coordination protocol has showed that several properties (like area coverage and role assignment), that are important for the environment in which coordination has been applied, have been substantially achieved.

One critical aspect of distributed coordination in a MRS is the robustness of the method, with respect to failures and malfunctioning of the robots. The method described in this paper made the team sufficiently robust to perform successfully during the competitions. However we believe that the methods for achieving robustness of the team performance deserve further investigation. Another important issue is the evaluation of the effectiveness of coordination methods. We are currently addressing this aspect both by improving the techniques and tools proposed in the paper and by addressing the analysis of a wider set of properties that are related to coordination and team work.

Acknowledgments

We would like to thank all the members of ART, who contributed to the development of the team. The experimentation of robotic team work has been greatly supported by the whole team. We would like to thank Claudio Castelpietra and Alessandro Scalzo particularly for their contribution to the implementation of the proposed method. We acknowledge the support of Consorzio Padova Ricerche, CNR, University of Rome "La Sapienza", and University of Genova.

References

Acar, E., Zhang, Y., Choset, H., Schervish, M., Costa, A.G., Melamud, R., Lean, D., and Graveline, A. 2001. Path planning for robotic demining and development of a test platform, In *Field and Service Robotics*.

Arkin, R., Collins, T., and Endo, T. 1999. Tactical mobile robot mission specification and execution, In *Mobile Robots XIV*, Boston, USA, pp. 150–163.

Balch, T. 1999. The impact of diversity on performance in multi-robot foraging, In *Proc. of Agents'99*.

Balch, T. 2000. Hierarchic social entropy: An information theoretic measure of robot team diversity. *Autonomous Robots*, 8(3).

Balch, T. and Arkin, R.C. 1995. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52.

Balch, T. and Arkin, R.C. 1998. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6).

Balch, T. and Parker, L.E. (Eds.). 2002. *Robot Teams: From Diversity to Polymorphism*, A K Peters Ltd.

Blitch, J. 1999. Tactical mobile robots for complex urban environments. In *Mobile Robots XIV*, Boston, USA, pp. 116–128.

Brendenfeld, A. and Kobialka, H.U. 2000. Team cooperation using dual dynamics. In *Proc. of ECAI2000 Workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems*.

Brooks, R., Maes, P., Mataric, M., and More, G. 1990. Lunar base construction robots. In *IEEE International Workshop on Intelligent Robots and Systems*, Tsuchiura, Japan, pp. 389–392.

Cao, Y.U., Fukunaga, A., and Kahng, A. 1997. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:1–23.

Carpin, S., Ferrari, C., Montesello, F., Pagello, E., and Patuelli, P. 2000. Scalable deliberative procedures for efficient multi-robot coordination. In *Proc. of ECAI2000 Workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems*.

Castelpietra, C., Iocchi, L., Nardi, D., Piaggio, M., Scalzo, A., and Sgorbissa, A. 2000. Coordination among heterogenous robotic soccer players. In *Proc. of International Conference on Intelligent Robots and Systems (IROS'2000)*.

Choset, H. 2001. Coverage for robotics—A survey on recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126.

Dudek, D., Jenkin, M., Milios, E., and Wilkes, D. 1996. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397.

Engelore, R. and T.M. (Eds.). 1988. *Blackboard Systems*. Addison-Wesley.

Evans, J. 1995. HelpMate: A service robot success story. *ServiceRobot: An International Journal*, 1(1):19–21.

Everett, H.R. and Gage, D. 1996, A third generation security robot. In *SPIE Mobile Robot and Automated Vehicle Control Systems*, Boston, USA, pp. 20–21.

Ferraresso, M., Ferrari, C., Pagello, E., Polesel, R., Rosati, R., Speranzon, A., and Zanette, W. 2000. Collaborative emergent actions between real soccer robots. In *RoboCup-2000: Robot Soccer World Cup IV*.

Fontan, M. and Mataric, M. 1998. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 15(5).

Gerkey, B.P. and Matarić, M.J. 2000. 'Principled communication for dynamic multi-robot task allocation'. In *Proceedings of the International Symposium on Experimental Robotics*, Waikiki, Hawaii.

Guibas, L.J., Latombe, J., LaValle, S., and Lin, D. 1999. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9:471–494.

Gutmann, J.-S., Weigel, T., and Nebel, B. 1999. Fast, accurate, and robust self-localization in the RoboCup environment. In *RoboCup-99: Robot Soccer World Cup III*, pp. 304–317.

Hannebauer, M., Wendler, J., Gugenberger, P., and Burkhard, H. 1998. Emergent cooperation in a virtual soccer environment. In *Proc. of DARS-98*.

Iocchi, L., Nardi, D., and Salerno, M. 2001. Reactivity and deliberation: A survey on multi-robot systems. In E.P.M. Hannebauer and J. Wendler (Ed.), *Balancing Reactivity and Deliberation in Multi-Agent Systems (LNAI 2103)*, Springer, pp. 9–32.

Jung, D. and Zelinsky, A. 2000. Grounded symbolic communication between heterogenoeus cooperating robots. *Autonomous Robots*, 8(3).

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., and Matsubara, H. 1998. RoboCup: A challenge problem for AI and robotics. In *Lecture Note in Artificial Intelligence*, Vol. 1395, pp. 1–19.

Konolige, K., Myers, K., Ruspini, E., and Saffiotti A. 1997. The saphira architecture: A design for autonomy. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(1):215–235.

Lee, J., Huber, M., Durfee, E., and Kenny, P. 1994. M-PRS: An implementation of the procedural reasoning system for multirobot applications. In *AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space*.

Miller, D. 1990. Multiple behavior-controlled MicroRobots for planetary surface missions. In *IEEE International Conference on Systems, Man, and Cybernetics*, Los Angeles, USA, pp. 289–292.

Nardi, D., Adorni, G., Bonarini, A., Chella, A., Clemente, G., Pagello, E., and Piaggio, M. 1999. ART-99: Azzurra Robot Team. In *RoboCup-992: Robot Soccer World Cup III*.

Noreils, F.R. 1992. Battlefield strategies and coordination between mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1777–1784.

Parker, L. 1993. Adaptive action selection for cooperative agent teams. In *Proc. of 2nd International Conference on the*

Simulation of Adaptive Behavior. Honolulu, USA, pp. 442–450.

Parker, L.E. 1998. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240.

Parker, L.E. 2000. Lifelong adaption in heterogeneous multi-robot teams: Response to continual variation in individual robot performance. *Autonomous Robots*, 8(3):239–267.

Piaggio, M., Sgorbissa, A., and Zaccaria, R. 2000a. Pre-emptive versus non-pre-emptive real time scheduling in intelligent mobile robotics. *Journal of Experimental and Theoretical Artificial Intelligence*, 12:235–245.

Piaggio, M., Sgorbissa, A., and Zaccaria, R. 2000b. A programming environment for real time control of distributed multiple robotic systems. *Advanced Robotics*, 14(1):75–86.

Piaggio, M., Sgorbissa, A., and Zaccaria, R. 2001. Autonomous navigation and localization in service mobile robotics. In *Proc. of IROS'01*.

Piaggio, M. and Zaccaria, R. 1997. An information exchange protocol in a multi-layer distributed architecture. In *IEEE Proc. Hawaii International Conference on Complex Systems*.

Shen, W.M. and Salemi, B. 2002. Distributed and dynamic task reallocation in robot organizations. In *Prooceedings of the 2002 IEEE Int. Conference on Robotics and Automation*. Washington, DC.

Simmons, R., Apfelbaum, D., Fox, D., Goldman, R.P., Haigh, K.Z., Musliner, D.J., Pelican, M., and Thrun, S. 2000. Coordinated deployment of multiple, heterogeneous robots. In *Proc. of International Conference on Intelligent Robots and Systems (IROS'2000)*.

Stone, P. and Veloso, M. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241–273.

Uchibe, E., Kato, T., Asada, M., and Hosoda, K. 2001. Dynamic task assignment in a multiagent/multitask environment based on module conflict resolution. In *Proced. of the 2001 IEEE International Conference on Robotics and Automation*. Seoul, Korea.

Veloso, M. and Stone, P. 1998. Individual and collaborative behaviors in a team of homogeneous robotic soccer agents. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pp. 309–316.

Werger, B.B. and Mataric, M.J. 2000. Broadcast of local eligibility for multi-target observation. In *Proc. of DARS*.

Luca Iocchi is Researcher at University of Rome "La Sapienza". His main research interests in the area of cognitive robotics and mobile robotics are action planning, self-localication, and coordination among multiple robots. He has attended many RoboCup competitions within the Italian team and as team leader of the team developed at University of Rome.

Daniele Nardi is Full Professor of Computer Science at the Faculty of Engineering, Univ. Roma "La Sapienza", Dept. Informatica e Sistemistica. His research interests include theoretical aspects of knowledge representation and reasoning, such as description logics and non-monotonic reasoning, as well as applications of knowledge-based systems in Cognitive Robotics and Information Integration. He is a member of IEEE, ACM, AIIA, AICA.

Maurizio Piaggio was born in Genova, Italy on April 1, 1970. He received the M.S. degree in Electronical Engineering from the University of Genova in 1993. In that year he worked at the LIFIA in Grenoble, France on robot manipulation and compliant control. In 1998 he received his Ph.D. in Computer Science in Genova in the field of robot software architectures. He is currently working at the Department of Communication, Computer and System Sciences of the University of Genoa. His current research interests are on robot architectures, mobilerobots, navigation and planning.

Antonio Sgorbissa was born in Genova, Italy on December 20, 1970. He received the M.S. degree in Electronical Engineering from the University of Genova in 1997.

In 2000 he received his Ph.D. in Computer Science in Genova in the field of autonomous mobile robotics. He is currently a research assistant at the Department of Communication, Computer and System Sciences of the University of Genoa.

His current research interests are on mobile robots navigation and localization, and real-time systems.