Coordination among Heterogeneous Robotic Soccer Players

C. Castelpietra[1], **L. Iocchi**[1], **D. Nardi**[1]
M. Piaggio[2], **A. Scalzo**[2], **A. Sgorbissa**[2]

(1) Dip. di Informatica e Sistemistica
Università "La Sapienza", Roma, Italy
<castelp,iocchi,nardi>@dis.uniroma1.it

(2) Dip. di Informatica, Sistemistica e Telematica
Università di Genova, Italy
<piaggio,scalzo,sgorbiss>@dist.unige.it

Abstract

Coordination among multiple robots has been extensively studied, since a number of practical tasks can be performed in a more effective way by employing a fleet of coordinated robotic bases. In particular, distributed coordination among *robotic agents* has been considered recently within the framework offered by the robotic soccer competitions.

In this paper we describe the methods and the results achieved in coordinating the players of the ART team participating in the RoboCup F-2000 league. The team is formed by several heterogeneous robots having different mechanics, different sensors, different control software, and, in general, different abilities for playing soccer. The coordination framework we have developed has been successfully applied during the 1999 official competitions allowing both for a significant improvement of the overall team performance and for a complete interchangeability of all the robots.

1 Introduction

Using multiple robots for accomplishing a given task has been shown faster and more effective [2], since they can be in different places at the same time, they can perform concurrent and cooperative actions and, in general, they allow for a decomposition of a complex task into simpler sub-tasks.

The literature shows several models and proposals for implementing multi-agent systems, but relatively few approaches have been applied to actual robotic agents. An analysis of the work in this field shows that different coordination strategies are to be chosen according to the particular environment (see [6] for a survey). For instance, the studies in robotic scouting teams working in unstructured environments lead to investigations in behaviors to achieve desired formations [3, 2], distributed planning is recommended when the robotic team has to compete for limited resources [1] or in structured and not very dynamic environments. In any case, distributed planning is suitable only if communication is reliable and the cost of communication relative to other actions is small. It has also been argued that robot teams working in *highly dynamic environments* get low benefit from complex plan negotiation. This is why, in this field, some researchers propose to eliminate any kind of negotiation in coordination [12].

Distributed coordination of *robotic agents* has been considered as one of the central research issues in the RoboCup competitions [4]. In a highly dynamic and uncertain environment such as the one provided by RoboCup games, the centralized coordination of activities underlying much of the work in Robotics does not seem to be adequate. In particular, the possible communication failures as well as the difficulty of constructing a global reliable view of the environment, require full autonomy on each robot.

In the RoboCup simulation league [5, 7] coordination plays a central role because of the high number of players (11). In the small size league coordination can take advantage of the availability of global information on the game status, since a centralized vision system and elaboration is used [11]. In the F-2000 (middle size) league, although the number of players is 4, coordination among the players is still a critical issue because the dynamics of the game make it necessary to avoid interferences among players of the same team aiming at gaining possession of the ball and because a player, in general, cannot score in a single action unless the ball is close enough to the opponents' goal. However, the distinguishing feature of the F-2000 league is the difficulty of reconstructing global information about the environment. Therefore, in the F-2000 league coordination needs to be achieved without laying down drastic prerequisites on the knowledge of the single players, but typically rely-

Proc. of International Conference on Intelligent Robots and Systems (IROS), 2000

ing on explicit communication [13]. However, due to the frequent communication failures the robots must depend neither on communication, nor on information provided by other robots.

The **ART team** [8] is composed by different elements developed in various Italian universities. Because of this kind of organization the players differ in the hardware and in the software. Consequently, 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.

Summarizing the hypotheses underlying the coordination problem in F-2000 category are: (1) *Communication-based coordination:* exploit the use of communication among the players to improve team performances, allowing the robots to acquire more information and to self-organize in a more reliable way; (2) *Autonomy in coordination:* the players are capable to perform their task, possibly in a degraded way, even in case of lack of communication; (3) *Heterogeneity in the multi agent system:* the players are heterogeneous both from hardware and software viewpoints.

Besides the above constraints, coordination in ART has been designed to deal both with roles (defender, attacker etc.) and with strategy (defensive, offensive). While the strategic level is currently demanded to an external selection (the human coach), roles are dynamically assigned (see [11]) to the various team elements during the game, depending on the configuration present on the field.

In the following sections we first consider the communication infrastructure, we then present the coordination protocol, discuss the experimental results on coordination of the ART players and finally draw some conclusions.

2 Communication infrastructure

The communication infrastructure is strictly related to the ETHNOS [9] software architecture whose protocol was at the base of inter-robot communication in ART. ETHNOS exploits a message based communication protocol called EIEP (Expert Information Exchange Protocol) which deals transparently both with inter process communication within a single robot and with inter-robot communication. Messages are transparently exchanged based on a publish/subscribe technique. For example a typical configuration that we have used allowed the robots in the team to communicate among themselves and and with an external supervisor (the coach).

It is worth noticing that, because of EIEP protocol, whenever we want to add (remove) a player to (from) the team, it is not necessary to explicitly inform each player about the modifications in the team's composition. In fact, the players just have to agree about the type of information they are ready to send and receive and ETHNOS automatically updates the necessary local and global databases. This has been very important in ART in which there were more than four types of robots available, with different characteristics of play, and thus different team compositions were selected for the single matches and also modified during the matches to better contrast the opponent. Moreover since in the RoboCup (and in general in mobile robotics) network communication is often wireless, 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. For this reason we have also designed a protocol for this type of applications, called EEUDP (Ethnos Extended UDP), that allows the transmission of messages with different priorities. The minimum priority corresponds to the basic UDP (there is no guarantee on the message arrival) and should be 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 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.).

3 Coordination

The major issues that we have addressed in the coordination protocol are the dynamic assignment of roles and the team strategy.

We adopted a *formation/role* system similar to the one described in [11, 10]. A *formation* decomposes the task space defining a set of *roles*. Each robot has the knowledge necessary to play any role, therefore robots can switch roles on the fly, if needed. However, the implementation choices are different due to the difference in the application domain: in the simulation league the focus is on communication failures, while in the F-2000 league the use of this kind of coordination is justified by the following experimental observations: (i) if more than one robot of the same team tries to perform the same action (f.i. to go to the ball) they very likely obstruct each other, worsening the team's performance; (ii) in order to win it is necessary to occupy properly various parts of the field. For instance not all the robots should attack: some must stay back to defend, otherwise they will concede many goals.

Therefore, the basic formation of an F-2000 team requires that one robot takes the role of going to the ball, another that of defending and another 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.

3.1 Coordination protocol

Inside each robot, the coordination module takes its input from the coordination messages. The output is the formation that the team will adopt and the role assigned to the robot.

The computation for the coordination protocol is distributed. The protocol is robust because it relies on a little amount of transmitted data. The coordination protocol includes the following two steps:

Step 1: Formation selection

The robots have at their disposal a number of predefined formations and possess the rules to select the formation to adopt, on the basis of the environment configuration. Since each robot's data do not necessarily coincide with those of the others, the robots may choose different formations and therefore each robot adopts the formation that gets the majority of votes.

Step 2: Role assignment

This step implements dynamic role assignment through explicit communication of the "utility" for the team that each robot will assume a certain role. This is achieved by the definition of a number of *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 $\{R_1, ..., R_n\}$ and m roles $\{r_1, ..., r_m\}$. The roles are ordered with respect to importance, i.e. in the current formation assigning r_i is more important than assigning r_{i+1}. This means that if $n < m$ then only the first n roles will be assigned, while if $n > m$ then $m - n$ robots will not be assigned any task. In our scenario we always guarantee that $n \leq m$, so that every robot will always be assigned a role.

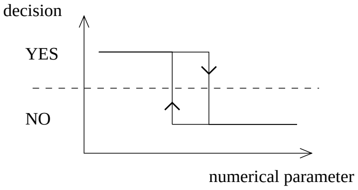

Figure 1: The hysteresis mechanism in decisions

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 robot R_i is assigned the role r_j.

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

1. For each role r_j compute and broadcast $f_j(p)$
2. For each robot R_i ($i \neq p$) and for each role r_j, collect $f_j(i)$
3. $\mathcal{L} = \emptyset$ (Empty the list of assigned robots)
4. For each role r_j do
    - (a) h = the robot R_i ($i \notin \mathcal{L}$) such that $f_j(i)$ has the higher value
    - (b) if $h = p$ then $A(p) = j$ (my role is r_j)
    - (c) $\mathcal{L} = \mathcal{L} \cup \{h\}$

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

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 [5]).

We have chosen a method of stabilizing decisions by means of *hysteresis* (see Fig. 1), 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 towards 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 system described 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 Experimentation

A successful coordination of the team depends on the correct implementation of the coordination protocol and on a suitable calibration of a certain number of numerical parameters, such as the hysteresis and the "utility functions" parameters. Calibration of these parameters requires ad hoc experimentation methods and tools that will be discussed in this section.

In addition, the heterogeneity of the ART robots makes the experimental phase 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 of different capabilities and a set of behaviors with peculiar speed characteristics and, that notwithstanding, robots must calculate comparable numerical values.

4.1 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 useful for verifying both the correctness of the protocol and the intended behaviour of the robots in each of the roles in different situations.

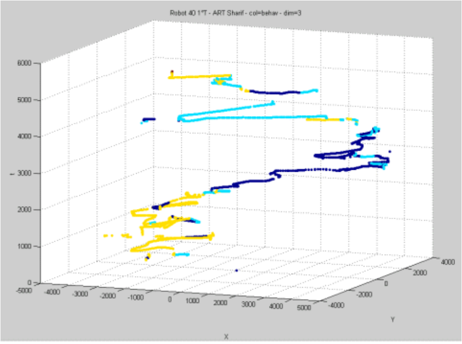

Figure 2: Graphical display of log files

First experiments involving the robots may be done without playing, with steady robots and moving the ball. At this stage one can adjust the discrepancies arising from differences in robots' implementations. The other experimental phase involving 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 *GoToBall*) and adjust the parameters to restore the expected behaviour.

Finally, an analysis of the log files generated during the games is useful for highlighting possible situations in which coordination has not given good results. With this respect we made use of a 3D viewer for experimental data that allows for displaying several information about one or more robots. In Fig. 2 we monitor the position of one robot (X and Y axes) over time (Z axis) with colors denoting the role assigned to the robot. By considering the data for the three robots and using this tool for 3D navigation of this representation, we are able to detect several interesting features of coordination as well as unexpected behaviors of the robots.

4.2 Results

The performance of the ART team at RoboCup 1999 [4] and at European RoboCup 2000 have provided substantial evidence that basic coordination among the team players has been successfully achieved. In fact, in several situations the players smoothly switched role and managed to get ball possession without obstructing each other and they have generally occupied the field in a satisfactory way.

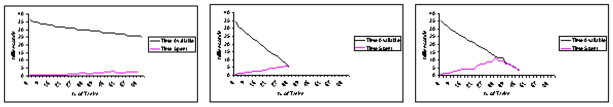

Figure 3: Network Communication in ETHNOS. From left to right: the monitor, Relé, Homer

The evaluation of the coordination protocol 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 analysis both for the communication and for the coordination layer.

Communication analysis. The reliability of communication with the EIEP hes 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 clearly the value depends on the computational load of the tasks in execution as well as the processor speed.

In Fig. 3 the graphs represent different machines (with different processing power) corresponding to two robots (Relé, and Homer) and a monitor, connected using Wavelans(c). The top line in the graph indicates the calculated time available each 100 ms for communication purposes. Clearly this value decreases as the number of tasks in execution increase (and so the computational load). The bottom line indicates the time spent in communication which also increases with the number of experts (this is because for 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, we noticed that when communications were not reliable, the lack of coordination negatively impacts on the overall performance of the team, but it never happened that robots were stuck without knowing what to do, due the coordination protocol. For instance, in case two robots do not communicate each other they will both assume the most important role for the current formation (i.e. they both will try to go to the ball or to tackle the opponent robot).

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

Match	GoToBall	Support	Defend
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%

Table 1: Robots' roles.

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%

Table 2: Robot position in the field.

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 *GoToBall* and *Defend* 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 in 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 occupied, even if there is not an explicit subdivision of the field in competence areas 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.

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. Suppose for example that a problem in the vision system causes the robot to see the ball very close to it, while it is not. This robot will be more likely assigned to have the role corresponding to go to the ball, but it usually will take more time than what previously specified to accomplish its task. In these 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.

5 Conclusions

The distributed coordination method presented in this paper has been successfully employed by all the members of the ART team during the 1999 and 2000 RoboCup competitions [4]. The effectiveness of the method has been proved by the fact that we were always ready to substitute any robot with another one, without affecting the performance of the overall team.

The goal of coordinating through a distributed protocol a multi agent system, formed by heterogeneous components, not only has been achieved, but actually provided a substantial contribution to the overall performance of the ART team. While the effectiveness of coordination has been addressed in the RoboCup environment, the techniques and the tools can be successfully exploited in other multi-robot domains, where similar assumptions are verified.

A key step that made coordination successful has been the experimental work carried out in order to attain the desired coordinated behaviour and the use of suitable tools for the analysis of data exchanged during the game.

From a technical perspective the proposed protocol is based on the explicit exchange of data about the status of the environment and is based on simple forms of negotiations. Simplicity in the protocol stems from the need to make rather weak assumptions on each robot's capabilities. An increase of such capabilities would lead to more complex protocols. However, we believe that a major issue in coordination is to find a suitable balance between the robot individual capabilities and the form of cooperation realized.

Acknowledgments We are grateful to the members of the ART team and acknowledge the support of "Consorzio Padova Ricerche", "Consiglio Nazionale delle Ricerche", Politecnico di Milano, Universities of Roma "La Sapienza", Padova, Genova, Palermo, Parma, AI*IA, Vesta pneumatics, Sony Italia, Images, Tekno.

References

[1] R. Alami, F. Ingrad, and S. Qutub. A scheme for coordinating multi-robot planning activities and plan execution. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence (ECAI-98)*, 1998.

[2] R. C. Arkin and T. Balch. Cooperative multiagent robotic systems. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case Studies of Successful Robot Systems*. MIT Press, 1998.

[3] T. Balch and R. C. Arkin. Motor schema-based formation control for multiagent robot teams. In *ICMAS-1995*, 1995.

[4] S. Coradeschi, L. Karlsson, P. Stone, T. Balch, G. Kraetzschmar, and M. Asada. Overview of RoboCup-99. *A.I. Magazine*. To appear.

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

[6] S. Kraus. Negotiation and cooperation in multiagent environments. *Artificial Intelligence*, 1997.

[7] F. Montesello, A. D'Angelo, C. Ferrari, and E. Pagello. Implicit coordination in a multiagent system using a behavior-based approach. In *DARS-98*, 1998.

[8] D. Nardi, G. Adorni, A. Bonarini, A. Chella, G. Clemente, E. Pagello, and M. Piaggio. ART-99: Azzurra Robot Team. In *Proceedings of 3rd RoboCup Workshop*. Springer-Verlag.

[9] M. Piaggio, A. Sgorbissa, and R. Zaccaria. A programming environment for real time control of distributed multiple robotic systems. *Advanced Robotic Journal*, 14(1), 2000.

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

[11] M. Veloso and P. Stone. Individual and collaborative behaviors in a team of homogeneous robotic soccer agents. In *Proc. of the 3rd International Conference on Multi-Agent Systems*, 1998.

[12] B. B. Werger. Cooperation without deliberation: A minimal behavior-based approach to multirobot teams. *Artificial Intelligence 110*, 1999.

[13] K. Yokota, K. Ozaki, N. Watanabe, A. Matsumoto, D. Koyama, T. Ishikawa, K. Kawabata, H. Kaetsu, and H. Asama. Uttori united: Cooperative team play based on communication. In *Robocup 98 proceedings*, 1998.