Communication and Coordination among heterogeneous Mid-size players: ART99

Claudio Castelpietra[1], Luca Iocchi[1], Daniele Nardi[1], Maurizio Piaggio[2], Alessandro Scalzo[2] and Antonio Sgorbissa[2]

[1] Dipartimento di Informatica e Sistemistica Università "La Sapienza", Roma, Italy
[2] Dipartimento di Informatica, Sistemistica e Telematica Università di Genova, Italy

Abstract. Distributed coordination among *robotic soccer agents* has been considered in the recent years within the framework offered by the RoboCup competitions, mostly in the simulation and F-180 leagues.
In this paper we describe the methods and the results achieved in coordinating the players of the ART team participating in the 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

Distributed coordination of *robotic agents* [7] has been considered as one of the central research issues in the RoboCup competitions [2, 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.

The coordination problem in the context of RoboCup was first faced in the simulation league [6, 8], where it 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 [12]. In the F-2000 (middle size) league, although the number of players is 4 (including the goal keeper), coordination among the players is still a critical issue because the dynamics of the game make it necessary to avoid interferences among players' actions. However, the distinguishing feature of the F-2000 league is the difficulty of reconstructing global information about the environment and thus coordination needs to be achieved without laying down drastic prerequisites on the knowledge of the single players.

The implementation of cooperative strategies for a team of robots can be addressed by relying on explicit communication [13] or by exploiting implicit

P. Stone, T. Balch, and G. Kraetzschmar (Eds.): RoboCup 2000, LNAI 2019, pp. 86-95, 2001.

communication and emergent behaviors [5]. Our choice has been to rely on explicit communication for implementing cooperative strategies; however, due to the frequent communication failures the robots must depend neither on communication, nor on information provided by other robots.

The Azzurra Robot Team (ART) [9] of robotic soccer players, participating in the RoboCup competitions in the F-2000 category, is composed of different players (both in the hardware and in the software) developed in various Italian universities. Because of this kind of organization, 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. Moreover, results on the coordination between players and the goal-keeper are described in [1].

Summarizing the hypotheses underlying the coordination problem 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 strategies (defensive, offensive). While the strategic level is currently demanded to an external selection (the human coach), roles are dynamically assigned (see [12]) to the various team elements during the game. In the following sections we describe the communication infrastructure and the coordination protocol, we 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 [10] 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. In the EIEP the robots are allowed to subscribe to communication clubs. For example, we may envisage a single club to which the different players belong or different clubs, to which only players which are performing a cooperative activity belong. Messages are exchanged with a publish/subscribe technique in which subscription and publication can thus be distinguished in internal, external in a specific club or external in all clubs. Whenever a message is published ETHNOS transparently and dynamically distributes the messages to the appropriate subscribed receivers. Figure 1 shows an example configuration that we have tested. In particular 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) which monitors the activity of all the players for displaying and debugging the team activity during a match.

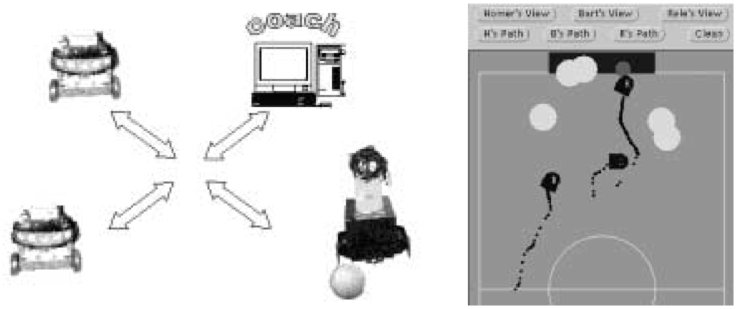

Fig. 1. Left: example of multi-robot ETHNOS configuration, Right: monitoring the team activity.

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: ETHNOS automatically updates the database of each club's members and consequently dynamically dispatches messages from senders to the appropriate receivers. 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 (i.e. radio link, Wavelan(c), 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. For this reason we have also designed a protocol for this type of applications, called EEUDP (Ethnos Extended UDP) because, based on the UDP, it extends it with the necessary properties. The EEUDP 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 [12, 11]. 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 needed to avoid that more than one robot of the same team tries to perform the same action (e.g. to go to the ball) and to occupy properly various parts of the field.

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.

In [13] a coordination method used in the F-2000 league is presented: the basic idea is that all the robots try to get the ball and the first one who reaches it asks the other robots to stay away from the ball. We believe that the dynamic roles' assignment is equally effective in gaining ball possession while it allows the robots to better occupy the various parts of the field.

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 set of pre-computed formations that can be used depending on the current state of the environment The automatic generation of these formations from a declarative specification of the robots' abilities is described in detail in [3]. The robots also possess the rules to determine which formation should be adopted on the basis of the environment configuration. But, considering that each robot's data do not necessarily coincide with those of the others, the robots may determine different formations. Therefore each robot adopts the formation that gets the majority of votes among those proposed by the team.

In order to avoid the risk of oscillations in the common decision relevant to the formation due to varying numerical parameters or to the message transmission delays, each robot has to moderate the frequency of changes in proposed formation. The main way to do it is to use some kind a decision-stabilization method as discussed below.

Step 2: Roles selection This step implements dynamic role assignment through explicit communication of the "utility values". Such values indicate the usefulness, with regard to the whole team, of each role to be assigned to each robot. 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 the RoboCup scenario we always guarantee that $n \leq m$, so that every robot will always be assigned a role.

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 $f_j(p)$ and broadcast the computed value
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.

3.2 Critical factors for effective implementation

In order to obtain an effective application of the above method a number of issues need to be dealt with properly: the stability of decisions, the recognition of situations where a robot cannot successfully accomplish the task associated with the assigned role and communication failures.

It is important for all the high level decisions taken by the robot, including those regarding coordination, to be stable with respect to possible oscillations of the numerical parameters upon which they depend (see [6] par.5.2).

We have chosen a method of stabilizing decisions by means 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

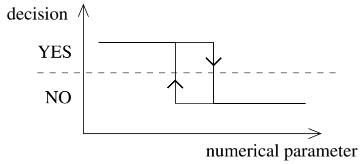

Fig. 2. The hysteresis mechanism in decisions.

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 important roles 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 that are discussed in the next section).

3.3 Example

As an example, we will analyze the numerical values of the utility function used to determine which player must move towards the ball. For convenience, the output is expressed as an estimated time, in seconds (consider that in this case the role is assigned to the one robot that obtains the lowest value):

- Base value: distance in millimeters from the ball, divided by the average robot speed, in mm/s.
- If the ball cannot be directly seen by the robot: add 1 second.
- If there are obstacles between the robot and the ball: add 1.4 seconds.
- If the ball must be passed to shoot in goal: add 180 degrees divided by the average rotational speed in deg/s.
- If the robot is stuck and cannot move: add 10 seconds.
- *Hysteresis:* add 1.8 seconds if at the moment one does not have the role for moving toward the ball.

4 Experimentation

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. As an 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.

A successful coordination of the team depends on a correct implementation of the coordination protocol and on a suitable calibration of a number of numerical parameters, such as the hysteresis and the utility functions parameters.

The experimentation of the coordination protocol must be done in stages which require the use of different tools. In particular, in order to realize an effective implementation of this approach and with the aim of evaluating this method, we have relied upon the use of different tools: a simulator, data analysis without play and during actual games, analysis of log files after the games.

The first and easier experimental setting is provided by a simulator. While a simulator, in general, cannot provide a precise characterization of all the aspects that influence the performance of the robot in the real environment, it can provide useful feedback to the design of the coordination system for actual robots. Through a simulator one can verify both the correctness of the protocol and the intended behaviour of the robots in each of the roles in different scenarios.

First experiments involving the robots may be done without play, with steady robots and moving the ball. At this stage one can adjust the discrepancies arising from differences in robots' implementations. In particular, one can compare transmitted values, due to different implementations of the utility functions, and differences can thus be spotted (together with major failures in the sensing capabilities of each robot).

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 take the role of moving towards the ball and adjust 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.

4.1 Evaluation

The performance of the ART team at RoboCup 1999 [4] have provided substantial evidence that basic coordination among the team players has been successfully achieved.

The reliability of communication with the EIEP was 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.

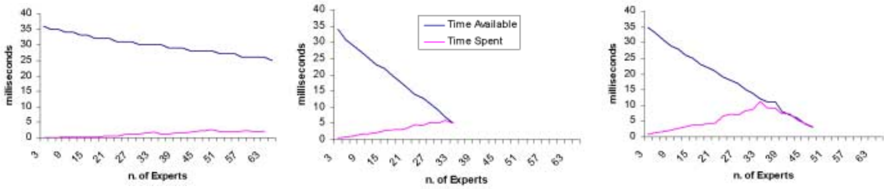

Fig. 3. Network Communication in ETHNOS. From left to right: the monitor, Relé, Homer (upper line = time available ; lower line = time spent).

In figure 3 the three 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. For example, if we consider Relé, the limit situations in which the two lines converge is also the limit beyond which the schedulability analysis fails. Instead, if we consider Homer, real time scheduling is possible also beyond the intersecting point but only if we accept communication degradation. Degradation is in particular interesting. In fact 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 with 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).

The coordination protocol has been evaluated by defining some measures that have been thought to be interesting in the RoboCup environment: the coverage of the field during the game, the average number of robots going to the ball at every time, role switching rate, etc. The following analysis has been computed from the data acquired during the games of European RoboCup 2000.

In Table 1a) we show the percentage of time in which at least one of the robot has occupied a zone of the field. 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.

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%

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. a) Robot position in the field. b) Robots' roles.

Table 1b) refers to the roles covered by the players during the games. Also in this case roles have been covered in an effective way, in particular the most important roles *GoToBall* and *Defend* have been executed almost at every time. With respect to a static assignment of roles, dynamic assignment still provides a good distribution of roles, but with the advantage of selecting the most appropriate robot for every role depending on the current situation of the environment.

In addition, the other statistical data have been computed:

- During the game there is in the average one role switch every 7 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.

This analysis shows the effectiveness of our distributed coordination that has significantly contributed to the overall performance of the team during the last competitions. In fact, in general the players smoothly switched role and managed to get ball possession without obstructing each other and they generally have occupied the field in a satisfactory way.

5 Conclusions

The distributed coordination method presented in this paper has been successfully employed by all the members of the ART team during the RoboCup 1999 and European RoboCup 2000 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 coordinate behaviour 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. We have used several measures and tools for tuning and evaluating the effectiveness of the coordination protocol.

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", Università di Roma "La Sapienza", Politecnico di Milano, Università di Padova, Università di Genova, Università di Palermo, Università di Parma, AI*IA, Vesta pneumatics, Sony Italia, Images, Tekno.

References

1. G. Adorni, S. Cagnoni, M. Mordonini, and M. Piaggio. Team/goal-keeper coordination in the robocup mid-size league. In *RoboCup-2000: Robot Soccer World Cup IV*, 2000.
2. M. Asada. The RoboCup physical agent challenge: Goals and protocols for Phase-I. In H. Kitano, editor, *Robocup-97: Robot Soccer World Cup I*, 1998.
3. C. Castelpietra, L. Iocchi, D. Nardi, and R. Rosati. Coordination in multi-agent autonomous cognitive robotic systems. In *Proc. of 2nd International Cognitive Robotics Workshop*, 2000.
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. Ferraresso, C. Ferrari, E. Pagello, R. Polesel, R. Rosati, A. Speranzon, and W. Zanette. Collaborative emergent actions between real soccer robots. In *RoboCup-2000: Robot Soccer World Cup IV*, 2000.
6. M. Hannebauer, J. Wendler, P. Gugenberger, and H. Burkhard. Emergent cooperation in a virtual soccer environment. In *DARS-98*, 1998.
7. Y. Lesperance, H.J. Levesque, F. Lin, D. Marcu, R. Reiter, and R.B. Scherl. A logical approach to high-level robot programing. In *AAAI FAll Symposium on Control of the Physical World by Intelligent Systems*, 1994.
8. F. Montesello, A. D'Angelo, C. Ferrari, and E. Pagello. Implicit coordination in a multi-agent system using a behavior-based approach. In *DARS-98*, 1998.
9. D. Nardi, et al. ART-99: Azzurra Robot Team. In *Proceedings of 3rd RoboCup Workshop*. Springer-Verlag.
10. M. Piaggio, A. Sgorbissa, and R. Zaccaria. A programming environment for real time control of distributed multiple robotic systems. *Advanced Robotic Journal*, 14, 2000.
11. 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, June 1999.
12. M. Veloso and P. Stone. Individual and collaborative behaviors in a team of homogeneous robotic soccer agents. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 309–316, July 1998.
13. K. Yokota et al. Uttori united: Cooperative team play based on communication. In *Proceedings of 2nd Workshop on Robocup*, 1998.