The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Multi-Robot Team Coordination through Roles, Positionings and Coordinated Procedures

Nuno Lau, *Member, IEEE*, Luís Seabra Lopes, *Member, IEEE*, Gustavo Corrente and Nelson Filipe

Abstract — **The coordination methodologies of CAMBADA, a robotic soccer team designed to participate in the RoboCup middle-size league (MSL), are presented. The approach, which relies on information sharing and integration within the team, is based on formations, flexible positionings and dynamic role and positioning assignment. Role/positioning assignment follows a new priority-based algorithm that maintains a competitive formation, covering the most important roles/positionings when malfunctions lead to a reduction of the team size. Coordinated procedures for passing and setplays have also been implemented. With this design, CAMBADA reached the 1st place in the RoboCup'2008 world championship. Competition results and performance measures computed from logs and videos of real competition games are presented and discussed.**

I. Introduction

As robots become increasingly available in different areas of human activity, researchers are naturally prompted to investigate how robots can cooperate with each in order to perform different tasks. Moreover, progress in wireless communication technologies enables information sharing and explicit coordination between robots. These are basic capabilities needed to support sophisticated cooperation and coordination algorithms. Given this increasing availability of robots and communication technologies, multi-robot systems have, in the last two decades, been receiving more and more attention from researchers [2][43][8].

Interest on multi-robot systems is further justified by the advantages they offer with respect to single robots. First, some tasks are simply too difficult or impossible to be carried out by a single robot. In other cases, by providing a larger work force, multi-robot systems can carry out tasks faster. Multi-robot systems also facilitate scalability, as larger problems can often be solved by adding more robots to the team. Finally, through their inherent redundancy, multi-robot systems offer robustness, as they may still work when a team member is damaged or malfunctioning.

These advantages make multi-robot systems useful in a variety of domains, such as exploration of unknown or changing environments [28][45][9] (including such diverse applications as ecological monitoring, rescue, de-mining or planetary exploration), mapping [10], foraging [13], transportation [20], manufacturing [41], intrusion detection and patrolling [17][3], or even entertainment [27].

The development of multi-robot systems raises many new research issues, not found in isolated robots. These new issues are concerned with how the individual robots can coordinate their actions to carry out the assigned tasks as efficiently as possible. Among other issues, the following can be mentioned: How are different sub-tasks assigned to different robots [21][13][29]? How can different roles be assigned to different robots [40][36][33]? If robots need to move in formation, how can the formation be controlled [43][7][11]? How can multi-robot plans be generated and/or executed [23][1]? Which information should robots exchange in order to enable coordination [15][25]? How can multi-robot systems be debugged [19][14]?

The authors have been addressing several of these issues in the robotic soccer domain, currently a popular scenario and application for research in multi-robot systems. In particular, the authors contributed to the development of CAMBADA, a RoboCup middle-size league (MSL) team (Fig. 1). The MSL is one of the most challenging leagues in RoboCup. Robotic players must be completely autonomous and must play in a field of 12m × 18m [30]. Teams are composed of at most six robots with a maximum height of 80 cm. Human interference is allowed only for removing malfunctioning robots and re-entering robots in the game.

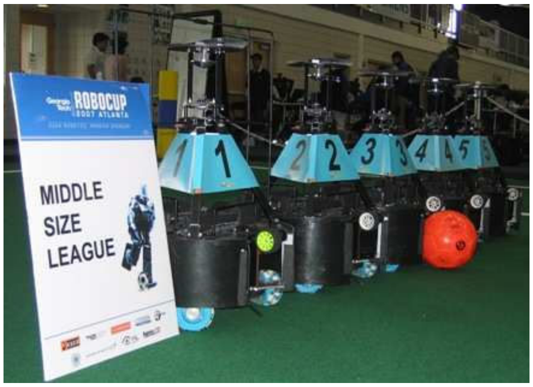

Fig. 1 CAMBADA robotic team

Building a team for the MSL is a very challenging task, both at the hardware and software levels. To be competitive,

Nuno Lau and Luís Seabra Lopes are with the ATRI/IEETA as well as with the Department of Electronics, Telecommunications and Informatics, Universidade de Aveiro, Portugal. (e-mails: { nunolau, lsl }@ua.pt).

Gustavo Corrente was a researcher with ATRI/IEETA, Universidade de Aveiro, Portugal, and is currently with Nokia Siemens Networks Portugal, Aveiro, Portugal. (e-mail: gustavo@ua.pt)

Nelson Filipe was an MSc student on Computer and Telematics Engineering at Universidade de Aveiro, Portugal (e-mail: nelson.filipe@ua.pt)

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

robots must be robust, fast and possess a comprehensive set of sensors. At the software level they must have an efficient set of low-level skills and must coordinate themselves to act as a team. Research conducted within CAMBADA has led to developments concerning hardware [6], computational and communications infrastructure [4][34][35], vision system [31][12], monitoring / debugging [19] and high-level deliberation and coordination [25]. This paper focuses on the last aspect, providing a detailed and up-to-date account of the currently used algorithms and their performance.

The complexity inherent to the MSL and, in particular, the difficulty of developing robots with robust sensorimotor capabilities and informative perception capabilities explains why most teams have implemented relatively simple coordination capabilities. The more advanced teams achieve coordination through the assignment of different roles to the robots [44][5][33]. Typically there is, at least, an attacker, a defender, a supporter and a goalie. As perception and sensorimotor capabilities become more sophisticated it will be possible to develop more sophisticated coordination algorithms. This trend is pushed further by the increase in team size (number of robots) as well as field size. A natural source of inspiration is the RoboCup Soccer Simulation League, where teams have been using coordination layers with strategy, tactics and formations [37][40], coordination graphs [24] and reinforcement learning [38]. As will be detailed in this paper, some of the coordination algorithms used in CAMBADA are adaptations of algorithms initially proposed and tested in the simulation league.

CAMBADA participated in several national and international competitions, including RoboCup world championships (5th place in 2007, 1st place in 2008) and the annual Portuguese Open Robotics Festival (3rd place in 2006, 1st place in 2007 and 2008). The excellent result obtained in RoboCup'2008 is largely due to the developed coordination methodologies, as the CAMBADA robots are among the slowest of the competition.

This paper is organized as follows: Section II presents the hardware and software architectures of CAMBADA players and provides details on the main software components involved in individual decisions of the players. Section III describes how players share information with teammates and how they integrate shared information. Sections IV and V describe the adopted coordination methodologies. Section VI presents and discusses competition results and various performance measures. Section VII concludes the paper.

II. Player Architecture

CAMBADA robots (Fig. 1) were designed and completely built at the University of Aveiro. Each robot fits into a cylindrical envelope with 485 mm in diameter. The mechanical structure of the players is layered and modular. Each layer can easily be replaced. The components in the lower layer, namely motors, wheels, batteries and an electromechanical kicker, are attached to an aluminum plate placed 8 cm above the floor. The second layer contains the control electronics. The third layer contains a laptop computer, at 22.5 cm from the floor, a catadioptric omnidirectional vision system, a frontal vision system (single camera) and an electronic compass, all close to the maximum height of 80cm.

The players are capable of holonomic motion, based on three omni-directional roller wheels. With the current motion system, the robots can move at a maximum speed of 2.0 m/s. As mentioned, this is less than many of the other MSL teams, which can currently move at speeds typically between 2.5 and 4.0 m/s (e.g. [32] [22] [39] [16]). The mentioned vision system allows detecting objects, the ball, players, and field lines on a radius of 5m around each player. The frontal camera allows detecting the ball further away. Each player also carries encoders, battery status sensors and, for detecting if the ball is kickable, an infra-red presence sensor.

The computational system in each robot is a set of processing nodes (several small microcontrollers for basic perception and sensorimotor control plus a laptop for high-level deliberation) connected through a Controller Area Network (CAN). All communications within the team are based on the standard wireless LAN protocol IEEE 802.11x profiting from large availability of complying equipment. The team receives referee's instructions through a wired LAN TCP link.

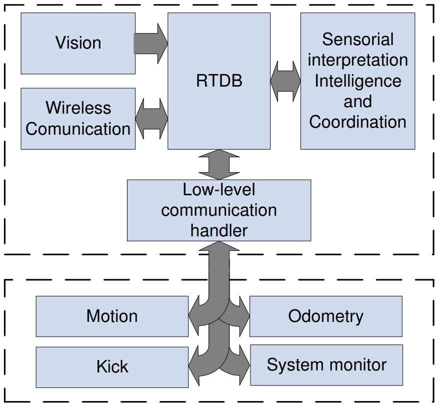
Vision
RTDB
Sensorial interpretation Intelligence and Coordination
Wireless Comunication
Low-level communication handler
Motion
Odometry
Kick
System monitor

Fig. 2. Layered software architecture of CAMBADA players, from [6]

On the main processing node (laptop), CAMBADA players run several software processes that execute different activities, such as image acquisition, image analysis, integration/deliberation and communication with the low-level modules (Fig. 2). The order and schedule of activation of these processes is performed by a so-called process manager (Pman [35]). Pman stores the characteristics of each process to activate and allows the activation of recurrent tasks, settling phase control (through the definition of temporal offsets), precedence restrictions, priorities, etc. The Pman services allow changes in the temporal characteristics of the process schedule during run-time.

The top-level processing loop starts by integrating perception information gathered locally by the player. This includes information coming from the vision processes, odometry information coming from the holonomic base, compass information and ball presence information. All this information is stored in a shared data structure called Real-Time Data Base (RTDB). The RTDB has a local area, shared only among local processes, and a global area, where players share their world models to the other players. The global area is transparently updated and replicated in all players in real-time. Self-localization uses a sensor fusion engine based on the publicly available engine described in [26]. By integrating information from field line detection, this engine produces self position estimates with a high level of confidence. Compass information is used to resolve ambiguities and detect self-localization errors. The final fusion step is to integrate local information with information shared by teammates. After this integration, part of the world state is written to the global area of the RTDB.

Deliberation in CAMBADA considerably relies on the concepts of *role* and *behavior*. Behaviors are the basic sensorimotor skills of the robot, like moving to a specific position or kicking the ball. The set of behaviors that are implemented in the CAMBADA agent are adapted to its catadioptric omnidirectional vision and holonomic driving systems. The combination of these technologies enhances the set of possible behaviors when compared to a differential drive robot or to an holonomic drive robot with a limited field of view. In brief, the current set of behaviors is the following:

- ***bMove*** uses two symbolic parameters: the target position where to move; and the position which the CAMBADA player should be facing in its path to the target. The symbols used are *OBall*, *TheirGoal* and *OurGoal*. This behavior may activate the functions of avoiding obstacles and avoiding the ball (used during the game repositions to avoid collisions with the ball).
- ***bMoveToAbs*** is another moving behavior; it allows the movement of the player to an absolute position in the game field, and also allows the player to face any given position. Obstacle avoidance is also included.
- ***bPassiveInter*** moves the player to the closest point in the ball trajectory and waits there for the ball.
- ***bDribble*** is used to dribble the ball to a given relative player direction.
- ***bCatchBall*** is used to receive a pass. The player aligns itself with the ball path and, when the ball is close, moves backwards to soften the impact and more easily engage the ball.
- ***bKick*** is used to kick the ball accurately to one 3D position, either for shooting to goal or passing to a teammate. Preparing for the kick involves determining the kick direction and power. Polynomial functions, whose coefficients were determined by experimentation, are used to compute kick power based on distance to target. Different functions are used according to the expected number of ball bounces, given the distance.
- ***bGoalieDefend*** is the main behavior of the goalie.

Roles select the active behavior at each time step. During open play, the CAMBADA agents use only three roles: *RoleGoalie*, *RoleSupporter* and *RoleStriker*. The *RoleGoalie* is activated for the goalkeeper. Further details about the developed roles and respective coordination mechanisms will be presented in sections IV and V.

III. Information Sharing and Integration

Sharing perceptional information in a team can improve the accuracy of world models and, indirectly, the team coordination [15]. Therefore, information sharing and integration is one of the key aspects in multi-robot teams.

In CAMBADA, each robot uses the information shared by the other robots, obtained through the RTDB, to improve its knowledge about the current positions and velocities of the other robots and of the ball. It is very important for our coordination model to keep an accurate estimate of the absolute position of the ball by each robot. The role assignment algorithm is based on the absolute positions of the robot and of its teammates. The teammates' positions are not obtained through the vision system and rely completely on the communicated estimated self positions of others.

Each agent communicates its own absolute position and velocity to all teammates as well as its ball information (position, velocity, visibility and engagement in robot), current role and current behavior.

Multi-robot ball position integration has been used in the middle-size league by several teams [44][18]. In CAMBADA, multi-robot ball position integration is used to maintain an updated estimate of the ball position, when the vision subsystem cannot detect the ball, and to validate robot's own ball position estimate, when the vision subsystem detects a ball.

Currently, a simple integration algorithm is used. When the agent doesn't see the ball, it analyzes the ball information of playing teammates. The analysis consists in the calculation of the mean and standard deviation of the ball positions, then discarding the values considered as outliers of ball position, and finally using the ball information of the teammate that has a shorter distance to the ball. To determine if the agent sees a fake ball, i.e., to validate the robot's own perception, we use a similar algorithm.

Communication is also used to convey the coordination status of each robot allowing robots to detect uncoordinated behavior (e.g., several robots with the same exclusive role) and to correct this situation reinforcing the reliability of coordination algorithms.

The communication between the base station and the robots informs the robots of the active play mode (decided by the referee). During development, the base station can be used to control several robotic agent characteristics like fixed roles, manually activated self-positioning, etc, all managed through the RTDB.

IV. Positionings and Roles in Open Play

For open play, CAMBADA uses an implicit coordination model based on notions like *strategic positioning*, *role* and *formation*. These notions and related algorithms have been introduced and/or extensively explored in the RoboCup Soccer Simulation League [40][36]. In order to apply such algorithms in the MSL, several changes had to be introduced. The approach is presented in detail in this section.

A. Formations and strategic positionings

A formation defines a movement model for the robotic players. Formations are sets of strategic positionings, where each *positioning* is a movement model for a specific player. The assignment of players to specific positionings is dynamic, and is done according to some rules described below. Each positioning is specified by three elements:

- Home position, which is the target position of the player when the ball is at the centre of the field
- Region of the field where the player can move, and
- Ball attraction parameters, used to compute the *target position* of the player in each moment based on the current ball position

All these items of information are given in a strategy configuration file. Using different home positions and attraction parameters for the positionings allows a simple definition of defensive, wing, midfielder and attack strategic movement models. Fig. 3 shows the formation of the team used in RoboCup'2008 for several ball positions.

The definition of formation in terms of strategic positionings was introduced in the SBSP model [36] for the Soccer Simulation League. This model also introduced specific notions of *tactic* and *strategy*, which are currently not used in CAMBADA.

B. Roles in open play

As mentioned before, the CAMBADA players use only three roles in play-on mode: *RoleGoalie*, activated for the goalkeeper, *RoleSupporter* and *RoleStriker*. *RoleStriker* is an "active player" role. It tries to catch the ball and score goals. The striker activates several behaviors that try to engage the ball (*bMove, bMoveToAbs*), get into the opponent's side avoiding obstacles (*bDribble*) and shoot to the goal (*bKick*). The *bKick* behavior can perform 180º turns while keeping possession of the ball.

In a consistent role assignment, only one player at a time takes on the role of striker. The striker is helped by other teammates which take on *RoleSupporter* [25]. Supporters maintain their target positions as determined by their current positioning assignments and the current ball position. As a result, supporters accompany the striker as it plays along the field, without interfering. In case the ball is captured by the opponent, some supporter hopefully will be in a good position to become the new striker. Occasionally, supporters can take a more active behavior. This happens when the striker can't progress with the ball towards the opponent goal and, instead, the ball remains behind the striker for more than some pre-defined time (e.g. 2 seconds in the adopted configuration). In this case, the closest supporter to the ball also approaches the ball, acting as "backup striker".

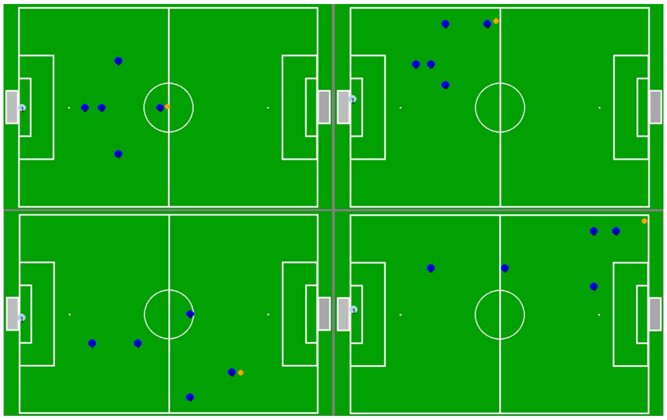

Fig. 3. Target player positions for several different ball positions

```
Algorithm: role and positioning assignment
Input:
  POS – array of N positionings
  BallPos – ball position
Input/output:
  PL – array of K active players (K =< N)
Local:
  TP – array of N target positions
{
  clearAssigments(PL);
  TP = calcTargetPositions(POS,BallPos);
  for each POS[i], i ∈ 1..N, in
    descending order of priority
  {
    if there is no free player
      then return;
    p = the free player closest to TP[i];
    PL[p].positioning = i;
    PL[p].targetPosition = TP[i];
    if POS[i] has highest priority
      then PL[p].role = striker;
      else PL[p].role = supporter;
  }
}
```

Fig. 4. CAMBADA Positioning and role assignment algorithm

C. Role and positioning assignment

Previous work on role assignment algorithms for robotic soccer is based on the concept of role exchange, measuring the utility of that exchange to decide its activation [37][40]. However, in MSL the number of available players varies as a result of several common situations, namely hardware and software malfunctions and referee orders. As the number of robots is small and varies a lot, the usefulness of role exchanges is reduced. The algorithm used in CAMBADA for role and positioning assignment is based on considering different priorities for the different roles and positionings, so that the most important ones are always covered.

The algorithm is presented in Fig. 4. Consider a formation with N positionings and a team of $K \leq$ N available field players (not counting the goal-keeper which has a fixed role). To assign the role and positioning to each robot, the distances of each of the robots to each of the target positions are calculated.

Then the striker role is assigned to the robot that is closest to the highest priority strategic positioning, which is in turn the closest to the ball. From the remaining *K*-1 robots, the closest to the defensive positioning (second highest priority) is assigned to this positioning, then the closest to the third level priority positioning is assigned next and the algorithm continues until all active robots have positionings and roles assigned. This algorithm results in the striker role having top priority, followed by the defensive positioning, followed by the other supporter positionings. The assignment algorithm may be performed by the coach agent in the base station, ensuring a coordinated assignment result, or locally by each robot, in which case the inconsistencies of world models may lead to unsynchronized assignments. In the latest competitions, positioning assignments were carried out by the coach at intervals of 1 second and the role assignments were individually carried out by each player.

V. Coordinated Procedures

Coordinated procedures are short plans executed by at least two robots. These plans in some cases involve communication resulting in explicit coordination. In the case of CAMBADA coordinated procedures are used for passes and set plays.

A. Passes

Passing is a coordinated behavior involving two players, in which one kicks the ball towards the other, so that the other can continue with the ball. Until now, MSL teams have shown limited success in implementing and demonstrating passes. In RoboCup 2004, some teams had already implemented passes, but the functionality was not robust enough to actually be useful in games [27] [42]. The CoPS team also support pass play [46].

Two player roles have recently been developed for coordinated passes in the CAMBADA team. In the general case, the player running *RoleStriker* may decide to take on *RolePasser*, choosing the player to receive the ball. After being notified, the second player takes on the *RoleReceiver*.

These roles have not been used yet for open play in international competition games, but they have been demonstrated in RoboCup'2008 MSL *Free Technical Challenge* and a similar mechanism has been used for corner kicks (see below). In the free challenge, two robots alternately took on the roles of passer and receiver until one of them was in a position to score a goal.

The sequence of actions on both players is described in Table I. They start from their own side of the field and, after each pass, the passer moves forward in the field, then becoming the receiver of the next pass. The coordination between passer and receiver is based on passing flags, one for each player, which can take the following values: READY, TRYING_TO_PASS and BALL_PASSED. In the case of a normal game, another pass coordination variable would identify the receiver.

Table I – Coordinated actions in a pass

RolePasser	**RoleReceiver**
PassFlag ← TRYING_TO_PASS	
Align to receiver	Align to Passer
	PassFlag ← READY
Kick the ball	
PassFlag ← BALL_PASSED	
Move to next position	Catch ball

B. Set plays

Another methodology implemented in CAMBADA is the use of coordinated procedures for set plays, i.e. situations when the ball is introduced in open play after a stoppage, such as kick-off, throw-in, corner kick, free kick and goal kick. Set play procedures define a sequence of behaviors for several robots in a coordinated way. For that purpose, the involved players take on specific roles. This role-based implementation of set plays not only was easy to integrate within the previous agent architecture, but also facilitated the test and tune of different possibilities allowing for very efficient final implementations.

RoleToucher and *RoleReplacer* are used to overcome the MSL indirect rule in the case of indirect set pieces against the opponent. The purpose of *RoleToucher* is to touch the ball and leave it to the *RoleReplacer* player. The replacer handles the ball only after it has been touched by the toucher. This scheme allows the replacer to score a direct goal if the opportunity arises.

Two toucher-replacer procedures are implemented. In the case of *corner kicks*, the toucher passes the ball to the replacer and the replacer continues with the ball (see pseudo-code in Fig. 5). The passing algorithm is as explained above.

Another toucher-replacer procedure is used in the case of *throw-in*, *goal kick* and *free kick* set plays. Here, the toucher approaches and touches the ball pushing it towards the replacer until the ball is engaged by the replacer, then withdraws leaving the ball to the replacer. The replacer also moves towards the ball, grabs it, waits that the toucher moves away and then shoots to the opponent goal. It should be noted that both the toucher and the replacer position themselves on the shoot line, so that, as soon as the toucher moves away, the replacer is ready to shoot. For the kick-off, a similar procedure is followed, but without reference to the shoot line, since the involved robots must be in their own side of the field.

```
Algorithm: RoleReplacer // for corner kicks
{
  if I have Ball then shoot to opponent goal
    else if Ball close to me
      then move to Ball
      else if Toucher already passed ball
             then catch Ball
             else wait that Ball is passed
}
```

Fig. 5. Replacer role algorithm for corner kicks

Finally, in the case of set pieces against CAMBADA, *RoleBarrier* is used to protect the goal from a direct shoot.

The line connecting the ball to the own goal defines the barrier positions. One player places itself on this line, as close to the ball as it is allowed. Two players place themselves near the penalty area. One player is placed near the ball, 45º degrees from the mentioned line, so that it can observe the ball coming into play and report that to teammates. Finally, one player positions itself in such a way that it can oppose to the progression of the ball through the closest side of the field. The placement of players is illustrated in Fig. 6.

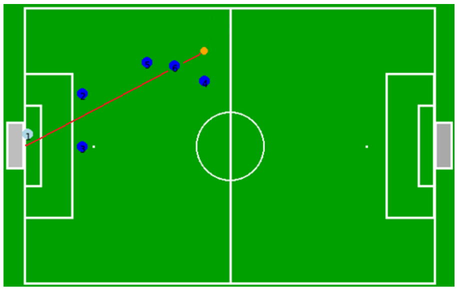

Fig. 6. Placement of RoleBarrier players

The assignment of the *RoleBarrier*, *RoleReplacer* and *RoleToucher* roles is executed by sorting the agents according to their perceived distances to the ball and selecting the closest ones, up to the maximum number of agents in each role. When selecting a role like the *RoleReplacer*, which is exclusive, the agent looks at the other teammates role decisions and if it finds a *RoleReplacer* with a lower uniform number it will never select that role. A similar approach is performed for the other exclusive roles. This assignment is always performed locally by each robot.

VI. Performance Evaluation

The CAMBADA team participated and won the MSL world championship in RoboCup'2008 (Suzhou, China, July 2008). Part of the performance evaluation results presented in this section are obtained by analyzing log files and videos of games in this championship. In addition, RoboCup'2008 competition results will also be presented.

As the CAMBADA team made it to the final, it was scheduled to play 13 games. One of them was not played due to absence of the opponent. For two other games, the log files were lost. The results presented in the following are therefore extracted from log files of the remaining 10 games.

Table II shows the average percentage of time any given player spends in each role, with respect to the total time the player is active in each game. It can be seen that players spend a considerable amount of time in set plays (44% of the total time of the player in a game, including the *RoleParking*, which moves the player to a position outside the field at the end of the first half and at the end of the game). This reflects the current contingencies of MSL games. More time is spent in set plays against CAMBADA (28.4%, since usually four players take the barrier role in these situations) than in set plays against the opponent (11.5% in toucher and replacer roles). According to the logs, players change roles 2.017 ± 1.022 times per minute. As role assignment is distributed (implicit coordination), it occasionally happens that two players take on the role of striker at the same time. On average, all inconsistencies in the assignment of the striker role have a combined total duration of 20.9 ± 27.4 seconds in a game (~30 minutes). The high standard deviation results mainly from one game in which, due to magnetic interference, localization errors were higher than normal. In that specific game, role inconsistency occurred 45 times for a total of 101 seconds.

Table II – Average time spent by players in different roles (in %) and respective standard deviation

Role	%time
Striker	10.4 ± 5.2
Supporter	45.2 ± 10.0
Toucher	5.9 ± 4.1
Replacer	5.6 ± 4.6
Barrier	28.4 ± 6.5
Parking	4.4 ± 6.4

Table III shows the average percentage of time any given player spends running each implemented behavior. In particular, the second column of the table shows such percentages irrespective of the role taken. The third column shows the percentages of time in each behaviour, considering only the periods in which players are taking the striker role. These values highlight clearly the specific features of the striker: much less time moving to absolute positions, since the striker most of the time ignores its strategic positioning assignment; much more time in moving (to the ball), dribbling and kicking.

Table III – Average time spent by players running different behaviors (in %) and respective standard deviation

Behavior	%time (any role)	%time (striker)
bMove	4.9 ± 3.0	43.7 ± 4.4
bMoveToAbs	74.7 ± 12.6	25.3 ± 4.7
bDribble	1.4 ± 1.2	13.4 ± 4.5
bKick	1.8 ± 1.5	14.6 ± 7.7
bCatchBall	0.2 ± 0.3	
bPassiveInter	0.3 ± 0.2	2.9 ± 1.1
bStopRobot	14.7 ± 7.0	

Concerning strategic positionings, relevant mainly to supporters as explained above, the average distance of the player to its target position (given by the assigned strategic positioning and the ball position) is 1.381 ± 0.477 m. The strategic positioning assignment for each player is changed on average 9.829 ± 2.228 times per minute.

As the CAMBADA players to not track the positions and actions of the opponent players, it is not possible to compute an exact measure of ball possession. However, the game logs enable to compute related measures, as shown in Table IV. The closest player to the ball is at an average distance of

1.2m from the ball (the field is 18m × 12m). The ball is perceived by at least one robot 91.7% of the time. The ball is engaged in a robot's grabber device 9.8% of the time.

Table IV – Measures related to ball possession

Average minimum distance to the ball (meters)	1.25 ± 0.33
Time with ball visible (%)	91.7 ± 3.5
Time with ball engaged (%)	9.8 ± 4.7

Table V – Competition results

	#games	#goals scored	#goals suffered	#points
Round-robin 1	5	41	2	15
Round-robin 2	4	16	3	9
Round-robin 3	2	5	2	3
Semi-final	1	4	3	3
Final	1	7	1	3
Total	13	73	11	33

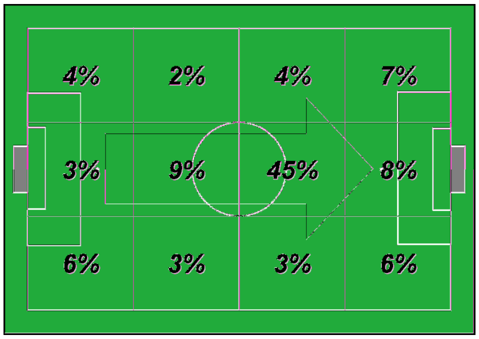

Fig. 7. Percentage of time the ball was in different locations of the field in 10 games (CAMBADA on the left)

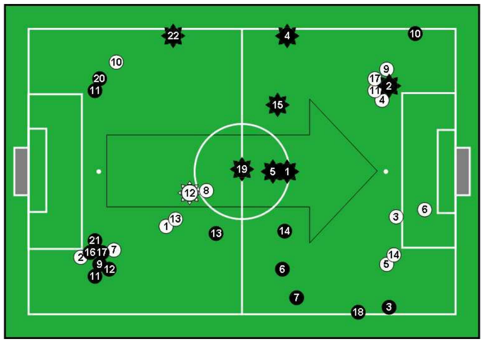

Fig. 8. Shoot locations in the final CAMBADA (black, on the left) - TechUnited (white, on the right) game in RoboCup 2008 (shoots are circles and goals are sun-like forms)

Fig. 7 shows the percentage of time the ball was in different regions of the field in the 10 games played by CAMBADA for which we have logs. We see that the ball was in the opponent's side for 73% of time, and that the game was mainly being played in centre of the field, towards the opponent's side. Fig. 8 shows the location in the field from where the ball was shot to goal in the RoboCup'2008 MSL final (CAMBADA-TechUnited).

Table V presents the competition results of CAMBADA in RoboCup'2008. The team won 11 out of 13 games, scoring a total of 73 goals and suffering only 11 goals. This result is even more notable if compared with the second best team which scored 38 goals and suffered 13.

VII. Conclusion

The paper presented and evaluated the coordination methodologies of the CAMBADA team, the 2008 world champion in RoboCup MSL.

During open play, an implicit coordination approach, based on formations, flexible positionings and dynamic role and positioning assignment, is used. The positioning of the team adapts to the external game conditions and maintains a strong defense and a good backup to the striker role. This is achieved through a priority-based positioning/role assignment algorithm that maintains a competitive formation even when robot malfunctions decrease the number of field players. This algorithm is focused on covering the most important roles/positionings and differs substantially from previously presented algorithms that were based on role exchange. The success of the approach can be seen, not only from the competition results, but also from the detailed analysis of game logs and videos, as presented in the paper. More importantly, and this is one of the clearest evidences, the good competition results were obtained despite the fact that CAMBADA robots clearly move at low speed (2m/s), when compared to most of the main competitors which move faster (2.5-4m/s).

The development of predefined role-based set plays proved to be very efficient both during the development phase, and during their execution in games. More than half of the 73 scored goals are direct result of these set plays.

One of the most significant aspects of this work is the integration of the described coordination methodologies in a complex multi-robot system and their validation in the challenging RoboCup MSL competition scenario. This contrasts with many other approaches described in the literature, which are often validated in more controlled robotic environments, if not in simulation.

Acknowledgment

The CAMBADA team was funded by the Portuguese Government – FCT,POSI/ROBO/43908/2002 (CAMBADA) and currently FCT, PTDC/EIA/70695/2006 (ACORD). We would also like to thank the rest of the CAMBADA team for an excellent work environment.

References

[1] Lesser, V., K. Decker, T. Wagner, N. Carver, A. Garvey, B. Horling, D. Neiman, R. Podorozhny, M. NagendraPrasad, A. Raja, R. Vincent, P. Xuan, and X. Zhang, "Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework," *Autonomous Agents and Multi-Agent Systems*, 9(1), pp. 87-143, 2004.

[2] Noreils, F. R., "Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment," *International Journal of Robotics Research*, 12. pp. 79-98, 1993.

[3] Agmon, N., S. Kraus, G.A. Kaminka, "Multi-robot perimeter patrol in adversarial settings," *Proc. IEEE Int. Conf. on Robotics and Automation*, Pasadena CA, 2008, pp. 2339-2345.

[4] Almeida, L., F. Santos, T. Facchinetti, P. Pedreira, V. Silva and L. Seabra Lopes, "Coordinating Distributed Autonomous Agents with a Real-Time Database: The CAMBADA Project," *Computer and Information Sciences -- ISCIS 2004: 19th International Symposium, Proc.*, C. Aykanat, et al. eds., LNCS, Vol. 3280, pp. 876-886, 2004.

[5] Arbatzat, M., et al., "Creating a Robot Soccer Team from Scratch: the Brainstormers Tribots," *Proc. of Robocup 2003, Padua*, Italy, 2003.

[6] Azevedo, J.L., M.B. Cunha, L. Almeida, "Hierarchical Distributed Architectures for Autonomous Mobile Robots: a Case Study," Proc. *ETFA2007- 12th IEEE Conference on Emerging Technologies and Factory Automation*, Patras, Greece, 2007, pp. 973-980.

[7] Balch, T., and R.C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, 14(6), pp. 926–939, 1998.

[8] Balch, T., and L.E. Parker, *Robot Teams: From Diversity to Polymorphism*, Natick, Massachusetts, A K Peters Ltd, 2002.

[9] Burgard, W., M. Moors, C. Stachniss, and F.E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics and Automation*, 21 (3), pp. 376–386, 2005.

[10] Carpin, S., "Fast and accurate map merging for multi-robot systems," *Autonomous Robots*, 25 (3), pp. 305-316, 2008.

[11] Cheah, C.C., S.P. Hou, J.J.E. Slotine, "Region following formation control for multi-robot systems," *Proc. IEEE Int. Conf. on Robotics and Automation*, Pasadena CA, 2008, pp. 3796-3801.

[12] Cunha, B., J. Azevedo, N. Lau, L. Almeida, "Obtaining the Inverse Distance Map from a Non-SVP Hyperbolic Catadioptric Robotic Vision System," U. Visser, et al., editors, *RoboCup-2007: Robot Soccer World Cup XI*, LNAI, Springer Verlag, Berlin, 2008.

[13] Dahl, T.S., M.J. Mataric, G.S. Sukhatme, "Emergent robot differentiation for distributed multi-robot task allocation," *Proc. 7th International Symposium on Distributed Autonomous Robotic Systems*, Toulouse, France, 2004, pp. 201-210.

[14] De Rosa, M., J. Campbell, P. Pillai, et al., "Distributed watchpoints: Debugging large multi-robot systems," *Proc. IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007, pp. 3723-3729.

[15] Dietl, M., J.-S. Gutmann and B. Nebel, "Cooperative Sensing in Dynamic Environments," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'01)*, Maui, Hawaii.

[16] *EtherCAT Robots win German Open*, Press Release, EtherCAT Technology Group, 8 May 2008, at http://ethercat.org/pdf/english/etg_032008.pdf

[17] Fagiolini, A., M. Pellinacci, G. Valenti, et al., "Consensus-based distributed Intrusion Detection for multi-robot systems," *Proc. IEEE Int. Conf. Robotics and Automation*, Pasadena CA, 2008, p. 120-127.

[18] Ferrein, A., L. Hermanns and G. Lakemeyer, "Comparing Sensor Fusion Techniques for Ball Position Estimation," *RoboCup 2005: Robot Soccer World Cup IX*, A. Bredenfeld, et al., eds., LNCS, 4020, Springer, 2006, pp. 154-165.

[19] Figueiredo, J., N. Lau, A. Pereira, "Multi-Agent Debugging and monitoring framework," *Proc. First IFAC Workshop on Multivehicle Systems (MVS'06)*, Brasil, October, 2006.

[20] Figueiredo, L., I. Jesus, J. Machado, J. Ferreira, and J. Martins de Carvalho, "Towards the development of intelligent transportation systems," *Proc. IEEE Intellig. Transport. Syst*, 2001, pp. 1206–1211.

[21] Gerkey, B.P., and M.J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems", *International Journal of Robotics Research*, 23 (9), pp. 939-954, 2004.

[22] Hafner, R., S. Lange, M. Lauer, and M. Riedmiller, "Brainstormers Tribots Team Description, *RoboCup International Symposium 2008, CD Proc.*, Suzhou, China.

[23] Joyeux, S., R. Alami, S. Lacroix, "A plan manager for multi-robot systems," *Proc. 6th Int. Conf. on Field and Service Robotics*, Chamonix, France, 2007, pp. 443-452.

[24] Kok, J., M. Spaan. and N. Vlassis, "Non-communicative multi-robot coordination in dynamic environments," *Robotics and Autonomous Systems*, 50 (2-3), pp. 99-114, 2005.

[25] Lau, N., L. Seabra Lopes, G. Corrente, "CAMBADA: Information Sharing and Team Coordination," *Autonomous Robot Systems and Competitions: Proc. of the Eigth Conference. 2 April 2008, Aveiro*, Portugal, Universidade de Aveiro, 2008, p. 27-32.

[26] Lauer, M., S. Lange and M. Riedmiller, "Calculating the perfect match: An efficient and accurate approach for robot self-localisation," *RoboCup 2005: Robot Soccer World Cup IX*, A. Bredenfeld, et al., eds., LNCS 4020, Springer, 2006.

[27] Lima, P, L. Custódio, I. Akin, A. Jacoff, G. Kraezschmar, B. Kiat Ng, O. Obst, T. Röfer, Y. Takahashi, C. Zhou (2005) RoboCup 2004 Competitions and Symposium: A Small Kick for Robots, a Giant Score for Science, *AI-Magazine*, 6 (2), 2005, p. 36-61.

[28] Low, K.H., G.J. Gordon, J.M. Dolan, et al., "Adaptive sampling for multi-robot wide-area exploration," *Proc. IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007, pp. 755-760.

[29] Michael, N., M.M. Zavlanos, V. Kumar, et al., "Distributed multi-robot task assignment and formation control," *Proc. IEEE Int. Conf. on Robotics and Automation*, Pasadena CA, 2008, pp. 128-133.

[30] MSL Technical Committee 1997-2008, *Middle Size Robot League Rules and Regulations for 2008. Draft Version - 12.2 20071109*, November 9, 2007.

[31] Neves, A., G. Corrente. and A. Pinho, "An omnidirectional vision system for soccer robots," *Progress in Artificial Intelligence*, LNCS. Berlin, nº 4874, Springer, 2007, pp. 499-507.

[32] Oubbati, M., M. Schanz, T. Buchheim, P. Levi, "Velocity Control of an Omnidirectional RoboCup Player with Recurrent Neural Networks," A. Bredenfeld, et al., Eds., *RoboCup 2005: Robot Soccer World Cup IX*, LNCS, vol. 4020, 2006, 691-701.

[33] Pagello, E., A. D'Angelo, E. Menegatti, "Cooperation Issues and Distributed Sensing for Multirobot Systems," *Proc. of the IEEE*, 94 (7), pp. 1370-1383, 2006.

[34] Pedreiras, P., F. Teixeira, N. Ferreira, L. Almeida, A. Pinho, F. Santos, "Enhancing the reactivity of the vision subsystem in autonomous mobile robots using real-time techniques," *RoboCup-2005: Robot Soccer World Cup IX*, I. Noda, A. Jacoff, et al., eds., LNCS, 4020, Springer, Berlin, 2006, pp. 371-383.

[35] Pedreiras, P., L. Almeida, "Task Management for Soft Real-Time Applications Based on General Purpose Operating Systems," *Robotic Soccer*, Pedro Lima (ed.), Itech Education and Publishing, Vienna, Austria, 2007, pp. 598-607.

[36] Reis, L.P. and N. Lau, and E.C. Oliveira, "Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents," *Balancing Reactivity and Social Deliberation in Multiagent Sytems: From RoboCup to Real Word Applications*, M. Hannenbauer, et al. eds., LNAI 2103, Springer-Verlag, 2001, pp. 175-197.

[37] Reis, L.P., and N. Lau, "FC Portugal Team Description: RoboCup 2000 Simulation League Champion", *RoboCup-2000: Robot Soccer World Cup IV*, P. Stone, et al., eds., LNCS 2019, Springer, 2001, pp. 29-40.

[38] Riedmiller, M., and T. Gabel, "On Experiences in a Complex and Competitive Gaming Domain: Reinforcement Learning Meets RoboCup," *Proc. of the 3rd IEEE Symposium on Computational Intelligence and Games (CIG 2007).* IEEE Press, 2007, pp. 17-23.

[39] Sato, Y., S. Yamaguchi, et al., "Hibikino-Musashi Team Description Paper," *RoboCup Int. Symposium 2008*, CD Proc., Suzhou, China.

[40] Stone, P., and M. Veloso, "Task Decomposition, Dynamic Role Assignment and Low Bandwidth Communication for Real Time Strategic Teamwork," *Artif. Intelligence*, 110 (2), 1999, pp. 241-273.

[41] Tewolde, G.S., G.H. Wu, Y. Wang, et al., "Distributed Multi-robot Work Load Partition in Manufacturing Automation," *Proc. IEEE Int. Conf. on Autom. Science and Eng.*, Arlington VA, 2008, pp. 504-509.

[42] van der Vecht, B., and P. Lima, "Formulation and Implementation of Relational Behaviours for Multi-robot Cooperative Systems," *RoboCup 2004: Robot Soccer World Cup VIII*, Springer LNCS Vol. 3276, 2005, pp 516-523.

[43] Wang, P.K.C., "Navigation Strategies for Multiple Autonomous Mobile Robots Moving in Formation," *Journal of Robotic Systems*, 8(2), pp. 177-195, 1991.

[44] Weigel, T. W., M. Auerbach, B. Dietl, et al., "CS Freiburg: Doing the Right Thing in a Group," *RoboCup 2000: Robot Soccer World Cup IV*, P. Stone et al., eds., Springer-Verlag, 2001, pp. 52-63.

[45] Yamauchi, B., "Frontier-based exploration using multiple robots," *Proc. Second Int. Conf. on Autonomous Agents*, 1998, pp. 47–53.

[46] Zweigle, O., R. Lafrenz, T. Buchheim, U.-P. Käppeler, H. Rajaie, F. Schreiber and P. Levi, "Cooperative agent behavior based on special interaction nets," *Intelligent Autonomous Systems 9*, T. Arai, et al., Eds. Amsterdam, The Netherlands: IOS Press, 2006.