Technical Report TR-2008-01 (January 2008)
© Intelligent Systems Laboratory, ECE, TUC
`www.intelligence.tuc.gr`

PETRI NET PLAN COORDINATION FOR ROBOCUP TEAMS

GEORGIOS KONTES
Department of Electronic and Computer Engineering, Technical University of Crete
Chania, Crete, 73100, Hellas (Greece)
gkontes@isc.tuc.gr

MICHAIL G. LAGOUDAKIS
Department of Electronic and Computer Engineering, Technical University of Crete
Chania, Crete, 73100, Hellas (Greece)
lagoudakis@intelligence.tuc.gr

For more than a decade, the international RoboCup competition promotes scientific research in robotics through soccer games between teams of autonomous robots. Effective coordination between the players of a team is a crucial ability for the success of a soccer team. Unfortunately, research in non-simulated RoboCup leagues has focused mostly on single player skills, demonstrating only limited results in coordinated team play. Team coordination in a human soccer game boils down to various team formations, tactics, and strategies. In our work, we adapt and transfer formations, tactics, and strategies from human soccer to our four-legged RoboCup team. We define roles for each player in each tactic and we implement these roles using Petri Net Plans (PNP). The choice of tactic and the assignment of the corresponding roles to players is performed dynamically during the game depending on the current game state using a simple broadcast-based communication scheme and a finite state machine. Our approach is implemented and tested on our four-legged RoboCup team KOURETES. The proposed coordination scheme can be easily expanded to include additional tactics and can be adapted to RoboCup teams of various size. We believe that Petri Net Plan Coordination can be useful in various robot team applications beyond robotic soccer, such as planetary exploration and search-and-rescue missions.

Keywords: Multi-Robot Coordination; Petri-Net Plans; Autonomous Robot Teams; Robotic Soccer; RoboCup Competition; Soccer Strategies.

1. Introduction

In its short history, the RoboCup competition[1] has grown to a well-established annual event bringing together the best robotics researchers from all over the world. The initial conception by Hiroaki Kitano in 1993 led to the formation of the RoboCup Federation with a bold vision: "*By the year 2050, to develop a team of fully autonomous humanoid robots that can win against the human world soccer champions*". The uniqueness of RoboCup stems from the real-world challenge it poses, whereby the core problems of robotics (perception, cognition, action, coordination) must be addressed simultaneously under real-time constraints. The proposed

Fig. 1. Team KOURETES (red color) at RoboCup German Open 2007.

solutions are tested on a common benchmark environment through soccer games in various leagues, thus setting the stage for demonstrating and promoting the best research approaches, and ultimately advancing the state-of-the-art in the area. Beyond soccer, RoboCup now includes also competitions in search-and-rescue missions (RoboRescue), homekeeping tasks (RoboCup@Home), robotic performances (RoboDance), and simplified leagues for K-12 students (RoboCup Junior).

The Four-Legged League of the RoboCup competition is among the most popular leagues, featuring four robot players (SONY AIBO robots) per team. The AIBO ERS-7 robot resembles a small pet dog and is equipped with a 576MHz 64-bit RISC processor, 64 Mb of RAM, a 350K pixel CCD camera, three infrared distance sensors, touch sensors, wifi communication, and a number of servos on its legs, head, tail and ears, providing 20 degrees of freedom in total. Games take place in a $4m \times 6m$ field marked with white lines, colored goals, and two colored beacons which serve as landmarks (Figure 1). Each game consists of two 10-minute halves and teams switch colors and side at halftime. There are several rules enforced by human referees during the game. For example, a player is punished with a 30-seconds removal from the field if he performs an illegal action, such as pushing an opponent for more than three seconds, grabbing the ball between his legs and head for more than three seconds, leaving the field, or entering his own goal area as a defender.

The main characteristic of the Four-Legged League is that no hardware changes are allowed; all teams use the exact same robotic hardware and differ only in terms of their software. This convention results to the league's characterization by a unique combination of features: autonomous vision-based player operation, legged locomotion and action, uniform robotic platform. Given that the underlying robotic hardware is common for all competing teams, research efforts have focused on developing more efficient algorithms and techniques for visual perception, active localization, omnidirectional motion, skill learning, and coordination strategies. During the course of the years an independent observer could easily notice a clear progress in all research directions related to single robot skills and abilities, however, one can also notice that little progress has been made at the robot team level.

The ability to coordinate within a team is certainly a crucial factor to the success of the team. Coordination comes in various forms from a player perspective: What should I do next? How can I collaborate with my teammates? How can the team act in a coordinated manner? In order to address such questions, the robotic players need to develop explicit mechanisms to support coordinated team action. In order to promote research efforts in coordinated team play, the RoboCup committee designed the passing challenge in years 2006 and 2007. According to the rules of the challenge, three robots are placed on the field in a triangular formation; they must pass the ball around without moving away from their initial position and they score points for accurate passing and successful grabbing. The ability of successful passing, even though it could be seen as a single robot skill, underlies many forms of team coordination in robot soccer. Given an accurate passing mechanism, team coordination in a soccer game extends to various team formations and strategies depending on the actual state of the game. Players can choose between different ways of positioning themselves on the field in order to move the ball faster, trick the opponents, and eventually score goals.

Team formations, tactics, and strategies is largely an unexplored area in the RoboCup research, with the exception of the Simulation leagues. In our endeavors as a RoboCup team competing in the Four-Legged league, we decided to take a radical step in behavior control and implement robot soccer strategies, which are inspired by human soccer strategies used in real soccer games. Taking under consideration that the ultimate goal of RoboCup is a game between robots and professional human soccer players, we believe that our current and ongoing work takes a critical step towards this goal. More specifically, we adapt and transfer formations, tactics, and strategies from human soccer to our four-legged RoboCup team. We define roles for each player in each tactic and we implement these roles using Petri Net Plans (PNP). The choice of tactic and the assignment of the corresponding roles to players is performed dynamically during the game depending on the current game state using a simple broadcast-based communication scheme and a finite state machine. Our approach has been implemented on our four-legged RoboCup team KOURETES and was tested during the games of RoboCup German Open 2007 and the demonstration games played at the 2007 Hi-Tech Innovators Partenariat Exhibition in Thessaloniki, Greece. The hierarchical structure of our coordination scheme allows for easy incorporation of additional tactics at a high level. It can also be extended to RoboCup teams with more robots, however new tactics with additional roles need to be developed in such case. The work presented in this paper is by no means limited to robotic soccer. In fact, the need to coordinate multiple robots arises in any application involving multiple robots. Such applications include planetary exploration, search-and-rescue missions, collaborative assembly, and area surveillance. The proposed scheme can be adapted to various robot domains, optionally in conjunction with other coordination methods, such as market-based methods[2] and auction-based coordination[3].

The remaining of the paper is structured as follows: Section 2 makes the connection between human and robot soccer play, while Section 3 defines the tactics we use, the corresponding roles, and the mechanism for tactic switching. Section 4 reviews the language of Petri Net Plans and describes the encoding of all roles as Petri-Net Plans. Finally, Section 6 summarizes the results of our experimentation and Section 7 reviews related work in the area and lists our future research plans.

2. From Human to Robot Play

Each team in a human soccer game consists of eleven players; one of them assumes the role of the goal keeper, whereas the others can freely move around the field. One of the oldest and most popular team formations is the so-called 4-4-2 formation (also described as 4-1-2-1-2 or 4-3-1-2) shown in Figure 2 (left). This is the favorite formation of the Argentina national soccer team; also, FC Porto under coach Jose Mourinho won the Champions League in 2004 playing mostly with this formation[4].

Fig. 2. The 4-4-2 soccer formation[4] (left). Our robot team formation for offense (middle). Our robot team formation for defense (right).

In a four-legged robot soccer game, each team consists of only four players (robots), one of them being the goal keeper. In order to adapt the 4-4-2 system, we assume that the RoboCup field corresponds to the center of an actual soccer field and we focus on the middle diamond formation. Since we lack a player in the middle we organize the players in a triangular formation. The actual positions are different for offense and defense as shown in Figure 2 (middle, right). In our work we consider four different tactics for applying the aforementioned formation: two for offense (Counter Attack and Passing Attack) and two for defense (Pressing Defense and Passive Defense). The decision to switch between tactics is made dynamically during the game.

In human soccer, when a defending team wins the ball, it strives to pass the ball from the Defenders to the Midfielders and from the Midfielders to the Attackers. Meanwhile, the Attackers and the Midfielders try to spread along the wings of the field, in order to break the opponent's defense and score.

In our RoboCup team, the switch from defense to offense is implemented with two different tactics. In the Counter Attack tactic, the robot which wins the ball attacks towards the opponent's goal supported by the robot further ahead, while the remaining robot stays behind and guards. Alternatively, according to the Passing Attack tactic, the robot with the ball tries to pass the ball to the robot further ahead in which case they switch to Counter Attack. In either case, a lost ball signals a switch from Offense to defense.

In human soccer, when the attacking team loses the ball, all the teammates, except the attackers, try to get themselves behind the ball. Everyone, including the attackers, is pressing for a mistake in the opponent team. However, when the team is strongly defensive, even the attackers move behind the ball line.

In our RoboCup team, we implement the switch from offense to defense with two different tactics depending on the current state of the game. If a robot loses the ball, the tactic switches to Pressing Defense, except when the ball makes it into the opponent's half of the field and there is no teammate anywhere near the ball, in which case the tactic switches to Passive Defense. In both cases, a hold on the ball signals a tactic switch from defense to offense.

3. Roles in Tactics

In this section we provide a brief description of the role of each player in each tactic. Each player has a role in the field which can change dynamically during the game depending on the current player and ball locations. In our work, we consider three roles, the Attacker (ATT), the Defender (DEF), and the Midfielder (MID), since the Goal Keeper role does not change over time. Each role corresponds to a different behavior depending on the tactic followed at the time as described below.

Counter Attack [Figure 3(left)]

ATT The Attacker dribbles with the ball towards the opponent's goal straight from the place where he won the ball. When he reaches the opponent's goal area, he either shoots directly to the goal or passes the ball to the Midfielder who awaits on the opposite side of the field at the corner of the penalty area. If such a pass occurs, the Attacker notifies the Midfielder to catch the ball.

MID The Midfielder supports the Attacker, positions himself at the opposite side (at the corner of the penalty area), and waits for a pass from the Attacker. If he receives a successful pass, he shoots to the goal.

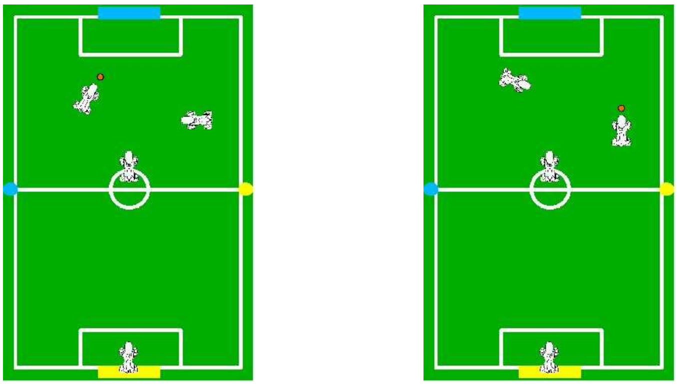

Fig. 3. The Counter Attack formation (left). The Passing Attack formation (right).

DEF The Defender stays at the center of the field, just past the center circle. In case the ball bounces back to him, he shoots towards the goal.

Passing Attack [Figure 3(right)]

MID The Midfielder dribbles towards the opponent's goal from the place he won the ball, looking for opportunities to pass the ball to the Attacker. Once a pass is made, he notifies the Attacker to catch the ball.

ATT The Attacker slowly moves towards the opponent's penalty area constantly facing the Midfielder. While moving or after reaching the opponent's penalty area, he looks for a pass from the Midfielder. As soon as a pass is successfully received, the tactic switches to Counter Attack.

DEF Same as in the Counter Attack above.

Pressing Defense [Figure 4(left, middle)]

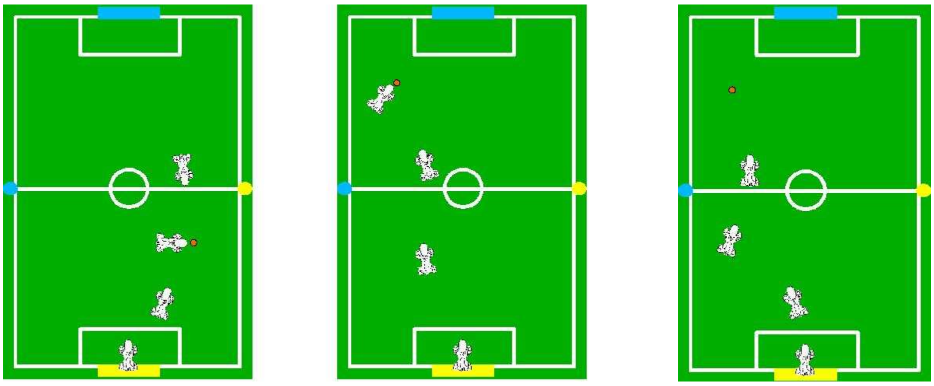

Fig. 4. The Pressing Defense formation (left, middle). The Passive Defense formation (right).

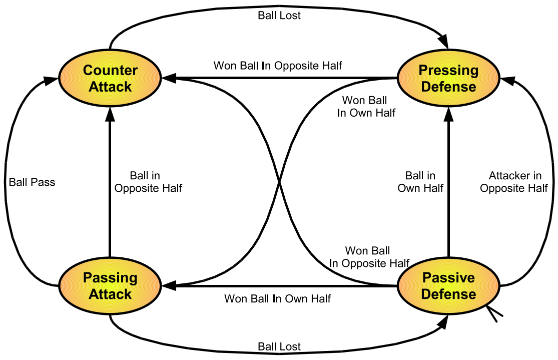

Fig. 5. FSM for Switching Tactics.

ATT While the ball is on the opponent's half of the field, the Attacker is chasing after the ball. When the ball enters his own half of the field, the Attacker moves around the middle line, continuously tracking the ball and placing himself between the ball and the opponent's goal.

MID While the ball is on the opponent's half of the field, the Midfielder moves in front of the middle line and places himself between the ball and the own goal. Only when the ball enters his own half of the field, he is chasing after the ball. He acts similarly to the Attacker, but in complementary halves of the field.

DEF The Defender places himself between the ball and the own goal, with the only restriction that he doesn't move past the middle line, even when the ball is on the opponent's half of the field.

Passive Defense [Figure 4(right)]

ATT While the ball is inside the opponent's half of the field, the Attacker moves along the middle line and places himself between the ball and the own goal. As soon as the ball enters his own, the tactic switches to Pressing Defense.

MID The Midfielder supports the Attacker by staying behind him visually tracking the ball at all times.

DEF The Defender stays between the ball and the own goal in front of his own penalty line.

The tactic to be played at any moment is selected dynamically depending on the current situation of the game. Coordinated tactic selection between the robots takes place through a Finite State Machine (FSM), which implements a simple selection protocol. Each robot executes its own local copy of the FSM and the uniqueness of role allocation is guaranteed by appropriate conditions over the information shared between the robots. In particular, the FSM is triggered by the signals broadcast by the robots, whenever a switch of tactic is deemed necessary. Any information needed by the FSM to make a transition, such as a pass attempt and ball/player position, is provided directly by the robots. The complete state diagram of the FSM along with the necessary conditions is shown in Figure 5.

Apparently, any robot winning/lossing the ball signals the team to choose an offending/defending tactic. While on offense/defense, a pass, the ball position, or the attacker location may also change the tactic. Transitions to the same state are possible in case of false alarms for tactic switch. Finally, if a player is penalized, the team automatically switches to Passive Defense (with a missing Attacker).

4. Petri Net Plans Role Implementation

We realized the roles of the players using Petri Nets Plans[5], expanding the infrastructure of the SPQR-Legged team (Italy). The formalism of Petri Net Plans allows for complex action interactions, such as synchronization, concurrency, non-instantaneous execution, and interrupts.

A Petri Net[6,7] is a graphical language used in modeling dynamical systems. It allows the description of a system in terms of a weighted directed graph, where the nodes denote places (circles) or transitions (boxes) and the edges between them represent possible paths of execution as well as certain conditions (weights). The flow of execution is denoted by means of tokens, which move between places causing transitions to "fire", if all the specified conditions are met. The Extended Petri Net variation[7] allows for inhibitory edges, whose firing behavior is complementary to that of the regular edges.

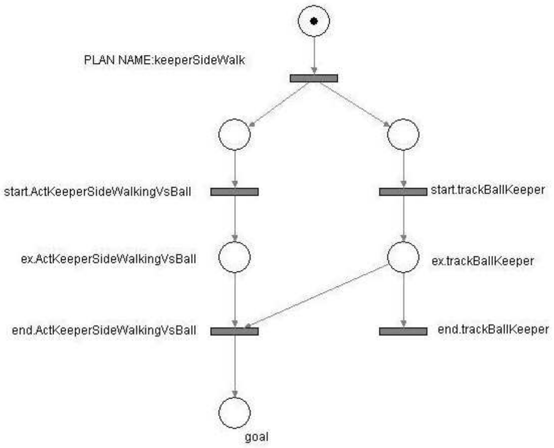

Fig. 6. Simple Petri Net Plan for moving to the ball while tracking it.

A Petri Net Plan[5] is a collection of actions structured as an Extended Petri Net with unweighted edges. Each action in the plan is explicitly described in three phases (initiation, execution, termination) corresponding to an equal number of places with two transitions between them. This description allows for constructing sequences, loops, splits, joins, and parallel branches of actions. Petri Net Plans have been successfully used in various robot domains, such as RoboCup, search-and-rescue, and multi-agent communication.

Figure 6 illustrates a simple Petri Net Plan. It consists of two actions (`ActKeeperSideWalkingVsBall` and `trackBallKeeper`) and its purpose is to move the goalkeeper sideways towards the side of the ball. This PNP performs a fork operation, allowing the two actions to be deployed at the same time, resulting in a parallel execution of the side walking of the robot and the simultaneous tracking of the ball. When the action `ActKeeperSideWalkingVsBall` finishes its execution, the action `trackBallKeeper` finishes as well, since the `end.ActKeeperSideWalkingVsBall` transition fires. The resulting place with the label `goal`, ends the execution of this PNP and returns.

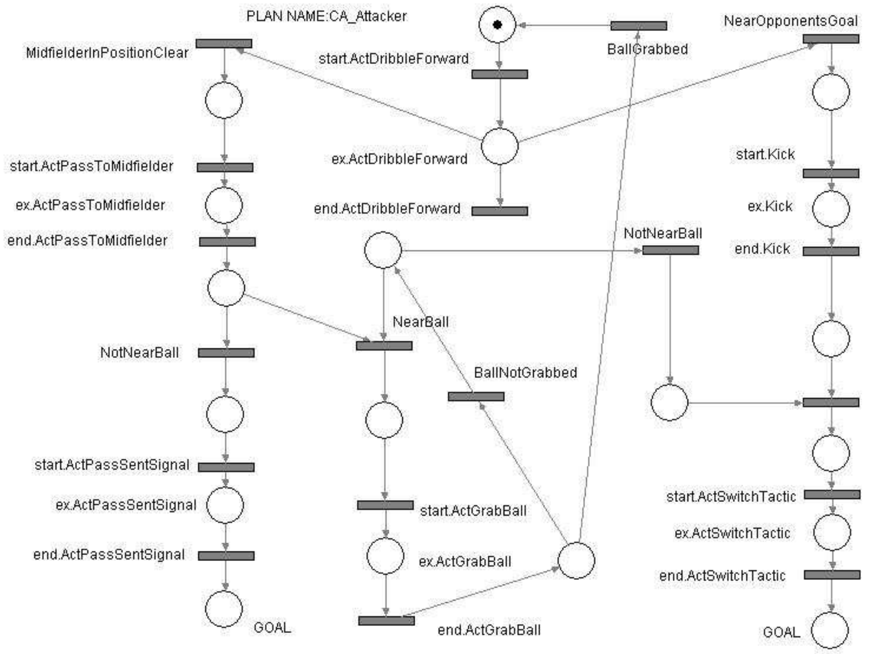

Fig. 7. PN Plan for the Attacker role in the Counter Attack tactic.

Each Petri Net Plan is defined in terms of various base actions. The degree of complexity hidden inside the base actions is inversely proportional to the degree of complexity in the plans. We have chosen a set of base actions, which strikes a balance between the complexity of plans and actions, guided by the desire to design reusable actions that can be used in many different plans. The different naming style of the actions denotes their structural difference: all actions beginning with the label `Act` have been implemented using procedural code and therefore consist of direct

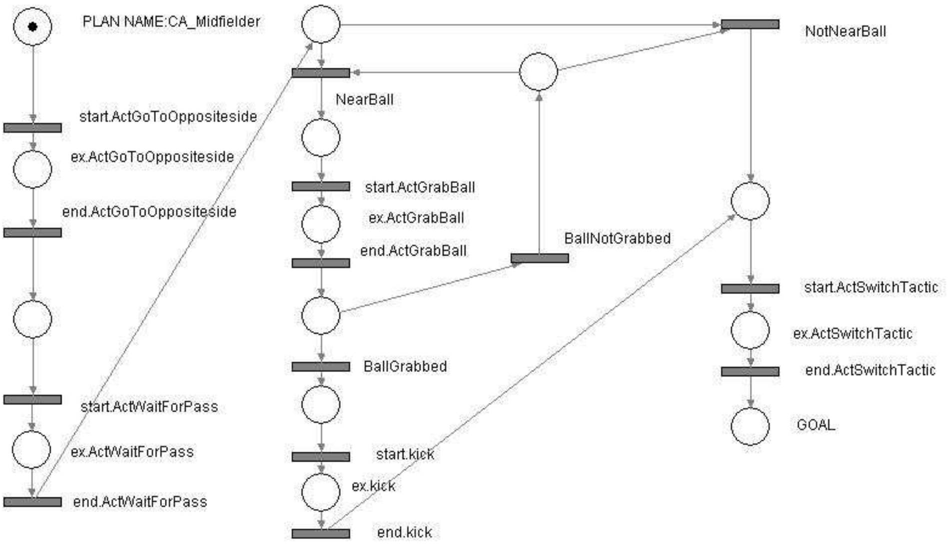

Fig. 8. PN Plan for the Midfielder role in the Counter Attack tactic.

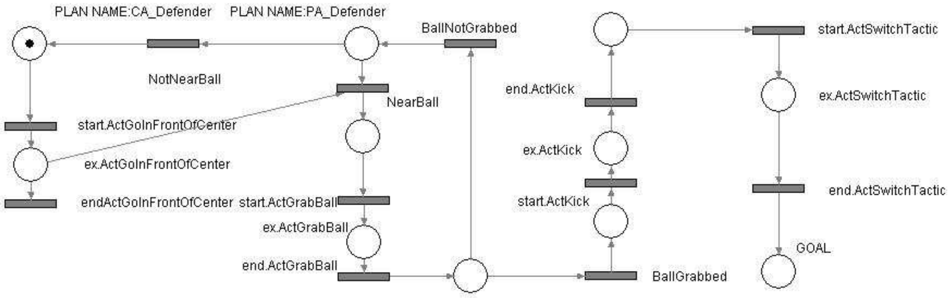

Fig. 9. PN Plan for the Defender role in the Counter and Passing Attack tactics.

C++ calls, whereas actions without the label `Act` refer to other PNP structures which can be invoked whenever the corresponding action is executed. This leads to a hierarchical construction of PNPs with resuable parts.

The Petri Net Plan implementing the Attacker role for the Counter Attack tactic is shown in Figure 7. Execution of this plan begins at the place with the single token (black dot). The first action (`ActDribbleForward`) is executed continuously until it is interrupted by one of two events. Either the Attacker reaches the opponent's goal area (condition `NearOpponentsGoal`) or figures out that the Midfielder is in position and cleared for a pass (condition `MidfielderInPositionClear`). In the first case, the Attacker executes the action `kick` and signals the team to choose the next tactic. In the second case, the Attacker executes the `ActPassToMidfielder` action

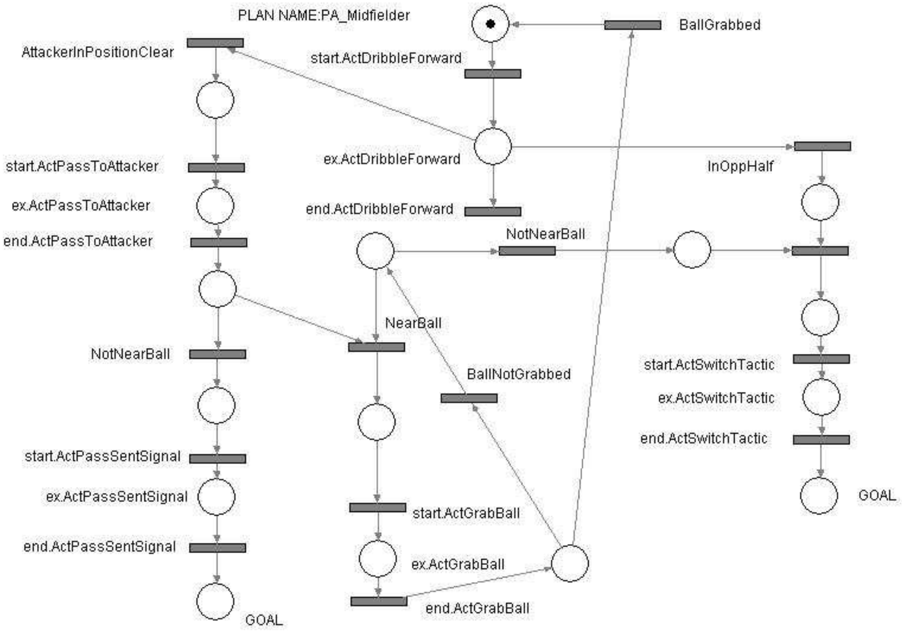

Fig. 10. PN Plan for the Midfielder role in the Passing Attack tactic.

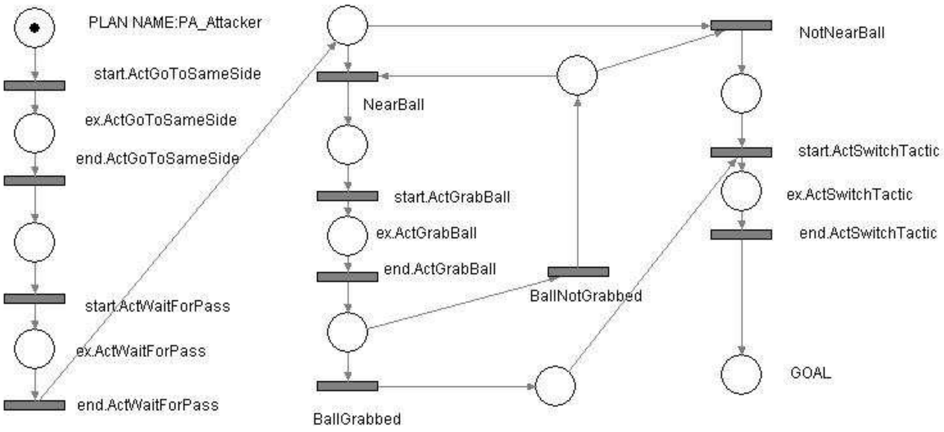

Fig. 11. PN Plan for the Attacker role in the Passing Attack tactic.

to pass the ball. A sensing action follows. If the ball cannot be seen anymore near the Attacker, he notifies the Midfielder that a (hopefully successful) pass was made. However, if and as long as the ball is still seen near the Attacker, he repeatedly attempts to grab it (action `ActGrabBall`) and repeat the plan. Otherwise, if the ball cannot be seen anymore near the Attacker, he signals the team to choose the

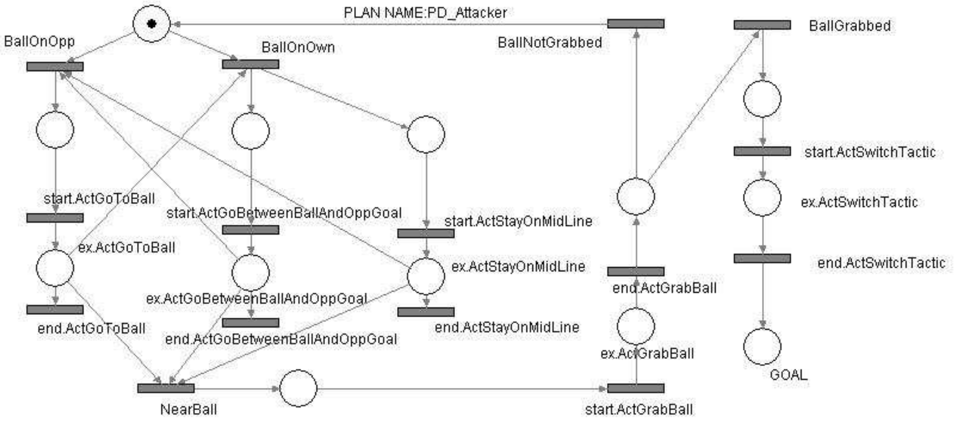

Fig. 12. PN Plan for the Attacker role in the Pressing Defense tactic.

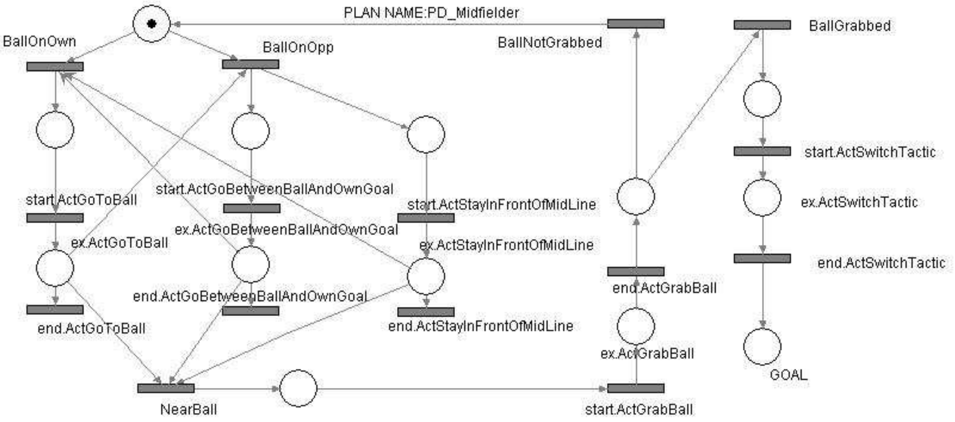

Fig. 13. PN Plan for the Midfielder role in the Pressing Defense tactic.

next tactic.

The Petri Net Plan for the Midfielder role in the Counter Attack tactic is shown in Figure 8. The Midfielder moves to the opposite side of the field and waits for a pass (along with the signal). If the pass is successful (`NearBall`), he grabs and kicks the ball to the opponent's goal, otherwise he repeatedly tries to grab the ball as long as it can be seen near the robot. In any case, if the ball moves far away from the Midfielder, he calls for a switch of tactic.

The Petri Net Plan for the Defender role in the Counter Attack tactic is shown in Figure 9. The Defender takes position in front of the center circle and waits. In the unexpected event that the ball bounces back to him, he attempts to grab it and kick it towards the opponent's goal. If successful, he signals for a switch of tactic,

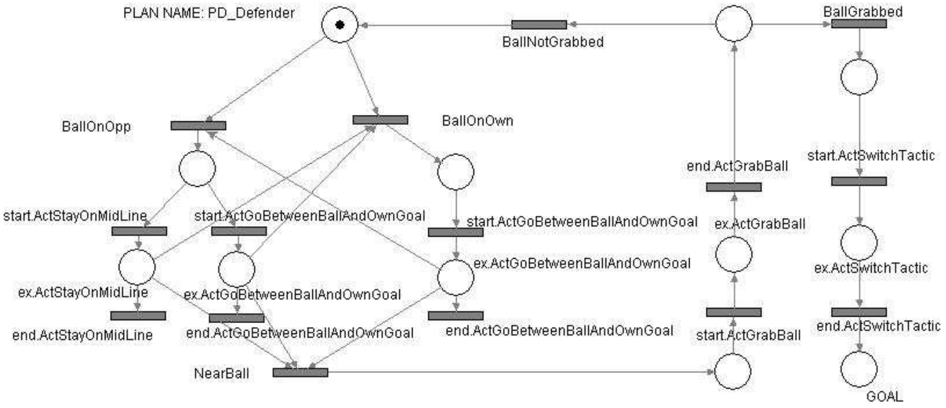

Fig. 14. PN Plan for the Defender role in the Pressing Defense tactic.

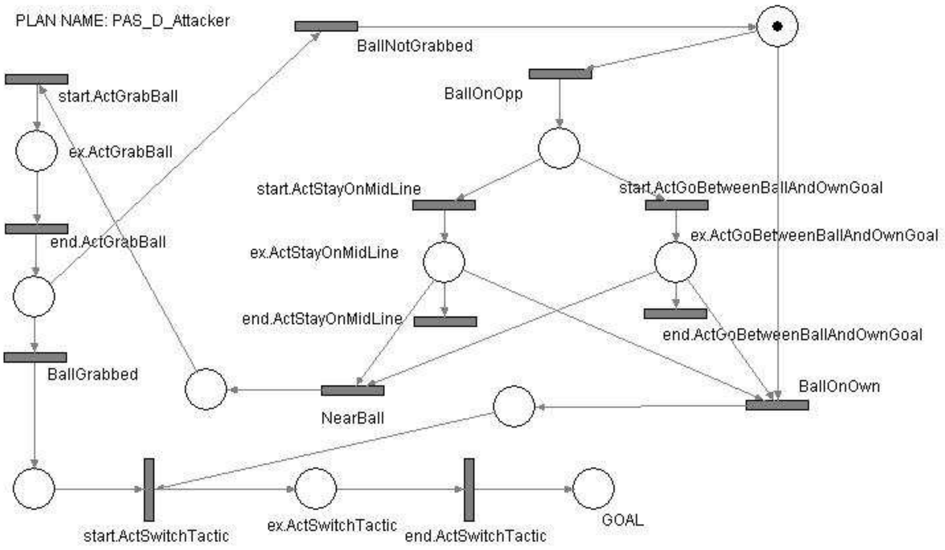

Fig. 15. PN Plan for the Attacker role in the Passive Defense tactic.

otherwise he returns to the top of the plan.

The Petri Net Plan for the Midfielder role in the Passing Attack tactic is shown in Figure 10. The Midfielder starts by moving forward until one of two events occurs. Either he enters the opponent's half and signals a switch of tactic to Counter Attack or identifies the Attacker further ahead and passes the ball. Similarly to the Attacker in Counter Attack, he either notifies for a possibly successful pass or repeatedly tries to recover the ball and repeat the plan.

The Petri Net Plan for the Attacker role in the Passing Attack tactic is shown

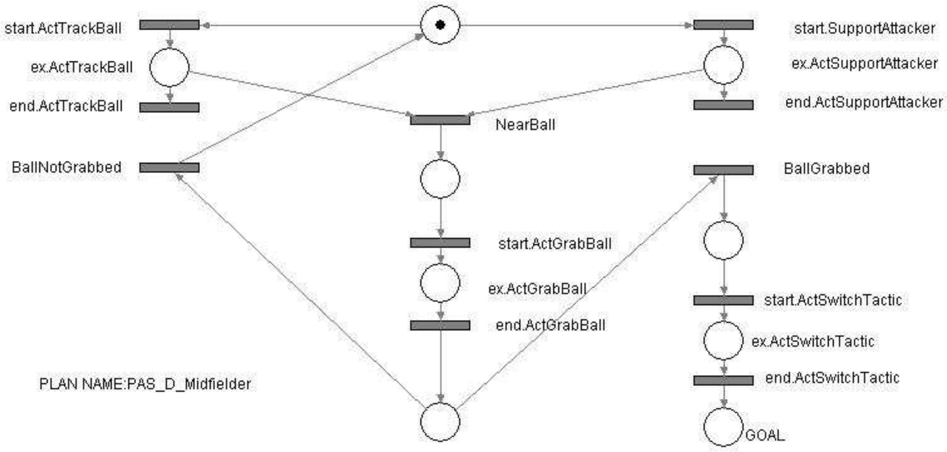

Fig. 16. PN Plan for the Midfielder role in the Passive Defense tactic.

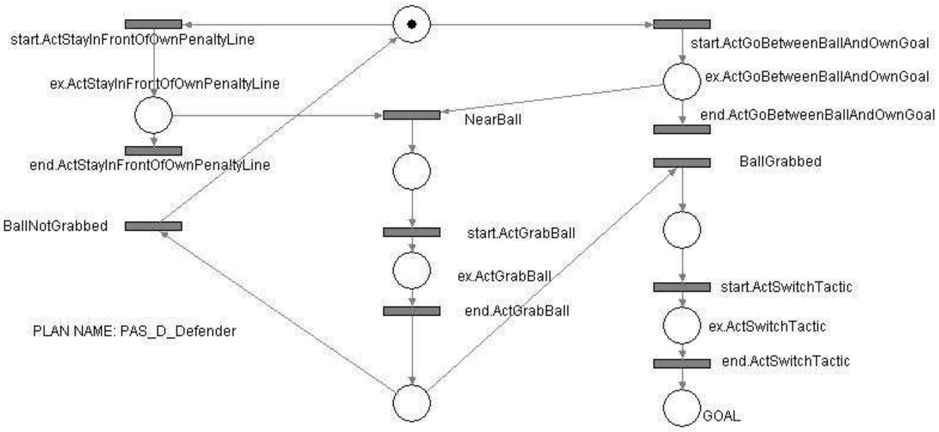

Fig. 17. PN Plan for the Defender role in the Passive Defense tactic.

in Figure 11. The Attacker moves to the same side of the field as the Midfielder, but further ahead into the opponent's half, constantly facing backwards, and waits for a pass. If the pass is successful, he grabs the ball and calls for a switch of tactic, otherwise he repeatedly tries to grab the ball as long as it is seen near the robot. If the ball is lost, he calls for a switch of tactic.

The Petri Net Plan for the Defender role in the Passing Attack tactic is identical to the one in the Counter Attack shown in Figure 9.

The Petri Net Plan for the Attacker role in the Pressing Defense tactic is shown in Figure 12. Initially, a choice of course is made depending on the current location of the ball. If the ball is on the opponent's half of the field, the Attacker continuously chases after the ball. If the ball is in his own half, then a fork occurs and the Attacker continuously executes two actions in parallel; one which keeps him around the middle line and one which moves him between the ball and the opponent's goal. If the ball switches from one half to another, the choice is adjusted appropriately. Finally, the Attacker breaks out of these loops, if he finds himself next to the ball. In this case, he repeatedly attempts to grab it; if successful, he signals a switch of tactic to Offense, otherwise he repeats the plan.

The Petri Net Plan for the Midfielder role in the Pressing Defense tactic is shown in Figure 13. The careful reader will notice that this plan is almost identical to that of the Attacker. The only difference is that the conditions `BallOnOpp` and `BallOnOwn` have been switched. This is expected as these two roles are complementary with respect to the two halves of the field.

The Petri Net Plan for the Defender role in the Pressing Defense tactic is shown in Figure 14. The Defender never chases after the ball, but positions himself between the ball and his own goal, without ever leaving his own half.

The Petri Net Plan for the Attacker role in the Passive Defense tactic is shown in Figure 15. Again, a choice is made depending on the current location of the ball. If the ball is found in the own half of the field, the Attacker signals a switch of tactic to Pressing Defense. Otherwise, he stays around the middle line between the ball and his own goal, looking for opportunities to grab the ball.

The Petri Net Plan for the Midfielder role in the Passive Defense tactic is shown in Figure 16. The Midfielder supports the Attacker by staying behind him and at the same time visually tracks the ball. He breaks out of this loop only if he manages to steal the ball or a signal is received to switch tactic.

Finally, the Petri Net Plan for the Defender role in the Passive Defense tactic is shown in Figure 17. This plan is almost identical to that of the Midfielder except that the Defender moves in front of the own penalty line and tries to position himself between the ball and the own goal. Again, he breaks out of this loop only if he manages to steal the ball or a signal is received to switch tactic.

5. Robot Communication

The last piece of our coordination scheme is the mechanism for passing signals between robots triggering tactic switches. Communication between robots in the four-legged RoboCup league faces several challenges. Besides signals transmitted through the physical world (e.g. sound or light), robots are allowed to exchange information through a low-bandwidth, unreliable, wireless network. This fact implies that only limited amounts of information can be exchanged, messages are not guaranteed to be delivered to all recipients, several packets may be delivered with delay or out of order, and simultaneous transmission of messages will result in loss with high probability. In addition, experience has shown that the entire network can go down unexpectedly or individuals robots can go off-line due to hardware failure. It is therefore understood that, under these conditions, careful network usage is required for reliable communication.

In our implementation, all communication takes the form of messages sent by a single robot to all other team members. This is required because each player executes its own local copy of the FSM and therefore trigger events must become common knowledge to ensure that all robots switch to the same tactic. Note that any tactic switching in the FSM is triggered by a single message. Thus, there is no need for simultaneous and/or synchronized messages and as a consequence the network is not congested with multiple different messages, which need to go through at the same time.

We use the UDP protocol for broadcasting each message to all robots including the original sender. Any player originating a message adds a special tag to the message indicating its own identity and a time stamp with the original submission time. Given that only a single message is broadcast over the network at any point in time, all players in the team (including original senders) are required to retransmit any message they receive stamped with the most recent submission time to ensure that all messages eventually reach all players. Messages stamped with an older submission time are not retransmitted and eventually cease from the network. In other words, the message that was broadcast last over the network is continuously circulated until the next message appears. We adopted this scheme to cope with the unreliability of the network. Given that the size of all these messages is rather small, even the resulting continuous traffic on the network does not lead to network congestion or violation of the bandwidth limit.

In order to validate our communication scheme, we used a simple experimental scenario, illustrated in Figure 18(left). As long as the ball is not seen by robot 1, robots 1 and 2 seek the ball, while robots 3 and 4 remain idle; this behavior reverses if the ball is seen by robot 1, so that robots 1 and 2 remain idle as long as the ball is visible and robots 3 and 4 seek the ball. The signal for behavior switch is initiated by robot 1 and is communicated to robots 3 and 4 via robot 2. In particular, when the ball appears/disappears from the visual field of robot 1, robot 1 broadcasts a message that needs to reach robot 2. Once robot 2 receives the message, he broadcasts a new message that needs to reach robot 3 robot 4. This behavior switch can be regarded as a typical tactic switch triggered by a single robot. We measured the percentage of successful delivery of all messages and, therefore, successful behavior switch over many repetitions of the experiment. Results with the original communication scheme, shown in Figure 18(middle), yield a prohibitively low percentage of success. In contrary, our communication scheme, shown in Figure 18(right), achieves delivery times to all recipients comparable to the network lag (1.5 seconds) for 72% of the messages and successful delivery of virtually all messages (99.7%) within 10 seconds. This result was obtained using a standard inexpensive wireless access point.

If a robot is inactive (does not control the ball) and receives no message for a predefined period of time (currently, 3 seconds), it assumes that the network is down and adopts the most conservative tactic (Passive Defense) until a new message arrives. That also covers the possibility of a robot being off-line, which is not rare in RoboCup games. This convention may result in a robot temporarily adopting a tactic that is possibly different than the team tactic, however it prevents the robot from entering a deadlock, where it waits indefinitely for an incoming message before taking action.

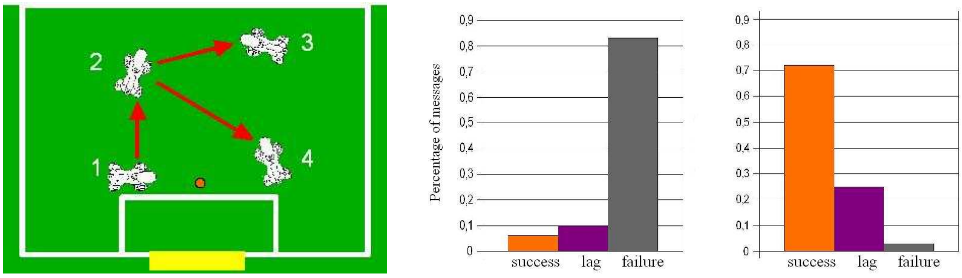

Fig. 18. Experimental Setup (left). Original (middle) and Improved (right) Communication

The passing mechanism needed for the offense tactics is also realized using communication; teammates exchange over the network the necessary information for making a successful pass. In particular, as soon as a robot is ready to pass the ball, it broadcasts an *intent-to-pass* signal along with its own coordinates in the field (available through the localization module). Provided a robot is available to receive the pass, it orients itself facing the passing robot, broadcasts its own position, and waits. The passing robot kicks the ball towards the receiving robot and broadcasts a *sent-pass* signal. Assuming that the area between them is not obstructed and given the fairly good accuracy of localization, there is high chance that the ball will end up near the receiving robot, thereby realizing a successful pass.

6. Results

For successful operation, our approach must be supported by an accurate localization module (for moving to and broadcasting correct positions), a passing module (for passing the ball in the correct direction), and a communication module (for transmitting signals). To provide these support services, we embedded our PNP coordination scheme within the software architecture of our RoboCup team Kouretes along with the modifications in communication and passing mentioned above. Our software architecture is an offspring of the software architecture of team SPQR-Legged 2006 (Italy), which in turn is an offspring of the software architecture of the German Team 2004 (Germany).

The performance of the proposed coordination approach cannot be demonstrated on paper, therefore we have produced video clips showing the roles and tactics in action[a]. Home laboratory tests revealed that indeed teammates make better use of the field with better positioning, even in situations where only a single robot is aware of the ball position. As indicated by our experiments captured on video, we have reasons to believe that the proposed coordination approach enforces a good degree of teamwork which was largely missing from the four-legged league.

The proposed coordinated team play and tactics were also employed during the games of the RoboCup German Open 2007 competition, as well as during the demonstration games played at the 2007 Hi-Tech Innovators Partenariat Exhibition in Thessaloniki, Greece against Team Cerberus (Turkey). Unfortunately, they were not fully utilized due to severe vision and localization problems in both cases and thus performance of PNP coordination in practice during actual games could not be fully assessed. Nevertheless, we observed a drawback of our static role assignment: if the midfielder and the defender are penalized and removed during passive or pressing defense (this can happen easily if they enter the own goal area due to a localization error) and the ball moves into the own half of the field, the attacker will remain near the middle line tracking the ball, totally unaware of the missing co-players, and without offering any help to the poor goalkeeper. Ideally, the role of the attacker would have changed to defender (so, that the own goal area is protected), whereas a penalized would assume the attacker role upon return to the field. A dynamic role assignment would likely resolve this problem.

7. Related and Future Work

Most teams in the four-legged league follow a role switching model for coordination. Team ARAIBO[8] emphasizes individual skills and implements a rather simple role switching module. All robots calculate the time they need to get to the ball and transmit it through the network; the robot with the smallest time becomes the attacker, the robot with the second smallest time becomes the supporter, and the remaining robot assumes a predefined role. Despite its simplicity, this approach is not robust against network failures and may lead to a poor offensive strategy. A similar strategy is followed by the German Team[9]. Again, robots assume roles according to the time they need to get to the ball, however there is a backup role assignment in case of network failures. Robots are required to share their individual cost functions over the network at all times. The rather complicated backup strategy depends solely on the position of the ball and may lead to bad formations in certain cases (e.g. when the ball is on the middle line).

Other teams follow a higher-level team strategy approach. In team Cerberus[10], robots exchange bids and run a virtual auction over the network to decide which robot will do what. Despite the low network utilization, crucial tasks may end up to inappropriate robots, if bid information is lost over the network. Team NUbots[11] employs a scheme, which takes into account the current score and the remaining

---

[a] Available from `http://www.intelligence.tuc.gr/kouretes`

time of the game. The field is divided into localized positions (left defender, center defender, etc.) and the robots can take upon one of two roles: chasing or positioning. The robot with the smallest distance to the ball takes the chasing role, while the other two robots go to the closest localized positions. Network traffic is kept at minimum levels, however the team does not always act in a coordinated manner, due to poor communication between the robots.

The main difference of our approach is that coordination takes place at team and not player level. The team decides on a common tactic (through the current message on the network and the FSM) and each robot automatically knows its role within that tactic and the conditions under which the tactic and roles may change. Therefore, there is no need for low-level, intense communication to determine role assignments to robots. This abstraction adds flexibility to switching between a variety of predefined simple or complex tactics, as well as robustness against network problems. Nevertheless, our work bears many similarities to the work of McMillen and Veloso for Team CMDash (U.S.A.)[12]. They too define tactics which they call "plays". There are fixed roles in each play, but the roles are localized, meaning that each robot chases after the ball when it enters its own area and assumes a convenient position when the ball leaves its area. The main problem of this approach is that robot areas overlap, therefore requiring extra effort for avoiding having two robots chase after the ball simultaneously. In addition, the role assignment is done once at the beginning of the play according to the area each robot is closer to, thus requiring some degree of negotiation at each switch, resulting in lost time and network congestion. Finally, the decision about the current play is made by a single robot (team leader), in contrast to our scheme where such decisions are made locally and individually, with no network overloading.

In the future, we plan to extend our coordination method along two dimensions. Auction-based coordination[2,3] borrows methods and techniques from economics (auctions, trading) and applies them to agent problems. Tasks are assigned to agents through an auction procedure, whereby agents place bids on tasks and winners are determined through a clearance procedure that optimizes a team objective. Auctions could be used for efficient dynamic role assignment with a chosen team tactic. In addition, multi-agent reinforcement learning offers opportunities for collaborative multi-agent learning (where many agents learn to collaborate as a team)[13] and competitive multi-agent learning (where two teams learn to compete against each other, but collaborate within the team)[14]. The scaling properties of these algorithms through exploitation of domain knowledge make them attractive for the RoboCup domain as factorization of the representation can be done on the basis of proximity between players during a game. Using such learning methods, appropriate tactic switching decisions could be learned directly through trial-and-error during actual RoboCup games, thereby enhancing team coordination with adaptive capabilities and replace the current static FSM-based scheme.

8. Conclusion

We presented a method for achieving coordinated team play in a RoboCup team, a feature that has not been emphasized in many RoboCup leagues. The formalism of Petri Net Plans allows the specification of tactics and roles for coordination in an intuitive and systematic way. The proposed method can be adapted and used beyond RoboCup to any multi-robot domain with well-defined roles and objectives.

Acknowledgments

We would like to thank Fr. Demetrios Alexandrakis and the parish of Panagia in Kounoupidiana, as well as the Technical University of Crete. This research was partially supported by the Marie Curie International Reintegration Grant MIRG-CT-2006-044980 within the 6^{th} European Community Framework Programme.

References

1. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, , and H. Matsubara, “Robocup: A challenge problem for AI,” *AI Magazine*, vol. 18, no. 1, pp. 73–85, 1997.
2. M. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-based multirobot coordination: a survey and analysis,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.
3. M. Lagoudakis, V. Markakis, D. Kempee, P. Keskinocak, S. Koenig, C. Tovey, A. Kleywegt, A. Meyerson, and S. Jain, “Auction-based multi-robot routing,” in *Proceedings of Robotics: Science and Systems*, 2005, pp. 343–350.
4. Anonymous, “Formation (football),” *Wikipedia*, 2007.
5. V. A. Ziparo and L. Iocchi, “Petri net plans,” in *ATPN/ACSD Fourth International Workshop on Modelling of Objects, Components, and Agents*, 2006, pp. 267–289.
6. T. Murata, “Petri nets: Properties, analysis and applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
7. J. L. Peterson, *Petri Net Theory and the Modeling of Systems.* Upper Saddle River, NJ: Prentice Hall, 1981.
8. K. Takeshita, T. Okuzumi, S. Kase, Y. Hasegawa, H. Mitsumoto, R. Ueda, K. Umeda, H. Osumi, , and T. Arai, “Technical report of team araibo,” The University of Tokyo and Chuo University, Japan, Tech. Rep., 2006.
9. T. Rofer, R. Brunn, S. Czarnetzki, M. Dassler, M. Hebbel, M. Jungel, T. Kerkhof, W. Nistico, T. Oberlies, C. Rohde, M. Spranger, C. Zarges, and et al, “GermanTeam 2005,” Humboldt-Universitat zu Berlin, Universitat Bremen, Technische Universitat Darmstadt, and Dortmund University, Tech. Rep., 2005.
10. H. L. Akin, C. Mericli, B. Gokce, F. Geleri, N. Tasdemir, B. Celik, and M. Celik, “Cerberus’06 team report,” Bogazici University, Turkey, Tech. Rep., 2006.
11. M. J. Quinlan, N. Henderson, R. H. Middleton, S. P. Nicklin, R. Fisher, F. Knorn, S. K. Chalup, and R. King, “The 2006 NUbots team report,” The University of Newcastle, Australia, Tech. Rep., 2006.
12. C. McMillen and M. Veloso, “Distributed, play-based role assignment for robot teams in dynamic environments,” in *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems (DARS 2006)*, 2006.
13. C. Guestrin, M. G. Lagoudakis, and R. Parr, “Coordinated reinforcement learning,” in *Proceedings of the International Conference on Machine Learning*, 2002, pp. 227–234.
14. M. G. Lagoudakis and R. Parr, “Learning in team markov games using factored value functions,” in *Proceedings of NIPS*2002: Neural Information Processing Systems: Natural and Synthetic*, 2002, pp. 1659–1666.