Team coordination among robotic soccer players

Matthijs T. J. Spaan and Frans C. A. Groen

Intelligent Autonomous Systems group, Informatics Institute,
Faculty of Science, University of Amsterdam, the Netherlands
{mtjspaan,groen}@science.uva.nl

Abstract. We present an approach for coordinating a team of soccer playing robots, used by Clockwork Orange in the RoboCup middle-size league. It is based on the idea of dynamically distributing roles among the team members and adds the notion of a global team strategy (attack, defend and intercept). Utility functions are used for estimating how well suited a robot is for a certain role. They are not only based on the time the robot expects to need to reach the ball but also on the robot's position in the field. Empirical results from the RoboCup 2001 tournament are presented demonstrating the value of extending role distribution with a team strategy.

1 Introduction

An interesting research topic in RoboCup is the problem of coordinating a team of agents, whether it be software agents or real world agents. Middle-size league teams traditionally focus on issues like robot control, real-time image processing, self localization and fusion of sensor data but in the last years actively coordinating the multi-agent system has come into reach. We present an approach for coordinating a team of middle-size robots which uses a global team strategy. This approach was used in Clockwork Orange, the Dutch RoboSoccer Team [5] at the RoboCup 2001 tournament. The team is a collaboration effort of the Delft University of Technology, the Utrecht University and the University of Amsterdam.

First we will shortly discuss related approaches used by some other teams. Next we describe our role-based team coordination mechanism and how it influences action selection. We finish with results obtained at RoboCup 2001 and draw some conclusions.

2 Related work

We view RoboCup middle-size league as an application in which intentional cooperation is the preferred choice for a cooperation model as opposed to the swarm model, a distinction made by Parker [3]. Swarm cooperation focuses on the emergent cooperative behavior in large groups of robots, each of which individually has limited capabilities. When robots achieve purposeful cooperation built on their individual higher level capabilities the cooperation is called intentional.

We opt for intentional cooperation since in the RoboCup domain several tasks such as defending and attacking are required, the number of robots is very limited, timing is crucial for success and the robots can be heterogeneous in their hardware and capabilities. Other teams consider a deliberative approach to the team strategy acquisition useless in such a dynamic and hostile environment [7].

A common approach for intentional cooperation is a system based on the distribution of roles among the field players. The players of CS Freiburg distribute roles amongst themselves, namely an active, support and strategic role [9]. Each robot determines its utility to pursue a certain role and informs its teammates. Based on these utilities a robot chooses his role. Roles are also distributed among the players of ART [1]. The roles they define are a main attacker which demands ball possession, a supporting attacker and a defender. The protocol for distribution of the roles among the players is based on utility functions.

Dynamic role assignment based on utility functions seems a flexible way for achieving cooperation, but most of the standard role distributing schemes as designed by other middle-size league teams seem to employ just one team strategy: *attack*. However if one defines ball possession as a prerequisite for being able to attack one can see that in a typical middle-size league only a limited amount of time is spent attacking. It usually takes a lot of time to find and obtain the ball. Therefore we have extended this model with two more team strategies: *defend* and *intercept*. The team strategy determines what roles have to be assigned among the team members. It extends formations as defined in [6] by assigning priorities to the roles to cope with a variable number of participating agents.

3 Team coordination

We have divided the soccer game in three states: either your team has the ball and attacks, the other team has the ball and your team defends, or nobody has ball possession and your team tries to obtain it. For simplicity we assume the game is characterized by ball possession alone, knowledge regarding the status of the ball is provided by our shared world model [2]. A problem of such a finite state machine could be that it is prone to oscillations between the team states. To prevent such an unfortunate situation the protocol for deciding on team strategy and distributing roles "locks" the team strategy during a cycle of the coordination process.

We define a team strategy as a distribution of certain roles over the available field players, where a role is defined as a mapping of situations to individual robot actions. The team strategy poses limitations on the possible mapping of roles to robots. Note that only three roles are associated with each team strategy as the goalkeeper does not actively participate in team play: it has its fixed role of goalkeeper and has fundamentally different hardware.

We have assigned priorities to each role in a certain team strategy, since during a game not all three field players may be in play but some roles are more important than others. Table 1 shows the roles for each team strategy, with #1

Table 1. Distribution of roles associated with each team strategy.

Team strategy	Role #1	Role #2	Role #3
Attack	AttackWithBall	PassiveDefend	AttackWithoutBall
Defend	ActiveDefend	PassiveDefend	ActiveDefend
Intercept	InterceptBall	PassiveDefend	InterceptBall

being the most and #3 being the least important role. If there is one field player available only role #1 will be assigned, with two field players active roles #1 and #2 will assigned etc. Role #1 is always a ball oriented role: with only one field player ready for duty you want it to go after the ball. This is also the case when communication fails, the robot will assume it is the only player in the team and fulfill role#1.

A robot in role PassiveDefend (PD) waits in front of its own goal for the opponent team to attack. Role InterceptBall (IB) chases the ball trying to obtain control over it. Role ActiveDefend (AD) is a defensive variant of the previous role. When a robot controls the ball it assumes role AttackWithBall (AWB), which usually means dribbling with it toward the enemy goal followed by a shot. Role AttackWithoutBall (AWoB) describes an auxiliary attacker moving toward the enemy goal together with the main attacker.

The role distribution approaches already in use at RoboCup usually have three roles: an attacker, a supporting attacker and a defender. These can be respectively identified with AttackWithBall, AttackWithoutBall and PassiveDefend. Adding team strategies suggests we also have to differentiate the existing roles. The added roles InterceptBall and ActiveDefend seem necessary in our view, but one could easily argue a different set of roles is necessary and sufficient to play a decent game of robot soccer.

Utility functions

To decide which robot should be assigned which role a mechanism based on utility functions is used. Each role is associated with a utility function which tries to measure how well suited a robot is for this role in the current state of the world. For an attacking role it is easy to define a useful utility function. This is more complicated for a defending role, since an attacker only has to focus on the ball while a defender should assume a good position waiting for things to come.

We propose to compose the utility functions for each role on two measures: first one is the time a robot expects it needs to reach the ball. This time is based on the shortest angle the robot has to turn to face the ball and the Euclidean distance between them. This is important for roles that want to chase after the ball: ActiveDefend, InterceptBall and AttackWithBall. For the last role we also take into account whether or not a robot controls the ball.

Second measure is how well the position of a robot is suited for the role. We added an evaluation mechanism for this purpose based on attracting and repelling areas on the field [4]. This measure is needed for the non ball oriented roles PassiveDefend and AttackWithoutBall, but it is also relevant for the ball oriented roles to fall back upon when the position of the ball is unknown.

Assignment of roles

Clockwork Orange's shared world model [2] simplifies the task of assigning roles to robots. When a robot notices a change in ball possession it alerts its teammates by telling them the new team strategy. The robot continues by calculating its own utility for each of the roles in the new team strategy (see table 1) and transmits this information to its teammates. Upon receiving the new team strategy they do the same. While waiting for a certain time on the utility scores of teammates each robot calculates these scores based on its own world model. Utility scores coming in from a teammate have preference however, which introduces a level of redundancy.

After receiving utility scores of each teammate or when a timeout occurred, each robot selects its role by comparing the utility scores each team member has for each role available, starting with the most important role. We try to limit the impact of oscillations between team strategies by letting the robot ignore all incoming new team strategies while still busy waiting for utility scores from teammates.

Impact of role on action selection

Assigning roles to robots only makes sense if the robot takes its role (and possibly those of its teammates) into account when selecting the next action it should take. It should execute the next action which benefits the team the most, and its role provides the robot with a description of what the team expects of it. Without going into too much detail we will shortly describe the use of roles in our action planning [4].

Action planning is modeled using Markov decision processes, similar to Tambe's approach used in the simulation league[8]). A robot's role determines its action space and influences its reward function. In order to be able to find a good solution to the Markov decision problem we discretize the action space (instead of the state space which remains continuous), which means a robot considers only a finite set of actions at a time. Actions are defined as having a certain type like move or dribble and certain parameters like target positions, whose number is potentially infinite. As a solution to this problem we only consider a finite number of target positions. The size of the set of actions lies in the order of magnitude of 50. The role of a robot determines the contents of this set: a defensive role will lead to more move actions than shoot or dribble actions while role AttackWithBall for instance will contribute more shoot and dribble actions than plain move actions.

Our reward function is designed as follows: estimate the desirability of the current world state by looking at several soccer heuristics, simulate the action on this world to obtain a new world state, estimate the desirability of this new world

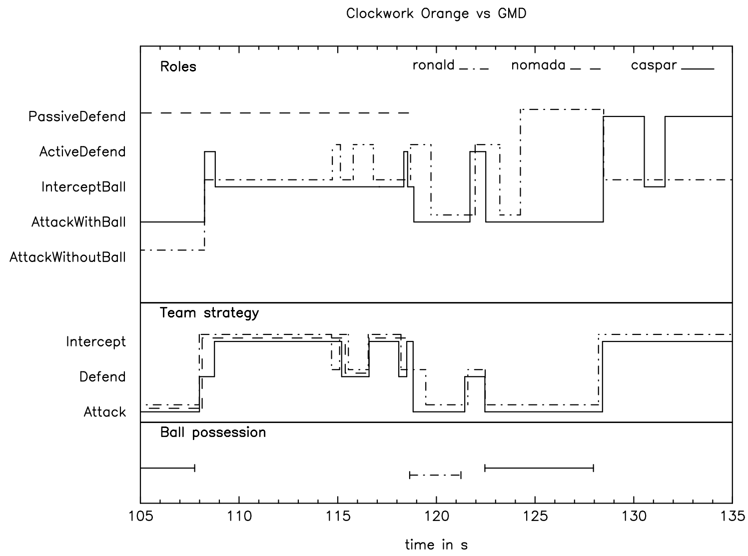

Fig. 1. Visualization of team strategy, role distribution and ball possession during a 30 second period of the RoboCup 2001 game against GMD. The x-axis displays the wall clock time in seconds since the start of the game. The robots involved are Ronald, Caspar and Nomada (removed from the game 118.88 seconds after game start). Their role and team strategy are shown on the y-axis. When one of them has ball possession a line is drawn in the bottom part of the graph.

state, and the difference between the two estimates is the reward. The soccer heuristics are about whether the ball is in one of the goals, ball possession, strategic positioning and how well the location of the robot is suited for its role. So to what extent an action is considered beneficiary to the team partially depends on the robot's role.

4 Results

Clockwork Orange successfully participated in the RoboCup 2001 tournament in Seattle, USA reaching the quarterfinals. In total seven games were played, resulting in three victories, one draw and three losses. At the German Open 2002 tournament Clockwork Orange became fourth. Results regarding team coordination will be presented from the last four and a half[1] last games from RoboCup 2001. We will start with the outcomes of these matches: Clockwork Orange vs. Trackies 0-8, Fun2maS 5-0, Artisti Veneti 3-0, GMD 1-4, CS Freiburg 0-4.

Next we will discuss a 30 second fragment from the match against GMD, shown in figure 1. This fragment gives an example of how our coordination

[1] During the half-time interval of the third game the team coordination implementation was frozen for the rest of the tournament.

Table 2. The portion of time the team spent in each team strategy. The surface area of the circles is proportional to the amount of time spent in each team strategy or role relative to the total time. Right columns show these total times in seconds.

	Attack	Defend	Intercept	Total
Trackies	•	•	●	1170.30
Fun2maS	●	·	•	2539.04
Artisti Veneti	•	•	●	2811.09
GMD	•	•	●	2588.36
CS Freiburg	•	•	●	3147.06

Table 3. The portion of time each robot spent in each role.

	PD	AD	IB	AWB	AWoB	Total
Trackies	•	•	●	·	·	1050.45
Fun2maS	•	·	•	•	·	2256.67
Artisti Veneti	•	•	●	·	·	2638.13
GMD	•	•	●	•	·	2467.76
CS Freiburg	•	•	●	•	·	2843.96

mechanism works under good conditions: the world model is consistent at an acceptable degree. The robots in our team are called Caspar, Ronald and Nomada.

At the start of the fragment Caspar has control over the ball and each robot believes the team strategy is *attack*, as they should. The team has correctly assigned the AttackWithBall role to Caspar and the AttackWithoutBall role to Ronald while Nomada is standing close to the own goal area in role PassiveDefend. Then Caspar loses the ball, the team switches to team strategy *intercept* and the two attacking robots switch to role InterceptBall. Between $t = 115$ and $t = 118$ there is some confusion whether or not the opponent controls the ball which leads to oscillations between team strategies *intercept* and *defend*.

At about the same time as Ronald gains possession of the ball Nomada is neatly shut down and removed from the game. This means the team has lost its defender and one of the other two should fill the gap. Ronald does so by switching to role PassiveDefend at $t = 124$, after having been in error for a short while: it was at the same time as Caspar in role AttackWithBall although it was really Caspar which controlled the ball.

Table 2 visualizes the distribution of team strategies for the games listed above. It shows team strategy *intercept* is the most common in all but one match.

Table 4. The portion of time spent executing an action of a certain type while in a certain role. Data is from all field player robots during the four and a half last games of RoboCup 2001.

	Turn	Move	Dribble	Shoot	Seek	Chase	Total
PD	•	•	·		•	•	3529.41
AD	•	•	·	·	·	●	1325.78
IB	•	•	·		•	•	4238.73
AWB	•	•	•	·	•	•	1812.27
AWoB	•	●	·		·	•	326.89

The exception is the game against Fun2maS in which *attack* is dominant. This seems consistent with the outcome of the match.

The distribution of roles over robots can be found in table 3. It shows that the AttackWithoutBall role has been assigned only a small amount of time. It is the least important role of team strategy *attack* which means it will only be used if there are three field players active, each accurately knows its position and one of them has ball possession.

To see the influence of a robot's role on its action selection we have included table 4. It shows what type of actions have been chosen while the robot was in a certain role. The data is a summary of all robots during the last four and a half games of RoboCup 2001 which lasts just over three hours. The table demonstrates that a robot's role influences the kind of actions it takes. Chase actions, whose purpose it is to obtain the ball, are not used as much in roles PassiveDefend and AttackWithoutBall as in the other roles, which one would expect given their nature.

5 Discussion

Figure 1 shows the good performance of our team coordination mechanism under good conditions, but even then inconsistencies sometimes occur. For instance at $t = 131$ Caspar briefly switches to role InterceptBall when it shouldn't. The reason is that Caspar's world model has not received a position update from Ronald's world model for a while, as Ronald has no estimate of its own position that is accurate enough to communicate. From Caspar's point of view it is the only active member of the team and it should thus fulfill the most important role in team strategy *intercept*: InterceptBall. These kind of problems are common and should be properly dealt with.

As table 2 shows the total amount of time spent in team strategy *attack* is rather low, only 28% of time (almost three and a half hours). This would seem to confirm our intuition that extending role-based approaches with a global team strategy makes sense, although we do not have the resources (two complete teams) to confirm this. In RoboCup simulation league however these lines of reasoning are common.

The team strategy sub-tables also show that most of the time (53%) is spent in *intercept*, resulting from the fact that it is hard to detect whether the other team controls the ball. This would call for a more sophisticated way of determining the applicable team strategy than just looking at ball possession alone. Ball possession is a good first indication but other concepts like ball position on the field or the relative distances of teammates and opponents can be added. This would also reduce the number of oscillations caused by the possibly rapid changes in ball possession.

Acknowledgements

We would like to express our gratitude to Nikos Vlassis for his support. This research is supported by PROGRESS, the embedded systems research program of the Dutch organisation for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the Technology Foundation STW.

References

1. C. Castelpietra, L. Iocchi, M. Piaggio, A. Scalzo, and A. Sgorbissa. Communication and coordination among heterogeneous mid-size players: ART99. In P. Stone, T. Balch, and G. Kraetzschmar, editors, *RoboCup 2000: Robot Soccer World Cup IV*. Springer-Verlag, 2001.
2. F.C.A. Groen, J. Roodhart, M. Spaan, R. Donkervoort, and N. Vlassis. A distributed world model for robot soccer that supports the development of team skills. In *Proceedings of the 13th Belgian-Dutch Conference on Artificial Intelligence (BNAIC'01)*, Amsterdam, 2001.
3. L. E. Parker. *Heterogeneous Multi-Robot Cooperation*. PhD thesis, Massachusetts Institute of Technology, 1994.
4. M. T. J. Spaan. Team play among soccer robots. Master's thesis, University of Amsterdam, 2002. `http://www.science.uva.nl/research/ias`.
5. M. T. J. Spaan, M. Wiering, R. Bartelds, R. Donkervoort, P. Jonker, and F. Groen. Clockwork Orange: The Dutch RoboSoccer Team. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup 2001*. Springer-Verlag, to appear.
6. P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110, 1999.
7. Y. Takahashi, S. Ikenoue, S. Inui, K. Hikita, and M. Asada. Osaka University "Trackies 2001". In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup 2001*. Springer-Verlag, to appear.
8. M. Tambe and W. Zhang. Towards flexible teamwork in persistent teams: Extended report. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(2):159–183, 2000.
9. Th. Weigel, W. Auerbach, M. Dietl, B. Dümler, J. Gutmann, K. Marko, K. Müller, B. Nebel, B. Szerbakowski, and M. Thiel. CS Freiburg: Doing the right thing in a group. In P. Stone, T. Balch, and G. Kraetzschmar, editors, *RoboCup 2000: Robot Soccer World Cup IV*. Springer-Verlag, 2001.