Coordinating Reactive Planning for Football Playing Agents

Erik Carling

Examensarbete för 20 p, Institutionen för datavetenskap,
Naturvetenskapliga fakulteten, Lunds universitet

Thesis for a diploma in computer science, 20 credit points,
Department of Computer Science, Faculty of Science, Lund University

Coordinating Reactive Planning for Football Playing Agents

Abstract

In order to play football efficiently the actions of a team need to be united. This work pursues an investigation of cooperation among agents. The construction of two cooperation models originating from two different viewpoints is described. They are implemented and integrated in the planning engine of the Team Sweden RoboCup team. The main challenges that collaboration on the football field faces is adjusting to the dynamic environment and acquiring the ability to take advantage of what is known about the teammates of a robot. This report contains results of a comparison of the two models of cooperation.

Koordinerad reaktiv planering av fotbollsspelande agenter

Sammanfattning

För att spela fotboll på ett effektivt sätt behöver lagspelarnas agerande sammanfogas till en helhet. Med denna utgångspunkt så eftersträvar detta arbete en vettig hållning till samarbete bland agenter. Konstruerandet av två modeller för samarbete som grundar sig på olika förhållningssätt till samverkan beskrivs. De implementeras och integreras i Team Swedens planerings motor för RoboCup miljön. Samarbetande robotar stöter på en mängd utmaningar på fotbollsplanen. De måste kunna anpassa sig till den dynamiska miljön och ta vara på den kunskap de har om sina lagkamrater. Denna rapport innehåller en jämförelse av de två modellerna för samarbete.

Preface

First of all I would like to thank my supervisor Jacek Malec for valuable guidance throughout the project. I also want to thank Daniel Påhlstorp and Dannie Ronnovius for providing me with the simulator using which I have tested my models. Finally, I am especially grateful to Charlotte Carling for helping me with the English language.

Erik Carling
Lund, 2003-08-18

Table of Contents

1 Introduction

1.1 RoboCup

The international competition of RoboCup [1] has been held annually since 1997. Football has been chosen as a dynamic game to make advances in AI, robotics and other related fields. Teams from all around the globe compete in a range of leagues with physical robots and in simulators. In order to perform well the participants need to master areas such as team coordination, real-time planning, sensor-fusion, autonomous agent design and robotics. The advances made are hoped to aid development in socially more significant fields and industries. The project itself also has the following ultimate goal.
"By 2050, develop a team of fully autonomous humanoid robots that can win against the human world champion team in soccer."

1.2 Formulating the Problem

Team Sweden [2] is a joint effort by Örebro University, Blekinge Institute of Technology, Lund University, Umeå University and Murcia University of Spain to participate in RoboCup. It has entered the contest since 1999 and over the years an effective method for reactive planning has been established. So far only sparse attempts have been made to coordinate planning within the team. As it is their view, as well as mine, that such attempts could hold great benefits, it was suggested as the starting point for this investigation. To unite the actions of robots in a rapidly changing dynamic environment puts demands on time consumption, strategy modeling and fault tolerance on a cooperation model.

The challenge is thus to find a way to cooperate that will benefit the overall performance of the team and that at the same time is sensitive to the characteristics of the domain. A model for this cooperation is then to be constructed, which incorporates the planning structure of Team Sweden.

1.3 The Report

This report aims to describe two separate models for united actions within a dynamic multi-agent environment and the assignment is introduced in Chapter 1. The concept of Multi-Agent Systems is then presented in Chapter 2 and the software architecture of Team Sweden is described in Chapter 3. In order to make the reader familiar with different theoretical approaches to team work a reasonable view on cooperation is discussed in Chapter 4. Furthermore the report seeks to show how these theories are translated into models, which are integrated into the Team Sweden architecture. A model based on the anticipation of the teammates' actions is introduced in Chapter 5. Another model based on the dynamic distribution of roles is described in Chapter 6. Test results and the evaluation of the utility of the two models are presented in Chapter 7 and finally the report is concluded in Chapter 8.

2 Background

2.1 Multi-Agent Systems

In the field of computer science the term agent refers to a program, or piece of program, that functions as an active entity in a computerized environment together with other processes or programs. The term agent often entails an autonomous quality for which the following definition has been suggested by Stan Franklin and Art Graesser [3].

*"An **autonomous agent** is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."*

I will incorporate the quality of making individual decisions based on perception of the environment, whenever I refer to agents in this report.

Multi-Agent Systems (MAS) consist of a defined space, either physical or computer generated, containing a number of objects. These objects can either be agents or some form of resource. A resource can be available or occupied by an agent and therefore a position might also be considered as such. The coexisting agents' use of resources may not overlap and their goals may be fully independent of each other. If this is the case the domain presents no real reason for interaction between agents. In systems with shared resources and mobile agents with agendas that affect each other encounters are likely to occur. Coordination of actions in such environments often benefits all the agents involved.

Wang, Wang, Wang and Soh [4] have described the domain of RoboCup Football (RCF) as Real-time, Cooperative and Adversarial (RCA). As such it requires agents to make real-time individual decisions, as the football field is a rapidly changing environment. It also promotes cooperation of agents within a team as they share the same goal. Finally the rivalry of two competing teams leads to the perception of the opponent as an agent with an agenda, which benefits from obstructing the own team's goal.

2.2 Sony Legged Robot League (SLRL)

The SLRL is one of the leagues of RCF in which the teams consist of four identical robots each. Since all teams use the same kind of robots the league is especially interesting from a software engineering viewpoint. The games are played on a field that is 460 cm long and 310 cm wide and the robots of the two teams are marked differently so they can be recognized. The goalie is marked separately and is the only player who is allowed to stay within the penalty area of its own team. On each side of the half way line and in every corner landmarks are placed to help the agents orient themselves. A match is played in two halves of ten minutes each and a limited-bandwidth radio communication is allowed between teammates.

Fig. 1. Sony's AIBO ERS-210a.

2.3 Sony's AIBO

The hardware used in the SLRL is the Sony AIBO ERS-210 model robot [5] seen in Figure 1. The dog-like robot is about 30 cm long and 27 cm tall. It has three degrees of freedom in each leg and 20 degrees of freedom in all. A color camera is located in its nose and microphones on the sides of its head. There is also a speaker in its mouth and a number of other various features. An API called OPEN-R has been developed by Sony to operate the different functions of the robot.

3 The Team Sweden Platform

The aim of this chapter is to make you familiar with the software structure and reactive planning of Team Sweden.

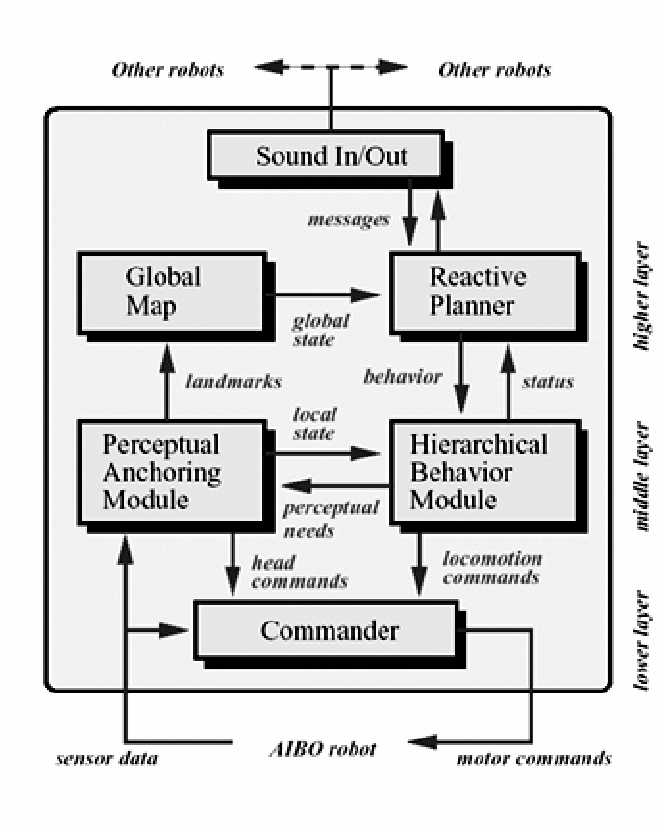

Fig. 2. Software architecture used by Team Sweden.

3.1 The Architecture

The agents of Team Sweden use the software structure shown in Figure 2 above. This layered architecture is based on the Thinking Cap model [6] developed at Örebro University. Three levels of control, from physical to theoretical are implemented. The lower level contains the commander module (CMD), which handles the motion of the robot, e.g. turning its head to let the camera register the environment and moving its feet to achieve an appropriate style of walking.

In the middle layer the perceptual anchoring module (PAM) holds a local map portraying the robot's view of the world. Objects are represented in agent-oriented coordinates and associated with a value called an anchor. It is a value on a [0, 1] scale, stating how recent, and therefore reliable, the information about the object is. Whenever the camera detects an object or when the robot moves, the map is updated and the anchor value set. Fuzzy logic is a branch of logics with truth-values between 0 and 1, which is used here to deal with knowledge representation. It is a suitable tool to manipulate knowledge about the environment in a dynamic domain.

IF (BallOnLeft and not(BallHere))	TURN(Left)
IF (BallOnRight and not(BallHere))	TURN(Right)
IF (BallAhead or BallHere)	TURN(Ahead)
IF (not(BallHere))	GO(Fast)
IF (BallHere)	GO(Stay)
ALWAYS	SIDE(None)

Fig. 3. Translation of the GoToBall behavior.

A set of high-level behaviors is implemented in the Hierarchical Behavior Module (HBM), also located in the middle layer. Complex high level behaviors are organized in a hierarchical rule structure that translates them to commands that can be executed by the CMD. Using information from the PAM together with rules based on fuzzy logic, high level behaviors are translated into low level ones. Figure 3 illustrates such a translation for the GoToBall behavior. E.g. the BallAhed fuzzy predicate has the value 1 if the ball is located straight in front of the agent and declines to zero at 30 degrees angle to the left or right.

The upper layer holds the Global Map (GM), which represents all that, is known about the current state of the game. It is updated by the PAM converting information about the objects to field-oriented coordinates. Here as well the objects are associated with a fuzzy anchored value stating their reliability. Teammate communication also helps to revise the GM trough map fusion. The Reactive Planner (RP), which is the subject of the next chapter, decides what behavior to use based on information from the GM.

3.2 The Electric Field Approach

The RP module is the planning engine of the agent structure. Its purpose is to choose which behavior the agent is to execute, based on information from the GM. This is done by simulating a number of available behaviors and evaluating the global states they result in. Adding positive and negative charges to strategically important locations creates an artificial potential field. By probing key positions in the field a heuristic potential can be assigned to a specific state. Values for all the charges proportional to their distance to the probed position are added together. The behavior resulting in the state with the highest potential measured is the one chosen. This procedure is known as the Electric Field Approach (EFA) [7].

There are two kinds of charges. Static charges, which stay the same throughout the game, make up the underlying basic strategy for attacking the global task of playing football. This task can crudely be formulated as, getting the ball into the opponents' net while keeping it out of your own. Thus positive charges are placed along the opponents' goal line and negative ones along that of the own team, illustrated in Figure 4. The formulation also suggests probing the position of the ball, as it needs to be manipulated in order to complete the task. This rough base of the EFA, evaluates states simply on the basis of closeness of the ball to the opponents' net.

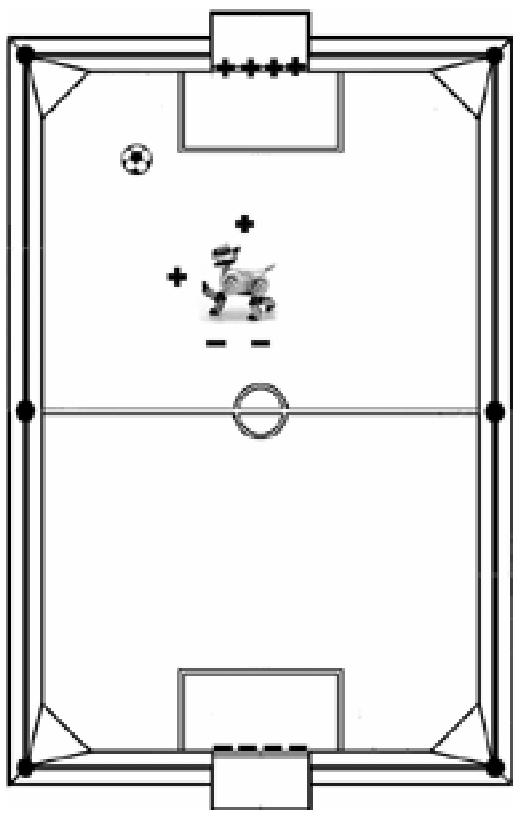

Fig. 4. Illustrating the static charges of the nets and the dynamic charges around the robot.

The way the other players are placed in the field should also be considered when grading a state. This increased sensitivity is obtained by adding dynamic charges to the basic potential field when presented with a new state. Opponents are represented by a single negative charge to avoid getting the ball near them. Teammates on the other hand are represented by a positive charge to encourage passes. This charge is placed in front of the robot, on the side of the opponent net, where a pass is preferably received. The potential of these charges is set proportional to the anchoring value of the respective agent to reduce the potential influence of perceptive faults.

To keep the planning agent on the desired side of the ball at all times, additional charges are set around the robot, as illustrated in Figure 4. Negative charges are placed behind the robot, on the side of the own net while a positive charge is placed on the other side of the robot. A head charge is placed in front of the robot, as this is a desirable place to have the ball. Together with the static charges this makes up the strategy for evaluating heuristic potential.

At any given time in a match a subset of the behaviors implemented in the HBM is available to the agent and make up part of the strategy of the team. It is possible to change the composition of this set dynamically during a game. By evaluating a few parameters, e.g. the position of the ball, it can be determined whether the situation has changed. A new strategy could then be chosen which could also include an additional probing position. To induce an agent to move to a passable position a strategic probe can be used, e.g. by adding an additional probe at some strategic position in the field where an agent is likely to receive a pass. If an agent knows that a teammate is in charge of the ball such a probe can be used to produce a supportive behavior. Ball possession can be communicated via the radio link.

The evaluation loop determines what behavior to use next and is triggered by any of the following events:

- The current behavior is about to finish.
- The current behavior is in trouble.
- The situation of the game has changed significantly since the last evaluation.

The behaviors of the current strategy are graded one by one. If the preconditions of a behavior are met it is simulated by moving the concerned objects on a copy of the GM accordingly. Dynamic charges are set in the resulting state as described above. The position of the ball is probed adding a value for each of these charges as well as the static charges. This value is proportional to the distance of the charge. If the strategic probe is used for the behavior evaluated, the value of the charges at its position is added to those of the ball probe. The behavior with the highest potential is the one chosen, translated by the HBM and executed by the CMD.

4 Constructing Team Play in RCF

In this chapter I will present various theoretical approaches to collaboration, discussing which is best suited for the RCF domain.

4.1 The Nature of Cooperation in RoboCup

Collaboration is achieved through the coordination of actions. It aims to benefit the respective agendas of the agents involved. The coordination effort can be made either before, after or be incorporated in the individual planning, thereby defining the nature of cooperation.

The RCA aspect of the RoboCup domain, introduced in Chapter 2, establishes the concept of teams. Within a team agents share a common goal and are consequently aware of each other's agendas. As this goal also includes the obstruction of the goal of the opponents and vice versa they are also partially aware of their agenda. These aspects make predictions about the actions of other agents possible to a certain extent. Collaboration in the football domain could benefit from the exploitation of such predictions. Furthermore real-time planning requires the cooperation models to be robust in order to deal with the rapid changes of the domain.

As the goalie has separate rules in the SLRL it will not be included in any collaboration schemes discussed in this report.

4.2 Negotiation

The most common way to combine efforts in MAS is probably through a negotiation process. Sascha Ossowski has given a suggestion to the outlines of a general model for such a process [8]. A plan is formed based on how the current state of the environment, as perceived by the agent, differs from the goal state it whishes to attain. The actions available, coinciding with this plan, are evaluated and ranked by some fitness function. Through communications with the other agents of the domain a bargaining procedure is initiated. A set of logical relations and rules is used to decide which of the behaviors should be executed together, based on what resources they occupy. The actions are fused to the best distribution with regards to both the individual and the group.

To cooperate through negotiation can be favorable in a couple of situations. Firstly when agents share resources that should not be occupied at the same time this is easily expressed in the logic rules. The bargaining procedure then makes sure that a conflict is avoided. In a RCF team the ball could be seen as such a resource. Behaviors involving the ball would in this way only be chosen by one teammate at the time. Apart from the ball, however, there are no other obvious joint resources in a football game. A bargaining rule system that has to keep track of the relations between all possible behaviors would have to be rather complex.

Secondly, heterogeneous agents with different software structure can collaborate, as long as they share a communicative protocol for the fusing of plans. One of the main concepts of negotiation is that individual planning is separate from, and thereby independent of, coordination. This advantage is lost in RCF with homogenous agents that are familiar with the software structure of their teammates. It also means that none of the specific RCA aspects of cooperation are considered in this approach.

4.3 Global Planning from a Local Perspective

The Tsinghua University of China has presented a cooperation model [9] rooted in the way a team's utility is measured for a certain state. To assign a specific value to any global state is difficult for an agent, since RCF is a partially observable domain. Their solution is based on modeling global utility as "the integration of influences each individual behavior contributes to the environment". Maximization of the collective utility is achieved by maximizing the sum of utilities gained trough individual actions. The compound utility P(AB) of action A with utility P(A) and action B with utility P(B) can be formulated $P(AB) = P(A) + P(B) + I(A, B)$. I(A, B) is the negative or positive influence of putting A and B together. This influence is 0 if they are independent of each other and $-\infty$ if A and B can not be executed together.

The planning process can be described in the following steps:

- Subtask generation: the subtasks that the current situation requires are generated.
- Subtask-Executor pair generation: the subtasks are linked to agents able to carry them out.
- Arrangement evaluation: calculation of the individual utility gained when a given agent performs a subtask.
- Generation of the execution set: the selection of arrangements with the highest global utility is calculated.
- Task assignment: the agent picks out its own task and executes it.

This procedure looks similar to central planning but every agent carries it out individually. In a dynamic environment team members are bound to have different opinions of what the global state looks like. Individual evaluation can therefore lead to different distributions of subtasks for different agents at any given time. A crucial subtask thus runs the risk of not being executed by any agent. The planning process, suggested above, assumes that perception of the global state is sufficiently accurate and equal among team members for consistent distributions to be made.

Shared knowledge about the actions of teammates, called privities, is used to minimize the risk of misunderstanding. Agents within a team are familiar with the behavior pattern of all the players and share some public rules to attain these patterns in certain situations. A shared planning structure for all the agents in a team makes these assumptions realistic. Mutual beliefs about the environment are also needed to reach privities. To know a teammate's view of the current state of the game one needs to be familiar with that agent's internal state. This can either be mediated trough communications or obtained by behavior recognition. In a rapidly changing domain, such as RCF, these methods might prove to be too time consuming. The solution is to simply regard the local state as a global one and assume that the teammates share this view of the environment.

There is a built-in fault tolerance in anticipating the actions of teammates to maximize team utility. If one agent has a faulty view of the global state the rest of the team still functions normally as planning is done individually. In this way the model also makes use of the specific RCA qualities of the domain, as anticipation is used to predict future actions of teammates.

4.4 Role Assignment

The most common way to coordinate a human football team is to assign different player types. The current formation used specifies an area in which a certain player type is desired to operate in any given situation, thus forming the spatial tactics of a team. Players are likely to be inclined to play certain types due to physical differences. The fast runner with the precise left foot will, e.g., play leftfielder advancing on the side to pass the ball to the forward. In a team of autonomous agents there are no such physical differences. Attacking the global task with a formation of player types, each addressing some subtask, still proves an effective way to unite their actions.

"*A role, r, consists of a specification of an agent's internal and external behaviors.*" This meaning of the term has been suggested by Peter Stone and Manuela Veloso [10]. Thus a role becomes the arguments to, and conditions for, the function that translates the agent's inner state into a behavior. An executing agent assigned a role, in the RCF domain, consequently adopts a certain player type.

Distribution of the roles determines the flexibility of the teamwork effort. Keeping the same player types static throughout the game would limit real-time coordination efforts and thus the computational power used. With the fast changes of the environment and limited amount of players this would cause agents to have to move unreasonable distances to comply with new states.

For every role a reference position can be associated, at every global state, from which it is ideally executed. For a homogenous team, roles can be dynamically assigned to minimize the team's distance to these positions at any time. This way a team can respond to a new state a lot faster. Stone and Veloso introduce such a system with flexible roles and protocols for assigning them. Information shared through communication determines which role is the best suited for each teammate.

4.5 Coordinating Planning Within Team Sweden.

Team Sweden has already explored some methods for cooperation. It is possible for an agent to claim possession of the ball using the radio link. An agent receiving such a claim from a teammate will change its current strategy. Using a strategic probe, described in 3.2, it can assume a supportive role by moving to a passable position.

Fusing methods for sharing local information through communication has been developed for the Team Sweden architecture by Påhlstorp and Ronnovius [11]. The team's shared view of the global state of the game has greatly improved thanks to their work. This joint map construction provides a favorable starting point for coordination as teammates can be assumed to have a similar perception of the global state at any given moment. The possibility arises to predict the actions of team members not only due to a shared goal but also a shared view of the global state.

5 Anticipation Model (AM)

In this chapter I will present a model for cooperation, which incorporates coordination in the EFA planning by anticipating the actions of the teammates to maximize the compound utility of the team.

5.1 The Foundation

The basis for the AM lies in the forging of two hypotheses. Firstly the GMs of the agents on the team are consistent enough to estimate the actions of teammates as well as one's own. The fusing of information through communication makes this a reasonable assumption. Secondly the global utility of a team can be viewed as the collected potential of its agents. Therefore the maximization of the sum of individual potentials also maximizes the team's utility as a whole. This viewpoint is derived from the Tsinghua University RoboCup effort described in 4.3.

The calculation of the potential of a state in the EFA, as discussed in chapter 3.2, is based on the adding of the influence of a number of charges to a probed area. This additive nature makes it reasonable to estimate the compound global utility of the team as an extension of this approach by adding the sum of individual potentials. As shown in 4.5, it is reasonable to assume a certain degree of consistency in the GMs of team members. The position of the teammates is assumed to be known to an agent at any given time. Anticipating that their internal state is equal to the own, planning for the whole team is possible. In this way the best distribution of actions can be obtained and the behavior maximizing the global utility chosen.

5.2 Adjusting to the EFA

Trying to apply the Tsinghua approach in EFA planning the first step is subtask generation. A number of behaviors, of those implemented in the HBM, are chosen to address the global task of playing football at a certain state of the game. This set makes up part of the current strategy of the team, and can be referred to as the "starting set". The EFA supports dynamic switching between different starting sets as the situation of the game changes. But for simplicity I will consider a single static starting set here as this feature does not have any direct influence on cooperation in the AM.

The complete homogeneity of the team makes all agents potential candidates for all the behaviors. To adjust smoothly to new situations in the field it is preferable that the agent best positioned for a behavior is the one to execute it. As the EFA evaluates the situation after simulating a behavior no consideration is given to the prior state. E.g. a GoToBall behavior would therefore receive the same potential simulated by an agent two meters from the ball as one only one meter away. Agents close to the ball are actually disadvantaged as other agents benefit from the positive charge at their position when GoToBall is simulated. Some form of spatial consideration needs to be given when evaluating behaviors that involve moving to a position. When calculating their potential I have chosen to simply subtract a smaller value proportional to the distance to this position. Other behaviors like e.g. KickForward need no such extra potential, as it requires the agents to already be located at a certain position in the field.

In order to compose a distribution of behaviors from the starting set, to meet a given state, rules about how they can be put together are needed, as shown in 4.3. As there are only three field players in a team in the SLRL domain, some simplifying assumptions are feasible to eliminate such a rule system:

1. It never benefits the team to have two or more agents executing the same behavior at the same time.

2. The benefit of executing a certain behavior is proportional to the potential its resulting state is given by the EFA. No additional benefit is given by executing it together with another specific behavior.

These assumptions reduce the generation of the execution set to finding the best permutation of three candidates from the starting set. The behavior to execute is chosen from the permutation with the highest combined potential.

5.3 The Behaviors to Consider

The composition of the starting set is a vital part of the coordination effort of the anticipation model. It has to contain behaviors that, executed together, satisfy the global task at the current state of the game. As I have chosen to consider the same starting set throughout the game it must be able to satisfy every possible state. The behaviors I have included in the starting set are:

GoToBall:	Move to the ball
KickForward:	Kick the ball in the current facing direction
FaceBall:	Turn to ball
PushBall:	Lightly push the ball forward and follow it
AlignBallAndNet1:	Face the ball and the opponent goal
GoBetweenBallNet2:	Go to a position between the own goal and the ball
OpenBehavior:	Go to a passable position

The preconditions of the behaviors determine whether they are applicable in certain situations. So even if a KickForward behavior is in the starting set it is only considered and evaluated when an agent is close enough to the ball.

I have constructed the OpenBehavior as a supportive behavior to be able to move to a position where it is likely to intercept passes and lost balls. Such a behavior has previously been investigated using the strategic probe presented in 3.2. However this approach considers the same static passable position throughout the game. The idea of the OpenBehavior originates from the SPAR algorithm presented at CMU by Veloso, Stone and Bowling [12]. They argue that an agent not currently in pursuit of the ball should position itself where it anticipates it is most likely to come in possession of it in the future. And such a position they calculate by maximizing the distance from other robots and minimizing the distance to the ball and to the goal.

I assume that the most open position is the one furthest away from any other agent. The calculation needs to estimate some starting positions, as it is not feasible to consider every position in the field. I have made an approximation by starting from the landmarks that are currently known to a player. For all these landmarks the combined distance to all the agents in the field is calculated. The landmark with the greatest total value is then chosen. The approximated open point is located between this landmark and its nearest agent. When the behavior is simulated, this point is probed instead of the ball. If the position is near the opponent goal far away from opponents it will receive a high potential.

5.4 The Evaluation Loop with Anticipation

The cooperative planning in the EFA engine with the AM means a few additions to the evaluation loop. The applicable behaviors in the starting set are simulated for all the field players in the team one by one. Planning for teammates is done with a copy of the GM in which the mate and the planning agent have changed places. Heuristic potentials are calculated and saved together with the corresponding agent and behavior. The distribution of behaviors with the highest sum of potentials is chosen as the execution set. In effect an agent does not necessarily choose the behavior with the highest potential. The choice is made to maximize the potential of the team based on the anticipation of the actions of all its members.

Finding the execution set is a computationally complex problem, which demands that all possible permutations of three behaviors are tested. The starting set can be compared to a complete graph, in which all possible ways to visit tree nodes must be tested. The complexity of the problem for a starting set of size n is: $n * (n-1) * (n-2) \rightarrow O(n^3)$. This, however, is a worst-case scenario. As the evaluated behaviors are saved they are sorted by their potential. When starting to go through the behaviors with high potential, good permutations are found early. A lot of the following permutations can be disregarded, as they are fast found to be uninteresting. As the number of behaviors is limited the computation is a feasible addition to the evaluation loop.

6 Dynamic Role Model (DRM)

Here I will present a model for teamwork based on the dynamic distribution of roles, which are modeled by modifying the charges and probes of the EFA engine.

6.1 The Basis of the Model

Roles can be constructed by modifying the distribution of charges and the probe locations of the EFA engine. A well-thought-out composition of such roles can produce efficient team play. In the rapidly changing environment of RCF dynamic distribution minimizes the time it takes a team to react to a new situation. As discussed in Chapter 4, a reference position in the field can be estimated for each role. When distributing a role at any state of the game the agent closest to this position should be chosen to execute it. This is the basic foundation of the DRM.

For a football team to meet a global state of the game with a number of roles corresponds to dividing the global task into subtasks. As an agent is only capable of executing one role at a time the number of roles should be equal to the number of team members. With three field players in the SLRL one offensive charging role (OCR), one offensive supportive role (OSR) and one defensive role (DR) is considered. These three roles need to be flexible enough to efficiently handle any situation that may arise together. The goal of a subtask is not the same as the global goal and therefore the basic strategy of a role is different from that of the team.

6.2 Modeling the Roles

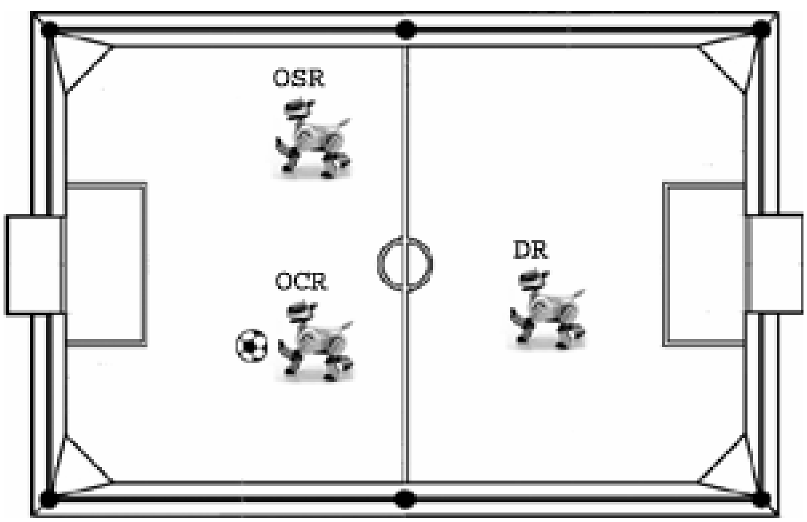

Fig. 5. Agents adjusted to their roles in a given state.

The task of the OCR is to pursue the ball moving it forward in the field, shooting it at the opponents' goal or passing it to a teammate. To induce such a behavior the static charges are set in the same way as described in 3.2. Positive charges are placed along the opponents' goal line and negative ones along that of the own goal. As the resource that the role seeks to modify is the ball, this is also the location in the field that is probed. Dynamic charges are also deployed in the way described for the EFA approach without coordination.

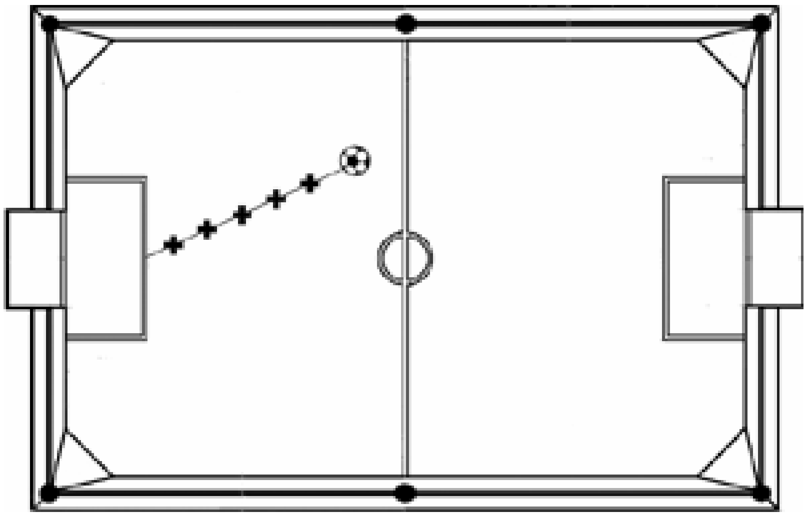

Fig. 6. The dynamic charges making up basic strategy for the DR.

The OSR is to place itself in a passable position upfield, at the same time being ready to intercept lost balls. As discussed in 5.3 such a position is approximated to the one farthest away from any other player. This is achieved by probing the agent's own position, instead of the ball, placing smaller negative charges on the position of all known other players. Keeping the negative static net charge of the own net but setting a smaller negative charge at the opponent's goal makes the behavior offensive but restrains the agent from running into the opponents goal. One could also consider keeping an open angle between the ball and the agent to ensure that it is passable, but so far I have not pursued this further. No charges are placed around the robot, as the position of the ball is not probed.

The goal of the DR is to stop offensive opponent attempts while also being ready for lost balls. This is achieved by placing itself between the ball and the own net ready to intercept the ball. Positive charges are placed along a line from the center of the agent's own penalty line to the ball to induce such behavior. No static charges are used as the basic strategic positioning for this role changes dynamically. Smaller positive charges are also placed on the positions of all known opponents. This causes the agent to charge an attacker with the ball. As the object that the role seeks to modify is the agent itself, the own position is probed here too. No charges are placed around the robot here either.

6.3 Distributing the Roles

An order of urgency can be established for these three roles, thereby stating the order in which they are to be assigned. It presents an explicit approach to the global task presented as three subtasks:

1. Move the ball up field → OCR
2. Defend the own net → DR
3. Present a pass alternative → OSR

As shown in 4.5, agents can be assumed to have a similar representation of the GM at any given time. With a good estimation of the position of every teammate, role distribution can be made without further communication.

An agent makes an individual evaluation of each consecutive role assigning it to the team member closest to its reference position. Once an agent is picked it is not considered for the next role. The reference position of the OCR is at the ball location. For the DR the whole line between the ball and the own net in the given state is considered as the reference position. As the OSR is always rewarded the agent least suitable for the other two roles, no position needs to be considered for it.

To make the procedure sensitive to the uncertainty of the agent representation, the distances to the reference points are divided by the anchor value associated with each team member. The order of assignment together with this sensitivity makes up a robust system for dynamic role distribution. If the evaluation has a different outcome in different robots simultaneously, the most important role will most likely be executed. It is likely to be awarded to the agent whose position was the most certain.

When the distribution of roles is done the agent executes the role it was given. Adjusting the charges and probes of the EFA accordingly it then follows the evaluation loop as described in 3.2. The starting set considered is the same as the one presented in 5.3.

7 Evaluating the Models

7.1 The Simulator

Some efficient way to test the cooperation models is needed in order to establish the different aspects of their utility. The obvious way would seem to be playing matches with a team of AIBOs using the Team Sweden software with the coordination extensions. In the absence of eight robots and a full size field such a test is not feasible. Furthermore, a fully computerized environment has the advantage of providing easy access to data from a large number of tests.

The AM and the DRM have been implemented and tested using a RoboCup simulator developed by Påhlstorp and Ronnovius [11]. It was developed for the cooperative map construction discussed in 4.5. Consequently, it uses the global structures of the Team Sweden architecture and simulates the various modules of the design with varied thoroughness. The movement of the robots is limited so that behaviors can only be translated into a direction and a speed. This constrains an agent's precision to a great extent. Kicks cannot be fully simulated as the only way to manipulate the ball is by bumping into it from some direction. Consequently only a number of the behaviors implemented in the HBM could be considered in the simulator. Vision is simulated by showing the agent the part of the global map that corresponds to the presumed robot's field of vision. The accuracy of perception can easily be adjusted by adding a preferred amount of noise to the local map. A great advantage of the simulator is that the source code is limited so that the RP module with the cooperative additions could easily be incorporated into it.

7.2 Performed Tests

To estimate how well a model for football cooperation performs it is ultimately desirable to see how many matches are won and goals scored using it against various opponents. As the possibility of precise movement is limited, the accuracy of goal attempts is low in the simulator. Additional aspects need to be tested to provide a base for the evaluation. In this way the effectiveness of the models can be established with a higher precision.

Ball possession is an important variable to calculate as it estimates how active the players are in the pursuit of the ball and how well they manage to keep it within the team. At the same time it approximates a team's ability to adjust to the dynamic environment, e.g., to intercept lost balls. This will thereby also, to some extent, give an estimation of how well a team is distributed on the field.

The amount of time the ball is located on either side of the field is also worth testing. It gives an estimation of how offensive, respectively defensive, a strategy is. Goals are made on the opponents' half of the field so a team should spend as much time there as possible. In order to play football offensively this is an important aspect to consider.

Goals scored and conceded are other variables considered to evaluate the models. As the game continues in the simulator without a kick off after a goal is scored, this variable has to be considered with caution. Several consecutive goals can be registered if the ball is trapped between a player and the net. When appearing close together like this, a number of goals should be considered as one, as they were the result of the same offensive opportunity. As discussed above, however, these variables involve a lot of chance due to the simulator design.

Changing the accuracy of the robots' perception can test the robustness of the models. The foundation of coordination for both the models lies in the own agent's view of the environment. For this reason it is interesting to examine how well they function if this view is inaccurate. This gives an evaluation of the fault tolerance of the other test variables.

The models were tested against the planning basic model of the simulator, the strategy of which is simply to have all the team members hunt the ball. In this way the utility of each model is tested independently of each other and an objective comparison can be made. The different variables were tested in three-minute passes. This was estimated to be enough time for a number of situations to occur in a game. Thirty such passes were evaluated for each model. The results are presented in Tables A and B.

Noise (%)	Ball Possession (%)	Time on Offensive Field (%)	Goals Scored	Goals Conceded
0	63	52	5	6
20	57	54	4	5
50	52	50	2	4

Table A: The test results for the AM.

Noise (%)	Ball Possession (%)	Time on Offensive Field (%)	Goals Scored	Goals Conceded
0	56	53	4	4
20	52	49	5	3
50	49	50	2	4

Table B: The test results for the DRM.

7.3 Evaluation

The results of the tests performed indicate that the limited motion and ball handling skills introduce a big element of chance into the simulator. Viewing the simulator's graphic visualization of the game it is obvious that the ball is often accidentally pushed off in some arbitrary direction. The validity of goals as a measure of successful football playing is, as mentioned earlier, questionable. Many of the goals are scored by chance. The seemingly random distribution of goals in the test results also indicates that this is the case. I have therefore mainly chosen to view the other variables as indicators of how well the models perform.

The AM showed a high degree of ball possession. Assuming the reliability of this variable it shows that the model distributes the agents well in the field. It would also imply that the AM adjusts effectively to the dynamic environment. The possession can be seen to decrease steadily with the increased amount of noise. This could be seen to reflect the fact that planning for the whole team is made on the basis of the distorted local map.

The DRM on the other hand shows a substantially lower percentage of ball possession. It can therefore be assumed that it does not distribute the agents as well as the AM and has a harder time adjusting to changes. One explanation could be that the charges making up the basic strategies of the roles were not compatible with the behaviors in the starting set. The amount of noise can here be related to the grounds for role distribution.

The time the agents of the AM spent on the opponents' half of the field is slightly longer then that on the own field. The difference is so small that no evident conclusion can be drawn from it. The strategy can be said to be neither offensive nor defensive. The DRM shows a similar result. The amount of noise does not seem to have any significance in this aspect. To increase the amount of uncertainty would probably also increase the amount of chance and therefore give a more random result.

The relatively limited success of the two models can probably be attributed, to a certain extent, to the impreciseness of the simulator. Opposing a team with a ball hunting strategy probably gives a one-sided evaluation of the utility of the team. There is, however, a lot of the components of the models that could have been constructed differently, e.g., the composition of the starting set and the position of the charges. A more thorough modeling and a careful evaluation of these models could probably improve their efficiency a great deal.

8 Conclusion

Coordination of agents is a complex issue that sets high demands on the diversity of the software structure. The constraints of the RoboCup domain raise these demands even higher. In this report two attempts to improve cooperation have been presented. The evaluation of the models showed at least one of them to be fit to deal with a dynamic environment. The reliability of the test results can however be questioned and further extended testing would be desirable in order to fully establish the utility of the models.

It would be interesting to see how the AM and DRM function within the complete Team Sweden architecture. This would add an additional level of movement precision to the team and make it possible to consider a large number of behaviors. This poses an interesting challenge for the future.

References

[1] The RoboCup Homepage
http://www.robocup.org
Verified 2003-08-18

[2] The Team Sweden Homepage
http://aass.oru.se/Agora/RoboCup/
Verified 2003-08-18

[3] Is it an Agent, or just a program?
http://www.msci.memphis.edu/~franklin/AgentProg.html
Verified 2003-08-18

[4] Hui Wang, Han Wang, Chunmiao Wang and William Y.C. Soh. Cooperation-Based Behavior Design. RoboCup 2001, Nan yang Technological University, Springer, 2001.

[5] The Sony AIBO Homepage
http://www.aibo.com
Verified 2003-08-18

[6] The Thinking Cap Homepage
http://aass.oru.se/~asaffio/Software/TC/
Verified 2003-08-18

[7] A. Johansson and A. Saffiotti. Using the Electric Field Approach in the Robo Cup Domain. Birk, Coradeschi, Tadokoro (eds) RoboCup 2001, Springer-Verlag, 2002

[8] Sascha Ossowski. Co-ordination in Artificial Agent Societies. Springer 1999.

[9] Yunpeng Cai, Jiang Chen, Jinyi Yao and Shi Li. Global Planning from Local Eyeshot; An Implementation of Observation-Based Plan Coordination in RoboCup Simulation Games. RoboCup 2001, Tsinghua University, Springer, 2001.

[10] Peter Stone and Manuela Veloso. Task Deccomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. Artificial Intelligence, 1999, Carnegie Mellon University, 1999.

[11] Daniel Påhlstorp and Dannie Ronnovius. Kooperativ kartkonstruktion för autonoma robotar. Degree thesis, University of Lund, 2003. Also available at: http://ai.cs.lth.se/xj/DannieRonnovius/rapport.ps.bz2

[12] Manuela Veloso, Peter Stone and Michael Bowling. Anticipation: A Key for Collaboration in a Team of Agents. Submitted to the 3rd International Conference on Autonomous Agents, Carnegie Mellon University, 1998