Executing Multi-Robot Cases through a Single Coordinator*

Raquel Ros
IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Barcelona, Spain
ros@iiia.csic.es

Manuela Veloso
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
veloso@cs.cmu.edu

ABSTRACT

It is challenging to design general robot soccer coordination behaviors that address individual states. We have successfully followed a case-based approach to define behaviors for a single soccer robot. In our multi-robot system we now distinguish *retriever* robots that access the case library, reason about the situation, and select the most appropriate cases. They communicate with the other robots and they all execute the retrieved case in a coordinated way. We evaluate our approach with two robots demonstrating that the robots successfully coordinate and the number of passes during a game highly increases compared to an approach with an implicit coordination mechanism.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms

Keywords

Robotics, coordination, case based reasoning.

1. INTRODUCTION

We have successfully presented a Case-Based Reasoning (CBR) approach for action selection in the robot soccer domain [4]. So far, we have only tested the work with a single robot. This paper extends this work to a multi-robot approach and having the robots coordinate using cases.

In the soccer domain, potential fields have been presented as coordination techniques in different occasions [6, 5] as well as dynamic role assignment [3, 1]. These approaches are mainly focused on coordination as an emerging property where the overall behavior of the team results in an organized sequence of actions. Besides this emerging coordination, an explicit coordination mechanism is necessary for certain situations to ensure the correct execution of actions. Kaminka and Frenkel [2] present an approach for flexible teamwork focused on synchronization and task allocation. They proved that rich interactions between robots can be a significant factor in task performance.

The intention of our approach is to be used in combination with other behavior-based approaches. The CBR approach is addressed to solve concrete situations that can be considered as exceptions during the execution of a game and need an explicit coordination mechanism to ensure success. The behavior-based approaches consist of defining high level behaviors (state-based behaviors) the robot executes based on the state of the environment. For example, a robot near the goal should try to get the ball and kick it to score avoiding the opponents, and a robot defending its goal should get the ball and clear it from the defense region.

Therefore the aim of this work is to include into an existing behavior-based approach a multi-robot case-based coordination approach. The robots execute state-based behaviors in general with an implicit coordination mechanism based on roles, while, for specific situations, cases will be triggered indicating the robots the actions to perform with an explicit coordination mechanism.

We believe that the advantage of using a case-based approach for coordination is that cases possess all the information about the actions to be executed by the involved members. Since all members are aware of the case being performed, they all know exactly what actions each member is going to execute and when. Thus, they only need to synchronize their actions and the expected outcome should occur (if there is no failure during execution).

We focus our work on the Four-Legged League of the RoboCup soccer competition. The RoboCup Four-Legged League Rule Book includes details on the official rules.

2. MULTI-ROBOT CASE DEFINITION

A case consists of the description of the environment (problem description) from a single robot's point of view and the actions the robots should perform for that state (solution description).

In our previous version of the CBR approach we presented

---

*Partial funding by the Spanish Ministry of Education and Science project MID-CBR (TIN2006-15140-C03-01) and partly sponsored by BBNT Solutions, LLC under contract no. FA8760-04-C-0002. Raquel Ros holds a scholarship from the Generalitat de Catalunya Government. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or any other entity.

AAMAS'07 May 14–18 2007, Honolulu, Hawaii, USA.

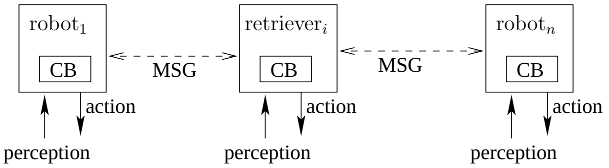


Figure 1: Case-based multi-robot architecture for n robots and $k = 1$ *retrievers*.

the problem description as a multi-robot approach (including a list of teammates). Yet, the solution description only considered a list of actions to be executed by a single robot. We now present a new definition where each robot is linked to the sequence of actions it should perform:

$$A = \{tm_1 : [a_{11}, a_{12}, \ldots, a_{1p}], \ldots, tm_n : [a_{n1}, a_{n2}, \ldots, a_{nq}]\}$$

where $n = 1..4$ is the number of robots, and p, q the number of actions teammate i performs. The actions are either individual actions (as get the ball and kick), or joint actions (as get the ball and pass it to robot i).

Although the problem to solve may have more robots on the field, the CBR approach can return a case with fewer robots. It first tries to find a case with an equal number of robots. If it fails, then it searches for cases with fewer robots until finding a good match (roles are not considered).

3. MULTI-ROBOT CASE EXECUTION

The multi-robot system is composed of n robots. All robots interact with the environment and among them, i.e. they perceive the world, they perform actions and they send messages to each other to coordinate. Each robot has a copy of the same case base so it can gather the information it needs to perform its tasks. We distinguish a subset of k robots, which are in charge of retrieving cases as new problems arise. We call these robots *retrievers*. There must be at least one retriever ($1 \leq k \leq n$). In this work we only consider one fixed *retriever* ($k = 1$). Figure 1 depicts the architecture for n robots.

When the *retrievers* retrieve a case, they inform the rest of the robots. A coordinator is chosen to synchronize the robots during the execution of actions. For simplicity in this work the *retriever* is set as the coordinator (although any of the robots could perform this task as well. In future work, we will improve the selection of the *retrievers* and coordinator robots). Next, all robots implied in the case retrieved, including the *retrievers* who take part in the case execution, perform the actions indicated in the case. Each robot can decide to abort the execution at any moment if, based on its own perception, the case is not applicable anymore.

Next we describe in more detail the robots behaviors based on our current implementation of the multi-robot system (one fixed *retriever*). All robots execute a *default* behavior, while only the *retriever* executes the *retriever/coordinator* behavior as well.

Retriever/Coordinator Behavior. Figure 2(a) shows the finite state machine for the *retriever*. In the initial state (INIT) the robot localizes in the field and searches the ball if it cannot see it (SEARCH). Next (RETRIEVE), the robot retrieves the case based on its own perception and the information given by the other robots (their positions in the field) and it informs the retrieved case ID to the rest of the robots (SEND INFO). At this point (WAIT SYNC) it waits for all the robots (including itself) to be prepared to execute the case (i.e. all robots go to their initial positions).

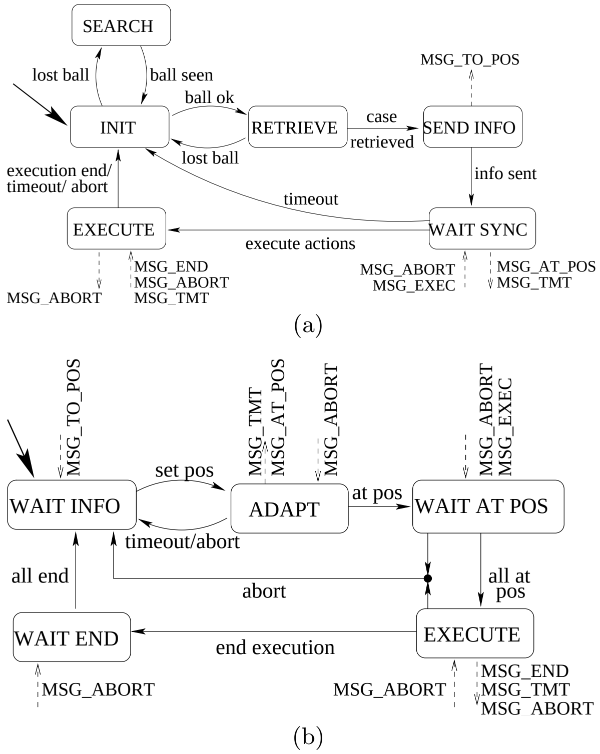


Figure 2: (a)Retriever/coordinator behavior. (b) Default behavior.

Once all robots report they are ready, the next synchronization point occurs (EXECUTE). The robot remains in this state until all robots (including itself) finish executing the actions indicated the case solution. Finally the robot goes back to the initial state (INIT) to retrieve a new case.

Default Behavior. Figure 2(b) shows the finite state machine of the behavior each robot executes (all robots including the *retrievers*). In the initial state (WAIT INFO) the robot waits to receive the information from the *retriever*. When the message arrives, the robot moves to its initial position (ADAPT) and sends a message to inform it is ready to execute the actions. It waits in the first synchronization point (WAIT AT POS) until all robots are prepared. Next (EXECUTE), it executes its assigned actions until finishing the sequence and waits for the others to finish as well (WAIT END). Finally, it transits back to the initial state waiting for the next retrieved case.

Any robot may abort the execution of the case due to three event: timeouts, lost messages, and lost cases. The first one occurs due to incorrect localization (the robots spend to much time reaching the expected positions). The second one is caused because of network problems. And finally, the third event is triggered when a robot based on its own perception detects that the current case is not applicable anymore (because the state of the environment has significantly changed). In any case, if the *coordinator* receives an abort or a timeout message, it immediately informs all the robots to abort the current execution and they all go back to their initial states.

4. EVALUATION

We performed two sets of experiments. The first one is addressed to evaluate the coordination protocol among the robots. While the second one is to analyze the impact of the CBR approach in the task of executing passes between robots and scoring goals.

Multi-Robot Interaction. We designed a first set of experiments to evaluate the success of the case execution based on the coordination protocol. A single trial consists on positioning two robots in the field and letting them execute a single case. The messages sent among the robots are logged in a file for further evaluation. We consider that a case execution succeeds when both robots complete the executions of their actions. Otherwise, the case execution fails.

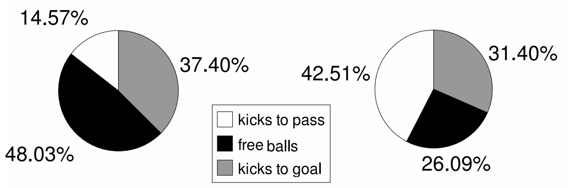
For this experiment we performed 210 trials. After analyzing the log files we obtained the following results: 130 (61.90%) cases were successfully executed and 80 (38.09%) failed. 22 trials (10.48%) failed due to timeouts, while 58 (27.62%) failed due to network problems.

As we can observe, the main reason for failure is because of losing messages. This is caused because we are using a simple communication layer. As future work we propose to design a more robust message mechanism to minimize the number of lost messages.

Multi-Robot Task. We address this second part of experimentation to analyze the performance of the approach based on the type of shots the robots execute in a period of time. We define three types of shots: *kick to pass* (a first robot kicks the ball towards a second robot, and the second one gets the ball to perform another action with it); *kick to goal* (the robot kicks the ball towards the goal and the ball enters the penalty area); and *free ball* (a robot kicks the ball to an open area of the field and there is no teammate to get it).

The experiment consists of a 60 minutes game where two robots from the same team try to score (in future work we will include opponents as well). The initial position of the ball is the center of the field and the robots are located in their defense area. Every time the robots score a goal (as defined previously) or the ball goes out of the field, it is positioned back to the center of the field and the robots remain at their last positions. The goal of the experiment is to prove that the team using a combined approach (our approach and an implicit coordination approach) maximizes the percentage of kicks to pass (is more cooperative) and minimizes the percentage of free balls with respect to a team with only an implicit coordination approach. The case base for this experiment is composed of 44 manually created cases, which we believe are enough to test this work.

Figure 3 shows the percentage of shots for each approach. We can observe that the percentage of kicks to pass using the combined approach (43%) is much higher than the pure implicit coordination approach (15%). Also, there is a big difference with respect to to the percentage of free balls (26% with the combined approach, and 48% with the implicit coordination approach). It is easy to see that both values are directly related. As the number of kicks to pass increases, the number of free balls decreases. When having kicks to pass, shots are directed towards some teammate, and not to an empty region on the field. This a very important issue when playing a real game with opponents: having free balls increases the chances of the other team to get the ball. Therefore, ensuring the possession of the ball by some teammate is a very desirable property during a real game. Finally, the percentage of kicks to goal with our approach is lower compared with the other approach. The reason is that the robots always try to cooperate passing the ball to a teammate, instead of trying to kick to goal individually.



(a) (b)

Figure 3: Percentage of shot types during experimentation: (a) implicit coordination approach and (b) multi-robot case-based coordination approach.

5. CONCLUSIONS

We have proposed a case-based coordination mechanism where we use a case-based reasoning approach to coordinate the multi-robot system through cases. We have presented a multi-robot system where a set of robots, the *retrievers*, first take care of the reasoning, proposing solutions for the different states of the environment. Afterwards, they interact with the remaining robots to jointly execute the actions of the solution proposed.

The experiments presented confirm the effectiveness of the coordination between robots to control the ball in the field by the robots of the same team and reducing the number of free balls. Yet, we still have to improve the message delivery in order to decrease the number of lost messages.

6. REFERENCES

[1] K. Yoshimura and N. Barnes and R. Rönnquist and L. Sonenberg. Towards real-time strategic teamwork: A RoboCup case study. In *RoboCup 2002*, 2003.

[2] G. Kaminka and I. Frenkel. Flexible teamwork in behavior-based robots. In *AAAI*, 2005.

[3] C. McMillen and M. Veloso. Distributed, Play-Based Role Assignment for Robot Teams in Dynamic Environments. In *Proc. of DARS*, 2006.

[4] R. Ros, M. Veloso, R. L. de Màntaras, C. Sierra, and J. Arcos. Retrieving and Reusing Game Plays for Robot Soccer. In *Advances in Case-Based Reasoning*, volume 4106/2006, 2006.

[5] A. Tews and G. Wyeth. Multi-Robot Coordination in the Robot Soccer Environment. In *Proc. of the Australian Conf. on Robotics and Automation*, 1999.

[6] D. Vail and M. Veloso. Dynamic Multi-Robot Coordination. In *Multi-Robot Systems.* 2003.