

ARCHITECTURE OF COOPERATION FOR MULTI-ROBOT SYSTEMS

André Scolari Conceição^{*,1} A. Paulo Moreira^{*}
Paulo J. Costa^{*} Luis Paulo Reis^{**}

** Faculty of Engineering of the
University of Porto - Porto - Portugal.*

[scolari, amoreira, paco]@fe.up.pt

*** Artificial Intelligence and Computer Science Laboratory
of the University of Porto - Porto - Portugal.*

lpreis@fe.up.pt

Abstract: This paper presents a new architecture for coordinating multiple autonomous robots in the execution of cooperative tasks. The architecture is based on the definition of a strategy that uses different tactics, setplays and roles. Roles enable to configure individual behaviour by performing specific tasks using a given set of actions. The proposed architecture allows flexible and efficient multi-robot operation in dynamic environments. The paper also presents an application of the proposed architecture to a complex domain such as middle-size robotic soccer. Our architecture is also generally enough to be applied in different robotic soccer leagues and similar multi-robot problems. *Copyright © 2006 IFAC*

Keywords: Architectures for Cooperation, Multi Agent Systems, Artificial Intelligence, Coordination, Teamwork.

1. INTRODUCTION

Dynamic complex systems with autonomous components often require appropriate mechanisms for cooperation, coordination and dynamic configuration. It is crucial that these mechanisms allow the system to execute continuously for a long period of time without interruptions.

(Frazao and Lima, 2004) presents an agent-based software architecture that intends to close the gap between hybrid systems and software agent architectures. In (Balch and Arkin, 1998), a behaviour-based approach to robot formation is presented. (Fujii *et al.*, 2005) presents a cooperative control method takes into account the achievement level

of the objective. The objective is defined from the essential requirements for achieving a task. In this method, the cooperative behaviour among agents is realized by performing the evaluation of degree of achieving an objective and by sharing the evaluation result among them. (Beaudry *et al.*, 2005) presents a architecture that is mainly constituted of 4 hierarchical levels: Main Decision, Pattern, Role, Behaviour, Action. In this architecture, each machine must be constituted of an arbitrary number of levels of decision nodes, leading to behaviour nodes which then activate specific actions to implement desired behaviour.

Given all this background, in this paper we present an architecture for coordinating multiple autonomous robots in the execution of cooperative tasks. This application may involve since cooperation in rescue tasks until cooperation of

¹ Supported by the Program Alβan, the European Union Program of High Level Scholarships for Latin America, scholarship n.E04D028256BR.

AGVs (Automated Guided Vehicles) in manufacturing systems. The proposed architecture allows more flexible and efficient robot operation. Our study case is the robotic soccer domain, which is a challenging testbed for research in multi-agent and multi-robot cooperation in a highly dynamic and uncertain environment. A soccer game is a specific but very attractive real-time multi-agent environment from the viewpoint of distributed artificial intelligence and multi-agent research. If we regard a soccer team as a multi-agent system, a lot of interesting research issues will arise (Kitano *et al.*, 1997b). According to an analogy between the robotic soccer and a general application, we can use the following terms: (Robotic soccer \leftrightarrow general application),

- Coach \leftrightarrow Central supervision system;
- Strategy, Tactics \leftrightarrow Operation mode (i.e. safety, economical, normal);
- Setplays \leftrightarrow Cooperation tasks (i.e. initialization, standby);
- Roles \leftrightarrow Functions (master AGV, AGV for charging);
- Tasks \leftrightarrow Tasks (i.e. transport to gate A, Go to gate B);
- Actions \leftrightarrow Low level controls.

The rest of the paper is organized as follows. Section 2 presents a brief overview of the Middle Size Robot League. Section 3 presents the 5dpo-2000 team description and the omni-directional mobile robots. In section 4 the team strategic coordination and tactics are presented. A hierarchy for the robot lower-level control (Roles, Tasks and Actions) is presented in section 5. In section 6, the Setplays is presented. Section 7 will be dedicated to opponent modelling and tactical selection. Finally, the conclusion and future works are drawn in section 8.

2. ROBOCUP-MIDDLE SIZE ROBOT LEAGUE OVERVIEW

RoboCup is an international research and education initiative. It is an attempt to foster AI and intelligent robotics research by providing a standard problem where wide range of technologies can be integrated and examined, as well as being used for integrated project-oriented education. For this purpose, RoboCup chose to use soccer game as a primary domain, and organizes RoboCup: (The Robot World Cup Soccer Games and Conferences). In order for a robot team to actually perform a soccer game, various technologies must be incorporated including: design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, and sensor-fusion. The Robocup Federation stated the ultimated goal of the Robocup initiative as

follows: "By 2050, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official FIFA rules, against the winner of the most recent World Cup of Human Soccer". (Kitano *et al.*, 1997a)

In the soccer middle-size robot league, two teams of 4-6 mid-sized robots with all sensors on-board play soccer on a field. Relevant objects are distinguished by colours. Communication among robots is supported on wireless communications. No external intervention by humans is allowed, except to insert or remove robots in/from the field. The main Middle size robot league rules and regulations for 2006 are (Committee, 2006):

- The dimensions of the field are 12x8m.
- Goals dimension are: 2m between the posts and the distance from the lower edge of the crossbar to the ground is 1m.
- The ball is: spherical, made of leather or other suitable material, of a circumference of not more than 70cm and not less than 68cm.
- The number of players: a match is played by two teams, each consisting of not more than six players, one of whom is the goalkeeper. The maximum difference of number of players between the two teams starting a match is defined as 1, if the team with less robot has 4 or more players.
- Robot size/weight: each robot must possess a configuration of itself and its actuators, where the projection of the robot's shape onto the floor fits into a square of size at least 30x30cm and at most 50x50cm. The robot height must be at least 40 cm and at most 90 cm. The maximum weight of a robot is 40 kg.
- The duration of the match: a match lasts two equal periods of 15 minutes.

3. TEAM DESCRIPTION

We focus attention on the mid-sized omni - directional mobile robots, as shown in Fig.1, built for the 5dpo-2000 Robotic Soccer team from the Department of Electrical and Computer Engineering at the University of Porto at Porto, Portugal. 5dpo-2000 team, since 1999, has already participated in 8 international Robocup competitions and 6 local events. Previous robots generations can be seen in (Moreira *et al.*, 2004) (Moreira *et al.*, 2003).

The robots are equipped with four omnidirectional wheels connected to geared motors. Connected to each wheel there is a industrial encoder to measure its speed. Each pair wheel-encoder is connected to a controller board. This board was developed by the team and has a microcontroler

that measures the wheel speed and implements a local controller. This controller maintains the requested speed and is based on a modified PID. This low level loop has a sampling frequency of 100Hz. The four controllers are connected to the PC by a RS-232 link running at 115200 baud. Each robot has a standard PC motherboard with a 2GHz Celeron processor. The motherboard is inside a specially made case. This case is attached to the robot frame with some shock absorptive material to protect it. Also, it eases inserting and removing the PC from the robot. Nevertheless, there is no hard disk and the Linux OS and the programs are stored in a 256 MBytes Compact Flash Card connected to the IDE.



Fig. 1. On the left the 5dpo-2000 Team Goal Keeper. On the right: the standard field player

The robot's control and communication structure can be seen in Fig. 2.

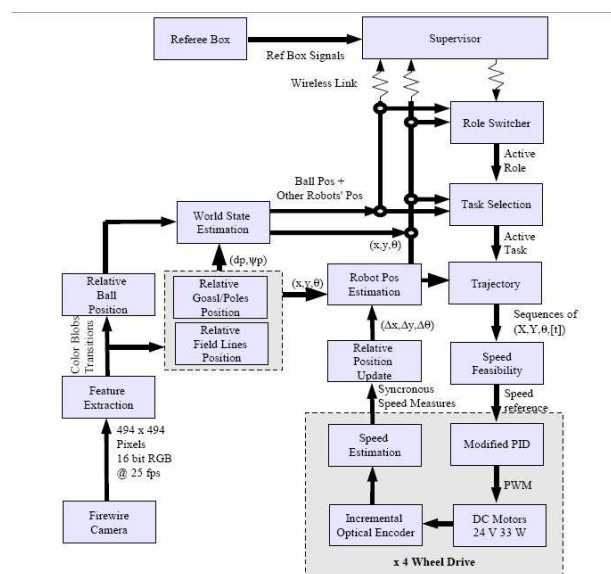


Fig. 2. Robot's control and communication structure.

There are four distinct software layers implemented by the 5dpo-2000 Team. These layers are

sometimes defined at physical boundaries but also at the quality of the data that they process:

- At the lowest level, there is the controller for each wheel. This system is very important for the performance of the low level skills. This software runs in a microcontroller located on the same board of the power electronics. It was optimized for the robot's dynamics. It uses a modified PID (Conceicao *et al.*, 2005b) (Conceicao *et al.*, 2005a) with an anti-windup protection, extra reference and measure filters and a feed forward path to speed up the transient response and compensate for some unavoidable non linearities. Another important feature is that it allows to sample the wheel speed at the same time for all the wheels. This avoids odometry and trajectory generation errors introduced by non synchronous sampling of the wheel speeds;
- In the next level there are a few modules. One deals with the generation of feasible translational and rotational speeds from a desired trajectory and converts those robot speeds in the desired wheel speeds. There is also the the module that uses the speeds reported by the optical encoders to generate the odometry data and update the estimated robot position. Another very important module is the one that deals with the image captured by the omnidirectional vision system and extracts the most important features: the color blobs associated with the ball, goals/poles and the other robots and also the green/white or white/green transitions that signal the field lines. This information is used to construct an estimation of the relative position of the ball, other robots and goals. Coupling that information with the extracted field lines, a estimation of the robot position can be achieved. That is used to update the world state as it is seen by the robot and also broadcast to all the other robot and the supervisor.
- In this level the the modules deal with somewhat more abstracted data. It is assumed that the robot position is known, with an estimated degree of accuracy and the output is the desire trajectory with eventual kicking moments. Here, there is a module that can change the actual role for the robot (attacker, second attacker or defender) when there is a request from the supervisor. In the absence of data from the supervisor (that can happen very often due to problems with the wireless communication) the robot defaults to its assigned role. Knowing the role and the game state, there is a state machine that selects the task to be performed by the robot. For each task, the desired trajectory can be generated

using different algorithms. The algorithms selected depend on the perceived world state: if there are nearby obstacles, if there is great confidence on the position estimate, if the robot is being blocked by other robot, etc.

- In the higher level there is the supervisor module that receives the information broadcast by all the robots and the referee box commands. It creates a world state that includes the robots' positions, their actual roles and the game state as informed by the referee. Another issue that is extracted by the supervisor is the "aliveness" of each robot. The localization data sent periodically by each robot is used as a keep-alive signal and the absence of that signal for some time can indicate a problem with the "missing" robot. That information can be incorporated in the role changing decisions. All that data is passed down to the robots with the coordinated role changes.

4. TEAM STRATEGY AND TACTICS

The Coach (central supervisor) is a privileged agent and is the reference point of the team. From the informations received from the players, the coach has the function to decide the strategy of the team, based on some activation rules.

The team strategic coordination is based on tactics (Reis and Lau, 2001)(Reis *et al.*, 2001). Tactics are responsible for coordinating changes in the behaviour of the team. Tactics are formed by some specific global parameters and by Setplays.

Setplays define the behaviour and positioning of the players for each specific situation of the game. Generally, situations like kick off, goal kick are attributed to Setplays. Also some specific conditions in the game, easy to recognize, like for example, a player free with the ball with free path for the opponent goal, also generate the use of specific Setplays.

Figure 3 presents a hierarchic diagram of the architecture of cooperation. The architecture is composed by several tactics that use different setplays for different game situations. For example in a goal kick situation, a given tactic may use setplay 1 (for example direct shoot to the goal) and another more conservative tactic may use setplay 3 (for example a pass to the nearest player followed by a dribble to the goal). If there is no set play active, players behave normally, i.e. they use their roles to decide their behaviour.

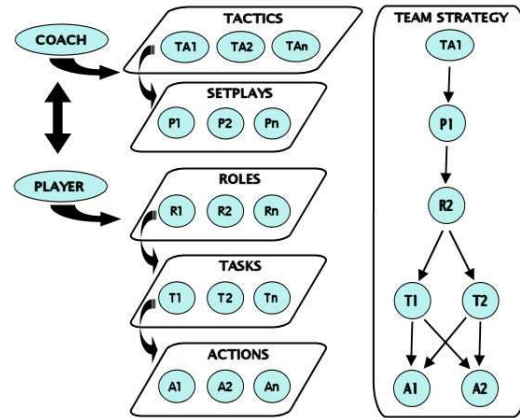


Fig. 3. Architecture of cooperation.

5. ROLES, TASKS AND ACTIONS

Roles, tasks and actions, form a hierarchy for the robot lower-level control. Roles control the robot behaviour through the use of tasks, while tasks instantiate actions to execute: A more detailed description of this architecture based on Roles, Tasks and Actions is:

- **Roles:** Each robot executes its own role, attributed following the coach instructions. Robot skills or strategical positioning can define its role(i.e. attacker, second attacker, defender, keeper).
- **Tasks:** Players skills (i.e. dribble, shoot, block the ball, pass the ball).
- **Actions:** Low level actions(i.e kick force, robot velocity, trajectory following control).

For example in our robotic soccer domain, a throw-in is a method of restarting play in which a given team may kick the ball from a given spot and the opponent player may not touch the ball until it is kicked. For example, if we consider the game situation throw-in-for-us, the team have to apply Setplay = "ThrowInforUs". A throw-in is awarded (Committee, 2006):

- when the whole of the ball passes over the touch line, either on the ground or in the air,
- from the point where it crossed the touch line,
- to the opponents of the player who last touched the ball.

In this Setplay, each player executes a Role through tasks. The execution of the SetPlay can be aborted, this happen if something wrong happens. This decision is based on rules set:

- the ball do not go to central player;
- the opponent get the ball;
- Setplay time is over.

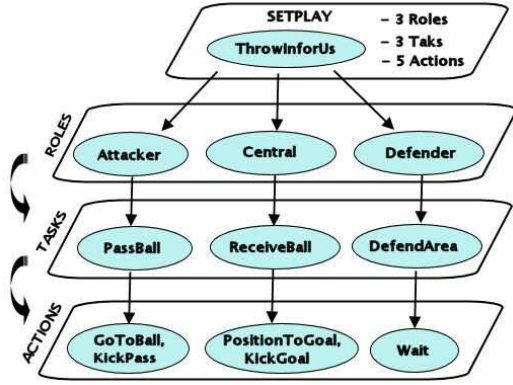


Fig. 4. Setplay: ThrowInforUs.

6. SETPLAYS

As well as in Tactics, the Setplays are selected from a rule set. The activation rules for the studied plays are usually based on game situations (i.e. goal kick, kick off, throw in). Setplays can also be used in other cases. For example in the case of passing the ball to another player of the team or in order to exercise a more rough play against the opponent.

The robot actions during a given setplay are defined through a simple and efficient language, that allows informations as position and time to be forward. Next, we present a simple implementation and easy representation of a Setplay(Reis and Lau, 2002):

```
(setplay Triang)
(step 0) (player 3) (wait 30) (time 20)
(formation 3 3 (0 0) 4 (-5 0) 5 (5 0))
(step 1) (player 3) (wait 5) (time 20)
(formation 3 3 (0 0) 4 (-5 10) 5 (5 10))
(actions 1 (pass 4 step 2) (pass 5 step 3))
(step 2) (player 4) (wait 5) (time 20)
(formation 3 3 (5 0) 4 (-5 0) 5 (0 10))
(actions 1 (pass 5 step 3) (pass 3 step 1))
(step 3) (player 5) (wait 5) (time 20)
(formation 3 3 (5 10) 4 (0 0) 5 (-5 10))
(actions 2 (pass 3 step 1) (pass 4 step 2))
```

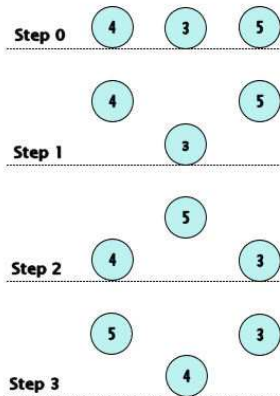


Fig. 5. Setplay: Triangulation.

In this setplay named Triang player 3 has initially the ball and starts its execution by communicating with the other players that the setplay Triang is going to be executed. It also sends some setplay parameters like the field point where the setplay is going to be executed, level of abortion (indicating the easiness to abort the execution of this setplay), setplay speed (indicating the velocity for the execution of the setplay). Setplay execution starts with the players moving to their initial formation position. After a given time set by the setplay speed parameter and time factor in the setplay definition, player 3 starts execution of setplay step 1. This execution is only started if all players are in their correct formation positions with a given tolerance that depends on setplay abortion parameter. In Step 1, players move to their new position and player 3 may perform two different actions: pass to player 4 leading to step 2, or pass to player 5, leading to step 3. In each step players move to different formation positions and have different actions that may perform leading to a new step until the plan is finished or aborted.

Figure 5 presents robot positions in each step. The Setplay is used to make a simple infinite triangulation, that may switch direction, between the three players.

7. OPPONENT MODELLING AND TACTICAL SELECTION

The default game strategy, includes a default scheme for tactical selections based on informations about the situation of the game (i.e. score, play time) and also informations about the opponent characteristics. A simple idea of selecting a tactic based on the time of the play and the score difference of the game, is presented as follows:

T_{ini}	T_{fin}	$DifScore$	$Tactic$
00	10	1,-2,-1	1,3,3
10	15	1,-2,0	1,3,1
15	20	-1,1,-2	1,2,3
20	30	-1,1,-1	1,2,3

Table 1. Tactical selection.

In this case, the tactics are: 1 = normal, 2 = defense and 3 = attack. Times T_{ini} and T_{fin} represent the initial and final time, respectively. In the column $DifScore$ of table 1, there are three possible score differences, which are going to decide which tactic is going to be used. For example, in the first row of the table, if the score difference is 1 positive goal, tactical 1 is assumed; if the difference is 2 goals against, tactical 3 is assumed instead.

The opponent modelling is another important source of information, which can be used to decide

which tactic to adopt. As modelling example, we can mention:

- team's collective behaviour;
- player's individual decision;
- ability in the action's execution;
- physical capacities, as velocity and type of kick;
- player's position through the game.

This information may be used by the coach for performing tactical selection in a more informed way.

8. CONCLUSION

This paper briefly presented an architecture enabling multi-robot cooperation in the execution of complex tasks in dynamic domains. A very similar structure is being used in our robotic soccer teams participating in the middle-size, small-size, Sony 4legged, simulation 2D, simulation 3D and rescue simulation leagues. It has revealed to be an useful and practical structure, which allows to configure the cooperation between robots in a systematic and structured way. In Robocup 2006(Bremen) our robots achieved first place in the Simulation 3D league and second place in the Small-size league. The architecture is also going to be applied in mobile robots of industrial applications (AGVs), which are responsible for the transport of material between job stations and automated storage retrieval system.

9. ACKNOWLEDGEMENTS

This research was partially supported by FCT under grant POSC/EIA/63240/2004. The authors wish to thank the Portuguese FCT - National Science and Technology Foundation for the fellowship granted.

REFERENCES

- Balch, T. and R. Arkin (1998). Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation* **14**(6), 926–939.
- Baudry, J., S. Marleau and P. Fournier (2005). Robofoot epm team description. *Robocup 2005*.
- Committee, MSL Technical (2006). Middle size robot league rules and regulations for 2006.
- Conceicao, A. S., A. P. Moreira and P. Costa (2005a). Dynamic model identification and control of a omni-directional mobile robot's motors. *Proceedings of the Scientific Meeting*.
- Conceicao, A. S., A. P. Moreira and P. Costa (2005b). Trajectory tracking for omni-directional mobile robots with restrictions on motor's velocities. *Proceedings of the Scientific Meeting*.
- Frazao, Joao and Pedro Lima (2004). Agent-based software architecture for multi-robot teams. *IAV 2004 - 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles* pp. 614–620.
- Fujii, H., F. Otsuka, N. Sema, M. Kawashima, E. Sugiyama, S. Niizuma, K. Kawakami, S. Makishima and K. Yoshida (2005). Eigen team description. *Robocup 2005*.
- Kitano, H., Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara and E. Osawa (1997a). Robocup: A challenge problem for AI. *AI Magazine* **18**, 73–85.
- Kitano, Hiroaki, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda and Eiichi Osawa (1997b). RoboCup: The robot world cup initiative. *Proceedings of the First International Conference on Autonomous Agents (Agents'97)* pp. 340–347.
- Moreira, A. P., P. Costa, A. Sousa and L.P. Reis (2003). 5dpo-2000 team description for year 2003. *Team Descriptions of the RoboCup 2003 Games and Conferences*.
- Moreira, A. P., P. Costa, A. Sousa and L.P. Reis (2004). 5dpo-2000 team description for year 2004. *Team Description Papers, RoboCup 2004*.
- Reis, Luis Paulo, Nuno Lau and Eugenio C. Oliveira (2001). Situation based strategic positioning for coordinating a team of homogeneous agents. *Balancing Reactivity and Social Deliberation in Multi-Agent System, From RoboCup to Real World Applications* **2103**, 175–197.
- Reis, Luis Paulo and Nuno Lau (2001). Fc portugal team description: Robocup 2000 simulation league champion. *RoboCup-2000: Robot Soccer World Cup IV* **2019**, 29–40.
- Reis, Luis Paulo and Nuno Lau (2002). Coach unilang - a standard language for coaching a (robo) soccer team. *RoboCup 2001: Robot Soccer World Cup V* **2377**, 183–192.