Towards a Distributed Multi-agent System for a Robotic Soccer Team.

Nadir Ould Khessal

Temasek Engineering School. Temasek Polytechnic
21 Tampines Ave 1 Singapore 529757.
nadirok@tp.edu.sg

Abstract. Many AI professionals consider RoboCup small robots league competition as an ideal platform for testing distributed artificial intelligence techniques. Among these techniques are Multi-Agent systems (MAS), which advocate collective intelligence by focusing on autonomy of agents and their intercommunication. Multi-Agent Systems have been used by computer scientists and software engineers in several disciplines such as Internet and Industry [16]. For the robotics community Multi-Agent Systems was a paradigm shift from the classical centralized approach in building intelligent machines. By the late 80's MAS were used in several multi robot systems ranging from cellular robots (Fukuda et al) [1] to a team of trash-collecting robots (Arkin et al) [2]. This paper describes a distributed approach in implementing the Multi-Agent system architecture of a robotic soccer team, Temasek POlytechnic Team(TPOT).

1. Introduction

The major characteristic of the RoboCup soccer competition is the dynamic nature of the environment in which robots operate. The only static object in the competition field is the field itself. Team and opponent robots as well as the ball can be placed anywhere in the field, be it a purposeful strategic positioning, a missed action or a forced displacement. This has led many researchers to shift from the traditional model-based top down control [3,4] to a reactive behavior based approach [5,6,7,8,9,10]. Robots need not waste a huge amount of resources building maps and generating paths that might prove useless at the time of action. Instead robots are supposed to react to the actual changes in the environment in a simple stimulus-response manner [11]. However due to the size limitations imposed by the RoboCup small robots league (15cm diameter circle) and rich visual input, on-board vision proved to be a complex and expensive task.
Due to these constraints, several RoboCup researchers [12,13] have turned to off-board global vision. Cameras are placed above the field from where relevant information about the field such as robot coordinates, identities and ball position is dumped to a stand-alone computer. This centralized approach in building the control

M. Veloso, E. Pagello, and H. Kitano (Eds.): RoboCup-99, LNAI 1856, pp. 357–365, 2000.

system has led to the adoption of a hybrid (deliberative and reactive) approach. Reactive behavior based agents are embedded in the robots for urgent time-critical actions such as obstacle avoidance and command execution. Visual data manipulation and filtering as well as high-level reasoning e.g. ball position prediction are done in the remote computer.

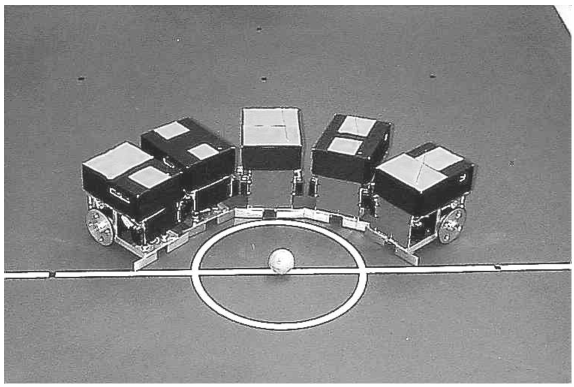

Fig. 1. Temasek Polytechnic RoboCup Team TPOT

In this paper we will present a hybrid control architecture, distributed among the robots (figure 1) and the host computer. Associated with every robot is an embedded agent, in charge of navigating the field and executing commands generated by the remote agent. Remote agents select and implement the required tasks, based on the visual data provided by the vision system and the strategy selected by the reasoning module.

2. System Architecture

The system hardware consists of a Pentium host computer, a vision system based on Newton labs Cognachrome vision card, RF transmission system and five robots (figure 2).

- ***The robot*** **:** The robot on-board controller is implemented in an 8 bit processor running at 9.216MHz with an on-board memory of 512kbyte RAM and 512Kbyte EEPROM. The board also includes a real time clock and programmable timers.

- ***Sensors:*** Attached to the robot are three infrared sensors mounted in the front and rear to detect obstacles whilst moving.

- ***The communication module:*** The host computer transmits commands to the robot via radio transceivers utilizing UHF radio waves. Each robot has its own transceiver and a unique node address. The low-powered wireless system transmits less than 1mw of power and is effective over distances of 3 to 30 meters. Two-way communication rates of up to 38.4Kbps are possible. The command set is transmitted as text code piggybacking on the transmission protocol. Commands are sent and received from the transceiver using an RS-232 interface.

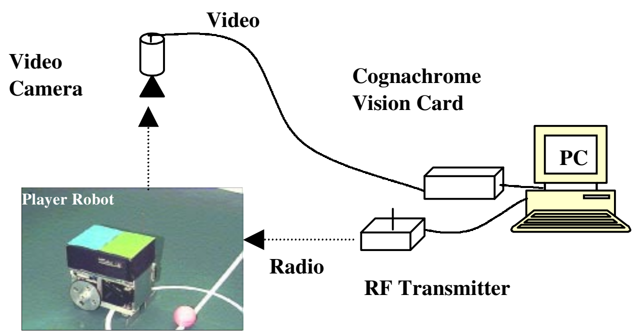

Fig. 2. System Overview

- ***Vision:*** A global vision system, which consists of color camcorders and a special image processor (MC68332), is used. The system is able to segment and track the robots and ball at a high frame rate. Each robot has two color pads. The image processor is trained to see the different colors and gives the locations of the center of gravity of the two color pads. Hence the orientation and robot position are known. Color pad areas are used to distinguish between different robots and minimize latency.

3. Distributed Multi-agent System

Our approach in implementing the control architecture of the robots is based on dividing each robot controller into two parts: Embedded agent running on the on-board processor and situated in the environment (field) and Remote agent running in the off-board host computer and situated in an abstract model of the filed. The embedded agent consists of several reactive behaviors competing with each other through the use of activation levels (inhibition and suppression). The main role of the embedded agent is to execute commands issued by the remote agent and navigate safely the soccer field while avoiding other robots and obstacles.

The remote agent on the other hand implements strategies generated by the reasoning module. Based on the current score and performance of the opponent team, the reasoning module selects a strategy from a pool of pre-designed strategies and downloads it to the remote agents.
This enables the agents to select the appropriate tasks for each robot. Such tasks will enable the robot to intercept the ball, follow a target or simply move to a predetermined position. These behaviors are implemented in every robot's remote agent except the goalie. This allows the robots to swap roles e.g. from being a defender to a forward and vice versa.

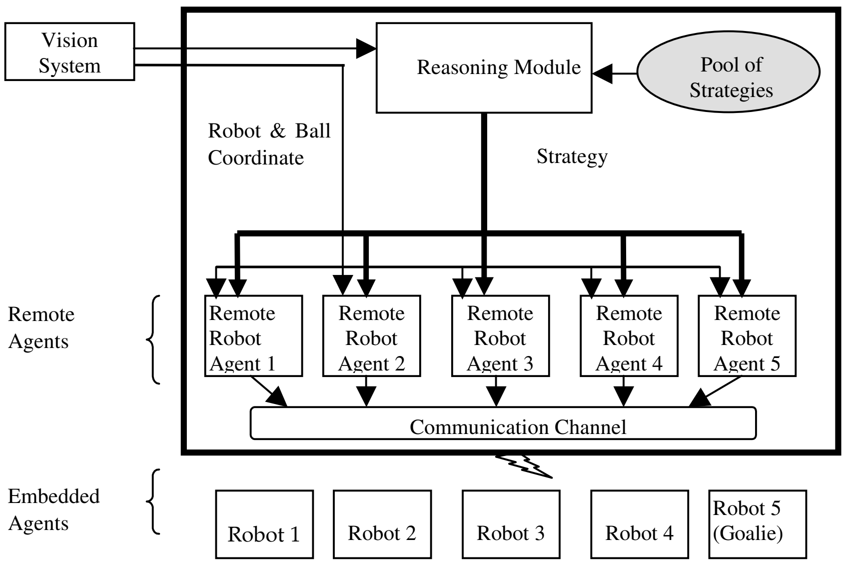

Fig. 3. Distributed Muti-Agent Architecture

3.1 Embedded Agents

An embedded agent consists of reactive behaviors designed to perform low-level navigational tasks. These behaviors are simple stimuli-response machines where the stimulus can be a sensory input or an incoming RF command. The arbitration mechanism is based on a fixed prioritization network [14]. The response of a single higher priority behavior takes over the control of the robot whenever the associated stimulus is present (figure 4).
Obstacle avoidance is the main autonomous task done by the robot. Using the three infrared sensors, the robot moves away from obstacles using the *Avoid Left* and *Avoid right* machines. The robot than moves a certain distance until a straight line path to the target position is clear. The remote agent detects that the robot is out of the

previously computed path and re-computes and transmits a new path. Due to the dynamic nature of the environment obstacles are not taken in consideration while planning a path for the robot.
Remote agents transmit paths in the form of a turning angle followed by a traveling distance. The two machines on the robot i.e. *Turn* and *Move* are in charge of executing these commands. Note here that unlike the *Move* machine, the *Turn* machine is of a highest priority and therefore un-interruptible.

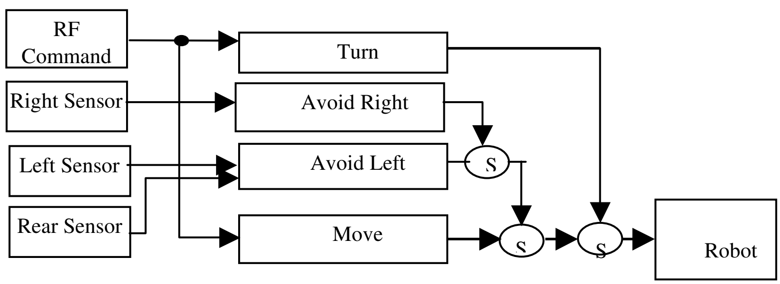

Fig. 4. A Network of Stimuli-Response Machines.

3.2 Remote Agents

For the robot to be able to play soccer it needs some basic skills such as moving towards the ball position, kicking the ball towards the goal area, intercepting the ball and passing the ball to a team member. Most of these skills could be performed by the following behaviors:

- *Intercept_ball.* This machine enables the robot to move behind a predicted ball position before kicking it towards a target area. The target area could be the opponent goal keeper area (in an attempt to score a goal figure 5.), a clear area in front of a team member (ball passing) or simply the opposite side of the field, in the case of a defending robot. This behavior is achieved by the robot first moving to an intermediate position. Once there, the robot charges towards the ball. The intermediate position is determined by computing the two lines starting at the edges of the target area and intersecting at the predicted ball position (*L1* and *L2* in figure 5). The distance between the intermediate point and the ball predicted position *d* is fixed (charging distance). The intermediate position corresponds to the midpoint of the base segment of the equilateral triangle *b*.
- *Follow:* This machine is designed to keep the robot following a target object. The target can be the ball, a team robot or an opponent robot. This is done to

keep the robot nearer to the ball and therefore in a better position to intercept the ball.

- *Homing*: Depending on the strategy being executed robots could be required to be placed at a certain position for the purpose of forming a defense wall for example. The Homing machine performs such actions.

Each of the above machines contains a path planner. This planner generates a path in the form of a straight line between the robot and its target position. After the path has been generated and transmitted the planner keeps track of the robot position. This will enable the remote agent to detect divergence of the robot from its most recent path.

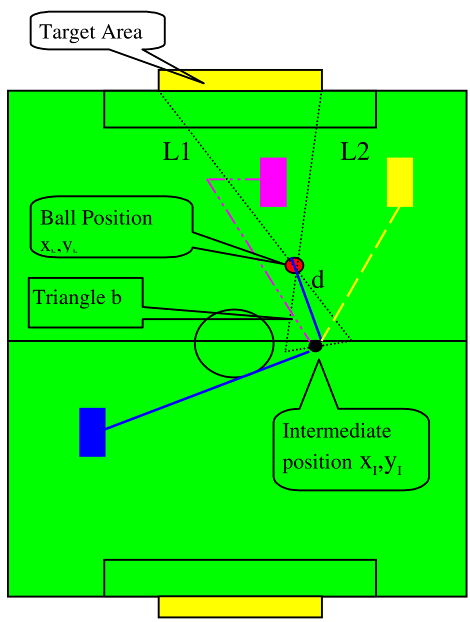

Fig. 5. To intercept the ball the robot needs to position itself in an intermediate position

Divergence of the robot could be caused by a hit from an opponent robot or a purposeful move to avoid an obstacle. In both cases the path planner computes and transmits a new path to the robot.

3.3. Behavior Arbitration

Table 1 shows a list of the behaviors implemented in each robot. The on-board agent consists of a set of reactive behaviors situated in the actual robot environment whereas the remote agents are situated in an abstract environment constructed using the visual data. Embedded behaviors use a competitive coordination mechanism; a fixed prioritization network is used to resolve behavior conflicts. The remote agent uses a supervisory coordination mechanism in the arbitration of its set of behaviors. Each robot is modeled in the following structure:

```
Robot_Player (ID)  (
           Robot_ID,   ;;each robot has an id number.
           Robot_X,    ;;the robot Cartesian coordinate.
           Robot_Y,    ;;
           Robot_Angle, ;;robot direction.
           Robot_Path, ;;most recent path generated.
           Robot _Behavior ;;current robot behavior.
 )
```

Using the Robot–Behavior parameter, the robot could be assigned any of robot remote task, i.e. *Intercept_ball, Follow and Homming*. It is therefore possible that more than one robot would be assigned the same task.

Behavior	Stimulus	Location	Environment	Coordination
Avoid_left	Infrared Sensor	Embedded	Real world –Field	Competitive
Avoid_right	Infrared Sensor	Embedded	Real world –Field	Competitive
Move	RF Channel	Embedded	Real world –Field	Competitive
Turn	RF Channel	Embedded	Real world –Field	Competitive
Intercept_ball	Visual Data	Remote	World Model	Supervisory
Follow	Visual Data	Remote	World Model	Supervisory
Homing	Visual Data	Remote	World Model	Supervisory

Table 1. Robot behaviors distributed among the robot's on-board and off-board agents.

4. Conclusion

The RoboCup small league main challenge to AI professionals is the size limitations. In contrary to the medium size league, very little computational power could be embedded in the robot itself. In this paper we summarize research done in the field of RoboCup and Multi-Agent Systems. Each robot player controller is subdivided into

two agents. On-board agents composed of a collection of survival behaviors and situated in the environment. The remote agents on the other hand are situated in an abstract model of the environment. While remote agents communicate with the embedded agents in a direct manner through the RF link. On-board agents have no explicit communication link with the remote agents. The remote agent's perceived environment is therefore the only source of information on the embedded agent.
This architecture was implemented in Temasek POlytechnic Team (TPOT) during the 1998 Pacific Rim Series RoboCup Competition in Singapore [15] and the 1999 World RoboCup Competition in Sweden [17].

References

1. Fukuda, T. et al. *Structure Decision for Self Organizing Robots Based on Cell Structures- CEBOT*. In Proceeding of the IEEE International Conference on Robotics and Automation. Los Alamitos, Calif, 1998.
2. Arkin, Ronald C. and Tucker Balch. *Cooperative Multi-Agent Robotic Systems. In Artificial Intelligence and Mobile Robots*, ed. David Kortenkamp,278-296. Cambridge, Mass.: The MIT Press.1998.
3. Albus, James S. *Hierarchical Control Of Intelligent Machines Applied to Space Station Telerobots*. IEEE Transactions on Aerospace and Electronic Systems, Vol. 24. NO. 5 Sept. 1988.
4. Housheng Hu and Michael Brady, *A Parallel Processing Architecture for Sensor Based Control of Intelligent Mobile Robots*. Robotics and Autonomous Systems, vol.17,1996, pp 235-257.
5. Brooks, Rodney A.. *Achieving Artificial Intelligence Through Building Robots* A.I.Memo 899, Artificial Intelligence Laboratory, MIT, May 1986.
6. Brooks , Rodney A. *Intelligence without Reason*. Computers and thought, IJCAI-91, Also A.I. Memo 1293 Artificial Intelligence Laboratory, MIT.
7. Maja J. Mataic. *Interaction and Intelligent Behavio.* Ph.D. Thesis, Department of Electrical Engineering And Computer Science, Artificial Intelligence Laboratory, MIT, 1994.
8. Arkin, Ronald C. *The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study*. IEEE Transactions on Systems, Man , and Cybernetics Vol. 20 No 6 Nov./DEC 1990.
9. Kenneth Moorman and Ashwin Ram.. *A Case-Based Approach to Reactive Control for Autonomous Robots*. AAAI Fall Symposium " AI for real time Mobile Robots", Cambridge MA, Cot 1992.
10. Erann Gat , Rajiv, Robert Ivlev, John Loch, and David P. Miller. *Behavior Control for Robotic Exploration of Planetary Surfaces.* IEEE Transactions on Robotics and Automation, Vol. 10,NO4, August 1994.
11. Brooks, Rodney A.. *A Layered Intelligent Control System for a Mobile Robot.* IEEE Journal Robotics And Automation . RA-2, March 1986, 14-23.

12. Manuela Veloso, Peter Stone, Kwun Han, and Sorin Achim. *The CMUnited-97 Small Robot Team. In RoboCup In RoboCup-97: Robot Soccer World Cup 1,* Ed. Hiroaki Kitano, 242-256 LNAI 1395, Spriger-Verlag, 1998.
13. Gordon Wyeth, Berett Browning and Ashley tews. *UQ RoboRoos: Preliminary Design of a Robot Soccer Team.* In Proceedings of RoboCup Workshop.5th Pacific Rim International Conference On Artificial Intelligence. Ed. Hiroaki Kitano, Gerald Seet. Singapore, 1998.
14. Nadir Ould Khessal. *An Autonomous Robot Control Architecture Based on A hybrid behavior arbitration*. In The proceeding of Modeling , Identification and Control Conference. Innsbruck, Austria 1999.
15. Nadir Ould-Khessal, Dominic Kan, Sreedharan Gopalasamy, Lim Hock Beng. *Temasek Polytechnic RoboCup Pacific Rim Series Small League Team.* In Proceedings of RoboCup Workshop.5th Pacific Rim International Conference On Artificial Intelligence. Ed. Hiroaki Kitano, Gerald Seet. Singapore, 1998.
16. Jennings, N. *Cooperation in Industrial Multi-Agent Systems.* Vol 43, World Scientific Press 1994.
17. Nadir Ould Khessal, Dominic Kan, Sreedharan Gopalasamy, Lim Hock Beng. *A Distributed Multi-Agent Architecture for Temasek RoboCup* Team. Proceedings of The Third International Workshop on RoboCup. Ed. Manuela M. Veloso.1999.