COOPERATIVE ROBOTIC SYSTEM USING DISTRIBUTED DECISION MECHANISMS WITH DELIBERATIVE CENTRAL SUPERVISOR*

JULIEN BEAUDRY, RICHARD HURTEAU, RICHARD GOURDEAU

Electrical Engineering Dept., École Polytechnique Montréal, 2900 Édouard-Montpetit Montréal, Québec H3T 1J4, Canada. Email : {julien.beaudry, richard.hurteau, richard.gourdeau}@polymtl.ca

Cooperative multi-robot systems with distributed decision mechanisms and distributed sensing may be the source of decisional conflicts which can lead to severe performance deterioration. A deliberative central supervisor is a simple approach to correct any incoherent decisions in the system. Given an application, the supervisor can be an autonomous software agent or a human-machine interface. Using Hierarchical Decision Machines (HDM) as distributed decision mechanisms, the decision supervision can use simple matrix representations of decisional data. The resulting architecture has been tested on a fully autonomous team of soccer-playing robots and results indicate that it is well adapted for, but not restricted to, the specific needs of autonomous multi-robot systems with real-time distributed sensing and decision taking.

1. Introduction

Following the ascension of behavior-based and hybrid systems [3], many modern robotic system architectures with similarities have been proposed. Architectures such as L-ALLIANCE [12], CAMPOUT [13] or XABSL [9] are good examples, although simpler architectures were used in experimental systems (for example in the Martha project [1]).

Clearly, the use of hybrid systems gives robotic systems a very large spectrum of possibilities: reactive capabilities for deterministic real-time response in highly dynamic environments, but also deliberation capabilities on various timescales for completion of complex tasks. A common way of discretizing level of reactiveness and deliberation in an architecture is by using a hierarchical structure.

The use of communication in a multi-robot system is of particular interest. Depending on the type of system involved, various constraints may apply

* This work is supported by the Institut de recherche d'Hydro-Québec and the Fonds québécois de la recherche sur la nature et les technologies.

concerning availability, bandwidth and reliability of communication protocol. Strict guidelines are therefore considered for the proposed architecture:

- A multi-robot system must be able to implement cooperative behaviors without the use of explicit communication.
- If explicit communication is possible for a given multi-robot system, the system should efficiently benefit of this possibility by augmenting its cooperative capabilities, but minimize the use of bandwidth.

Many other key characteristics have been identified as essential to ensure usefulness of a given architecture: modularity and scalability, multi-platform development, generic level programming and independence to hardware.

2. Multi-robot architecture proposed

It is possible to define a generic control architecture for a single robot as illustrated on figure 1. In a multi-robot system with distributed sensing and decision mechanisms, this architecture is the basic structure of each robot. A multi-robot architecture must then consider this individual level control architecture as its housing for distributed decision mechanisms. Its simple and modular structure allows its implementation with any robotic system and architecture, such as architectures with strong theoretical and experimental background like the 4D/RCS architecture [2]. Additional modules like intermediate reactive controllers or communication modules can be fitted to the generic architecture.

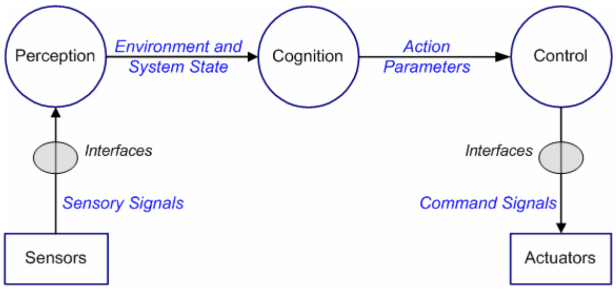

Figure 1. Generic robot level control architecture.

Three major modules are considered: *Perception*, *Cognition* and *Control*. The *Cognition* module is the core of the intelligent system. All the information of the system can be accessed and interpreted by this module, direct link between *Perception* and *Control* are therefore disallowed. The decision mechanisms are enclosed within the *Cognition* module.

2.1. *Distributed decision mechanisms*

The proposed decision mechanism is the *Hierarchical Decision Machine* (*HDM*) [4], a structural representation similarly implemented in other architectures [8], [11]. This decision machine uses a hierarchical structure and works as a succession of sequential decision mechanisms. The graphical representation of the HDM and its decision mechanism are illustrated in figure 2. XABSL [9] is a good example of similar architecture taking profit of this simple representation. The HDM is implemented in object-oriented C++. A given machine may be defined by an arbitrary number of hierarchies, each one containing an arbitrary number of *Decision_node*s. The succession of decisions, called the *Decision Line*, is always terminated by the selection of a *Behaviour_node*, which updates various *Action_parameters* in order to activate specific *Action*s of the set offered by the multi-robot system. Each type of node has access to useful information, like robot internal states and an appropriate model of its dynamic environment.

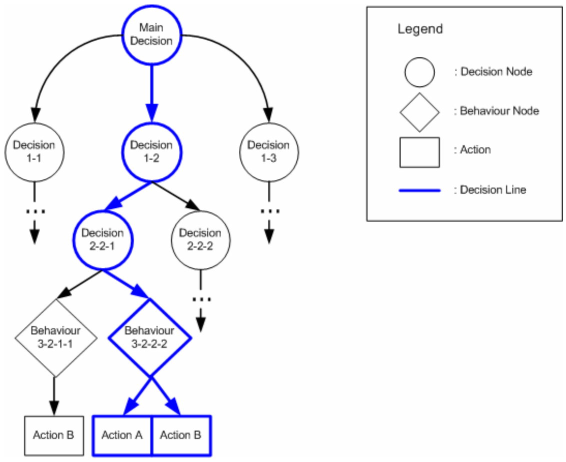

Figure 2. Graphical representation of the Hierarchical Decision Machine.

The HDM is designed with the objective of being distributed on every robot of a multi-robot system. In order to introduce cooperative behaviours between robots, pre-established agreements can be used on each decision machine, similar to locker-room agreement concept [16]. These agreements can implement various cooperative capabilities like resource sharing or dynamic role allocation.

2.2. *Deliberative decision supervision*

As a design characteristic, a multi-robot system must be able to cooperate without the use of external help. It is however possible that such system can be a source of decision conflicts between robots, in particular if perception capabilities are also distributed. In this case, instead of using distributed techniques which can still yield to conflicts, such as negotiation [6] or utility functions [5], the concept of *Decision Supervision* is introduced. A central deliberative process can control the ongoing distributed decision mechanisms on the complete hierarchy of a given decision machine and on every robot of the system.

Technically, the supervisor is a central element connected to every robot of the system. Using a client/server approach, it can centralize and redistribute any pertinent data. With the HDM simple structure, a *Decision Line* for a given robot can be represented as a *Decision Vector* (V_{D_i}) of length equal to the number of hierarchies in the decision machine. The supervisor can retrieve each *Decision Vector* of the system and form the *Decision Matrix* (M_D). These data structures can be defined as follows:

$$V_{D_i} = \begin{bmatrix} D_{i-1} & D_{i-2} & \dots & D_{i-N_D} \end{bmatrix}, \quad M_D = \begin{bmatrix} D_{1-1} & D_{1-2} & \cdots & D_{1-N_D} \\ D_{2-1} & D_{2-2} & \cdots & D_{2-N_D} \\ \vdots & \vdots & \vdots & \vdots \\ D_{N_R-1} & D_{N_R-2} & \cdots & D_{N_R-N_D} \end{bmatrix} \quad (1)$$

With this matrix obtained and updated using communication with robots, the supervisor can define a *Supervision Matrix* (M_S) composed of *Supervision Vectors* (V_{S_i}). These variables can be defined in a similar way:

$$V_{S_i} = \begin{bmatrix} S_{i-1} & S_{i-2} & \dots & S_{i-N_D} \end{bmatrix}, \quad M_S = \begin{bmatrix} S_{1-1} & S_{1-2} & \cdots & S_{1-N_D} \\ S_{2-1} & S_{2-2} & \cdots & S_{2-N_D} \\ \vdots & \vdots & \vdots & \vdots \\ S_{N_R-1} & S_{N_R-2} & \cdots & S_{N_R-N_D} \end{bmatrix} \quad (2)$$

When a robot receives its *Supervision Vector* it can respond, deliberately, to supervision order. Using such simple representation, conflicts can be rapidly addressed and easily solved. Although this scheme requires explicit communication with every robot of the system, the necessary bandwidth is kept minimal with simple numerical values (integers) exchanged. Still, control of decisional data update rate must be possible.

3. Test bench: autonomous soccer-playing robots

A team of fully autonomous and cooperative soccer-playing robots has been used to test and validate the proposed decisional architecture. The environment in which the robots operate is highly dynamic, adversarial and offers unpredictable characteristics.

3.1. *Description of the experimental platform*

The multi-robot system used as a test bench consists of a total of six soccer-playing robots [15]. The team of robots, shown in figure 3, is conforming to the rules of the Middle Size Robot League (MSL) of the RoboCup World Championship [14]. Each robot is fully autonomous, contains an embedded computer, an omnidirectional vision system [10], a wireless LAN connection, and appropriate electromechanical devices.

Figure 3. Team of autonomous soccer-playing robots.

A *teamserver* is connected to every robot to act as the decision supervisor. The client/server capabilities are part of MICROB [7], a C++ robotic library used as a valuable toolbox in the software architecture.

3.2. *HDM developed for a team of soccer-playing robots*

The decision machine developed for the soccer-playing robots had to respond to two specific needs, the cooperative team play and individual skills for the game itself, but also conform to all the rules of the MSL. The overall view of the decision machine is shown in figure 4. Even if this graph has been simplified, it can be seen that having a view of a complete decision machine is fastidious. Studying particular segments of a machine is more appropriate.

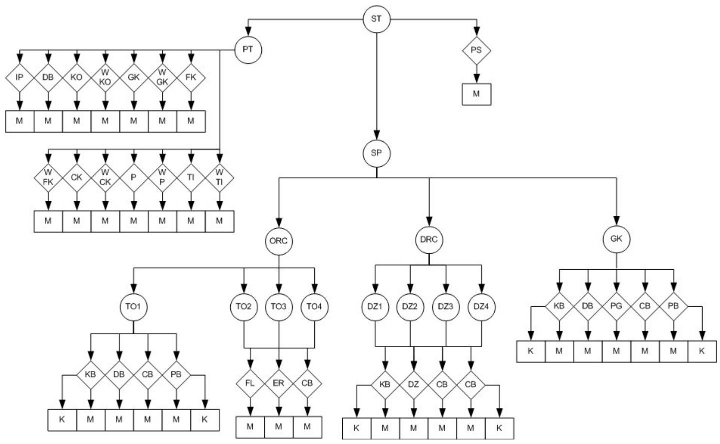

Figure 4. Overall view of a HDM developed for soccer-playing robots.

This decision machine presents a five level hierarchy: *Mode*, *Pattern*, *Role*, *Behaviour* and *Action*. The machine is composed of 128 individual nodes. But, using the object-oriented implementation of the HDM, only 21 different basic nodes have been used, since many are reutilized.

During game play situation, the mode *SoccerPlayer* allows the team of robots to cooperatively play soccer. Basically, there are offensive and defensive patterns, dynamically selected depending on game situation. The pre-established rule can be as simple as:

- if the ball is in defensive zone, defensive pattern is selected
- otherwise, offensive pattern is selected

This is an example of simple pre-established rule for cooperative team play. Such simple rule can be the source of decision conflicts, what happens when the ball is in the center of the field? Similar but more elaborate rules are often needed to minimize source of conflicts.

Offensive and defensive patterns use formations with predefined characteristic. For example, the *OffenseRC* pattern uses a dynamic role allocation procedure to implement an offensive formation where each robot selects an appropriate role. An example of such formation with its corresponding decision mechanism is showed on figure 5. Dynamic role allocation here is based on relative positions of robots to ball and opponents goal.

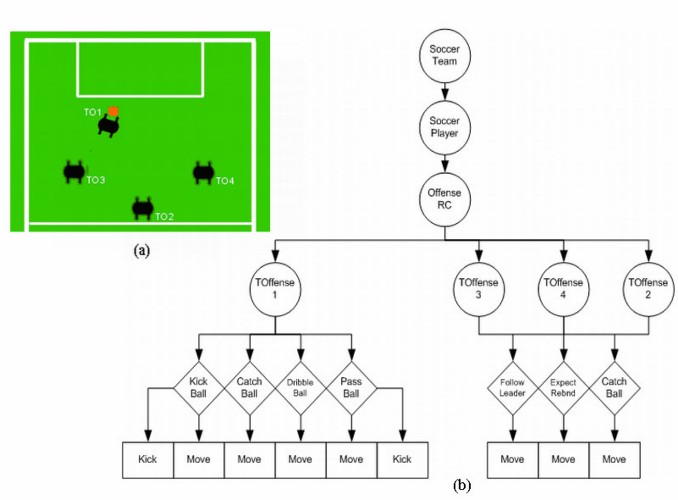

Figure 5. Dynamic offensive formation (a) and corresponding decision machine (b).

3.3. *Decision supervision for this type of multi-robot system*

Decision supervision for this system had to respond to three specific needs:

1. Accurately respond to referee calls
2. Supervise the dynamic selection of the same pattern by each teammate
3. Supervise the selection of an exclusive role by each teammate

Each robot's *Decision Vector* is composed of four elements corresponding to decision machine hierarchy. With a team of five robots a twenty elements *Decision Matrix* is obtained. Consequently, the *Supervision Matrix* contains twenty elements. The dynamic role allocation for offensive pattern (see figure 5) can illustrate the supervision mechanism. When the team is in the offensive pattern, players must select an exclusive role. A *Decision Matrix* showing a conflict for the role selection and a resulting *Supervision Matrix* could be:

$$M_D = \begin{bmatrix} 0 & 0 & X & X \\ 0 & 2 & \mathbf{1} & X \\ 0 & 2 & \mathbf{1} & X \\ 0 & 2 & 3 & X \\ 0 & 2 & 4 & X \end{bmatrix}, \quad M_S = \begin{bmatrix} 0 & -1 & -1 & -1 \\ 0 & 2 & \mathbf{2} & -1 \\ 0 & 2 & \mathbf{1} & -1 \\ 0 & 2 & 3 & -1 \\ 0 & 2 & 4 & -1 \end{bmatrix} \qquad (3)$$

Response to supervision in the robots is done by a parallel thread managing communication with the server. In other words, the robot does not count on this communication, but it could receive a useful *Supervision Vector*. It is possible to adjust the update rate of the supervision mechanism and bandwidth of communicated data can therefore be controlled.

4. Results

Using the already existing soccer-playing robots hardware and software modules, every needed functionality for the multi-robot system has been rapidly developed. Enabling intuitive programming of decision machines is a major benefit of the proposed architecture. More precise performance analysis and results are presented in next paragraphs.

4.1. *Resource usage*

The architecture is to be used in deterministic real-time and also scalable robotic systems. Resource usage must therefore be kept minimal. CPU and memory usage has been measured on experimental robots. The CPU usage on the robots has been measured to a mean of 5.2% on Pentium III 800MHz processors and 7.6% on Celeron 566MHz. Plenty of CPU is left for vision algorithms. However, memory usage is important for a relatively small HDM, with 13.36MB used on robots. Duplication of nodes is responsible for this result.

Time of completion for decision mechanisms has also been measured. This time can vary depending on the active decision line. Measurements are given in table 1. Completion time is always kept below 200us which allows for fast and deterministic control loops.

Table 1. Completion time of decision mechanism measured on experimental robots.

Computer configuration	Min. time (us)	Max. time (us)	Avg. time (us)
Pentium III LP 800MHz	107	135	120
Celeron 566MHz	145	172	158

4.2. *Decision supervision relevance*

Measurements have showed that even with meticulously defined cooperative rules, the HDM developed was the source of decision conflicts. Tests have been made considering a two minutes period where dynamic role allocation is measured for the offensive pattern *OffenseRC*. In this pattern, if the same role is used by two robots at the same time a conflict is detected. Figure 6 shows the dynamic role allocation of the four field players without (*a*) and with (*b*) supervision. According to these results, decision conflicts occurring using the cooperation agreements can be reduced from a near 45% to a level of approximately 2.5% of the total working time. This two minutes test considers a short period of time and variability is important, but every other test executed showed similar results.

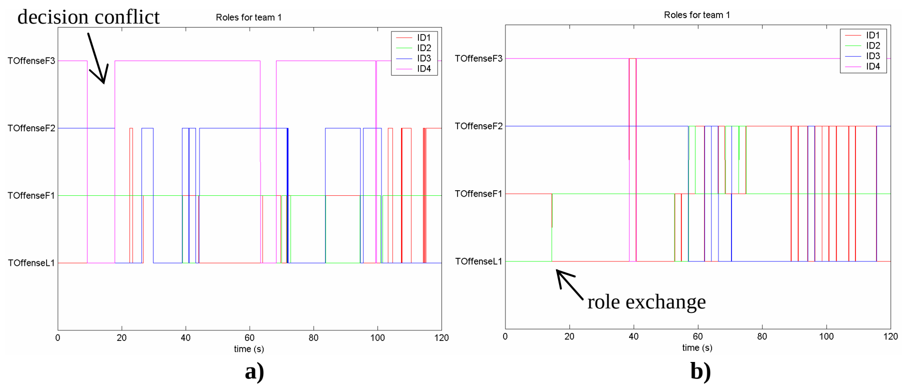

Figure 6. Dynamic role allocation without (*a*) and with (*b*) supervision.

Performance increase in a given system may not be directly measured with decision supervision enabled, but other benefits should also appear. For soccer-playing robots, conflicts can be the source of collisions and other perceptible problems leading to hardware problems. Furthermore, conflicts can increase robots energy consumption and compromise consistent strategic team play.

5. Future Work

The project presented in this paper established some basic elements for multi-robot systems with distributed sensing and decision mechanisms. These elements should serve as building blocks for more elaborate work on cooperative multi-robot systems. A combination of central and distributed deliberation is probably the most powerful approach in terms of global intelligence, robustness and efficiency of a system. More work should be done to demonstrate this point.

An ongoing project that uses the proposed architecture concerns generic learning methods for hierarchical multi-robot systems. Soccer-playing robots are again used as a test bench. Development of graphical possibilities of the HDM and dynamically modified HDMs are examples of future developments.

References

1. Alami R., Fleury S., Herrb M., Ingrand F., Robert F.. *Multi Robot Cooperation in the Martha Project.* In IEEE Robotics and Automation Magazine, Vol. 5, No. 1. IEEE. 1997, pp.36-45.
2. Albus J.S. et al. *4D/RCS: A Reference Model Architecture for Unmanned Vehicle Systems Version 2.0*. NISTIR 6910, National Institute of Standards and Technology. USA. 2002.
3. Arkin R.C.. *Behavior-Based Robotics*. The MIT Press, Cambridge. 1998.

4. Beaudry J.. *Machine décisionnelle pour systèmes multi-robots à perception distribuée*. M.Sc.A. thesis, Electrical Engineering Department, École Polytechnique de Montréal. 2005.
5. Chainowicz L., M.F.M. Campos M.F.M., Kumar V.. *Dynamic Role Assignment for Cooperative Robots*. Proceedings of the 2002 IEEE International Conference on Robotics & Automation. 2002.
6. Emery R., Sikorsky K., Balch T.. *Protocols for Collaboration, Coordination and Dynamic Role Assignment in a Robot Team*. Proceedings of the 2002 IEEE International Conference on Robotics & Automation, 2002, pp.3008-3015.
7. Houde R., Blain M., Côté J.. *Manuel de l'usager pour Microb*. Internal report IREQ-2000-075, Institut de recherche d'Hydro-Québec, 2000.
8. L'Archevêque R., Dupuis E.. *Autonomous Robotics and Ground Operations*. Proceeding of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS 2003. Japan. 2003.
9. Lötzsch M., J. Bach J., Burkhard H.-D., Jüngel M.. *Designing Agent Behavior with the Extensible Agent Behavior Specification Language XABSL*. In 7th International Workshop on RoboCup 2003, Lecture Notes in Artificial Intelligence, Padova, Italy. 2003.
10. Marleau S.. *Système embarquée de localisation et de perception pour un robot mobile*. M.Sc.A. thesis, Electrical Engineering Department, École Polytechnique de Montréal. 2005.
11. Nesnas I.A., Wright A., Bajracharya M., Simmons R., Estlin T., Kim W.S.. *CLARAty: An Architecture for Reusable Robotic Software*. SPIE Aerosense Conference, 2003, pp.121-132.
12. Parker, L.E.. *ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation*. IEEE Transactions on Robotics and Automation, 1998, pp.220-240.
13. Pirjanian P., Huntsberger T.L., Trebi-Ollennu A., Aghazarian H., Das H., Joshi S., Schenker P.S.. *CAMPOUT: A control architecture for multi-robot planetary outposts.* In Proceedings of the SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III, Vol. 4196, Boston, MA, Nov. 2000.
14. The RoboCup Federation. *RoboCup Official Site*. The RoboCup Federation, online, April 2006: http://www.robocup.org.
15. Robofoot ÉPM. *Robofoot ÉPM Official Site*. Robofoot ÉPM, online, April 2006: http://robofoot.polymtl.ca.
16. Stone P.. *Layered Learning in Multiagent Systems*. The MIT Press, Cambridge. 2000.