Extension of the Behaviour Oriented Commands (BOC) Model for the Design of a Team of Soccer Players Robots

C. Moreno[1] – A. Suarez[1] – E. Gonzalez[1] – Y. Amirat[2] – H. Loaiza[3]

Université d'Evry – LaMI – Bd François Mitterrand 9100 Evry France[1]
{moreno, suarez, gonzalez}@lami.univ–evry.fr
Université Paris XII – LIIA – 122, Rue P. Armangot Vitry S/Seine France[2]
amirat@univ–paris12.fr
Universidad del Valle –PayRA – Ciudad Universitaria – Cali – Colombia[3]
hloaiza@eeie.univalle.edu.co

Abstract. The Real Magicol soccer players team for Robot Cup 99 is based on a BOC architecture extension which combines reactive and deliberative reasoning by the distribution of the knowledge system into modules called behaviors. The hardware remains the same as the 1998 Robot Cup one, our research bearing on the cooperative architecture based on agent concept.

1 Introduction

Soccer players constitute a complex application of collective robotics. In fact, in such a context, the control architecture of a robot grows in complexity since this robot has to evolve in dynamic environment and cooperate with its partners in order to develop a team game. Such an application has to be considered as a set of decision–making, processing and communications, rather than a centralised and monolithic set, where little room is left to the autonomy and little importance is granted to interactions between the different robots. This agent oriented approach allows a redistribution of the decision–making among the different levels of the organisation leading consequently to a distributed, dynamical and reactive organisation.

The control architecture of a soccer player robot constitutes a complex system. This complexity is due to the necessity to make coexist both mechanisms of knowledge management (deliberative actions), and constraints of reactivity to "survive" in a dynamic environment. To process this problem, several architectures have been proposed in mobile robotics.

Deliberative architectures [MORAVEC 89] [CHATILA 85] use a centralised model of the environment to verify sensory information and to find the best response in the physical world. All actions of the robot are directed to a known final goal. These architectures allow to endow the robot with reasoning capacities that confer it the

M. Veloso, E. Pagello, and H. Kitano (Eds.): RoboCup-99, LNAI 1856, pp. 390–399, 2000.

function of planning.
Reactive architectures [BROOKS 86] [CONNEL 90] are interesting to study the emergent behaviour of the robot from its primitive behaviours. On the contrary of deliberative architectures, these ones neither allow the planning of operations nor authorise elaborate reasoning. These architectures use a set of modules of behaviours that react directly to variations of the environment by using simple transfer functions. Consequently, need of a model of the world disappears.
Hybrid architectures [SIMMONS] offers a compromise between the two previously described approaches to be both reactive and capable to follow a plan.
We propose in this paper the concepts that are going to allow the modelization of an organisation of soccer playing robots from both of a local point of view (the agent robot component) and a global point of view (interactions between 'robot' agents). The objective is to propose a control architecture model for each robot, that includes a both deliberative and reactive dimension, in order to take into account, on the one hand, its interactions with the environment and on the other hand, its interactions with its partners to make emerge a strategy of group. These works concern the participation of the team "Real Magicol" (Realismo Mágico Colombiano [Realismo Màgico] to the RoboCup–99 in the category " middle size ". The hardware architecture as well as general concepts of control architecture implemented by this team have been described in a communication presented during the RoboCup–98 WorkShop [LOAIZA 98]. Given the amount of constraints of soccer playing robots, we have proposed an architecture of control based on the Behaviour–Oriented Commands (B.O.C) concept [CUERVO 96]. This architecture confers to the robot both deliberative (in order to reason on complex situations) and reactive capacities (in order to respect deadlines). More specifically, it relies on a decomposition in competitive modules which interaction allows to endow the robot of typical reactive, deliberative or hybrid behaviours.
In this paper, we present results of works started on the extension of the B.O.C model in order to endow each robot with a supplementary deliberative dimension oriented towards the group strategy. This is necessary in the case of a soccer playing robots since it allows to endow each robot with a motivation going in the senses of the satisfaction of a collective tendency. Consequently, it allows through collaboration and co–operation mechanisms between robots to make emerge a collective game at the organisation level. In the 2nd. paragraph, we introduce notions of classification of roles to define an organisational model of the team. The objective being to make emerge at the organisation level behaviours with collective tendency. We present in the paragraph 3, the extension of the BOC model and particularly deliberative behaviour units "individual strategist" and "strategist of group". The paragraph 4 describes collaboration and co–operation mechanisms between robots, used for the implementation of the distributed B.O.C. We have also introduced at the goalkeeper control architecture a BlackBoard cartographer to provide relevant information, used in the game strategy. The objective being to guarantee a temporal coherence when deliberative behaviour units oriented to the strategy of group are activated.

2 Emergent collective behaviours and organisational model

Each of robots playing on the field can be considered as an agent participating to the strategy of game. Indeed, if a game strategy can be assimilated to the convergence of a set of actions to a goal or to several ones, it is then necessary to decentralise and assign each action to each member of the team and to establish a collaboration between them. Thus, the emergent collective behaviour results from the sum of local agents behaviours.
To obtain a coherent collective behaviour, it is essential to well co–ordinate 'robot' agents. Each has to have a predetermined role to avoid a divergence of the consequent of players actions. More, the efficiency of co–operation of players results also from their good synchronisation. Without this synchronisation, the collective behaviour is inoperative, and consequently, the strategy of group is non–existent. Indeed, because of the uncertainty of the "freshness" of information from others, each robot has then to evolve individually and to act in autonomous manner. If this determinism is absent at the individual level (robot level), the realisation of a behavioural model is made impossible. Thus, behaviour units work stand by themselves and no emergence arise. To reach these objectives, a real time deterministic system managing communications is necessary.
According to the previous discussion, the fact that the global behaviour of robots is the consequent of local behaviours of each one implies a good distribution of roles to each robot.

Classification of roles, notion of formation
The chosen classification is in relation with a division of the field in zones. Each robot is assigned to a zone of activity, they are told in formation. Here is an example:
- The goalkeeper, number 1 on the figure 1, rest in the zone of goal of the team.

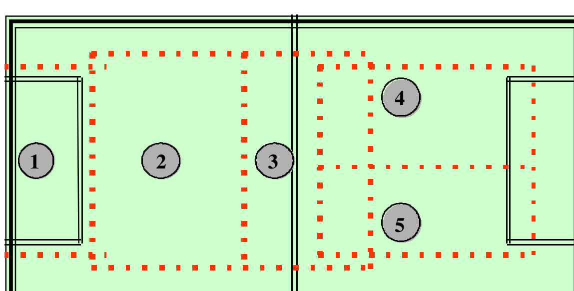

Fig. 1. The team formation

- The two defenders, numbers 2 and 3 on the figure 1, are in two distinct zones, sharing transversely half of field.

- The two attackers, numbers 4 and 5 on the figure 1, are also in two distinct zones, sharing lengthways half of field from adverse side.

Note that these zones may have a non empty intersection. Moreover, formations may be subdivided into sub formations. Each one includes then a subset of roles and interactions between these roles. Although that in a given time the robot performs only one role, it has a knowledge of all roles of the team. This allows the organisation to evolve according to two different approaches:

- The dynamic change of the role. For example, robots A and B may change their roles as well as their positions on the field if the robot A has to follow the ball in the zone of the robot B.
- The dynamic change of the formation, by taking into account the remaining time as well as the current result of the game. All the team may change its strategy leading to a more offensive or more defensive formation.

Each robot can choose between three action modes:

- It remains in its pre–defined position (the inactive state)
- It moves to the ball by tempting to shoot it to the adversary goal or, to pass it to its partner (the active state)
- It tries to intercept the ball that displaces to the goal (the auxiliary state).

These three modes of action define different behaviours according to the environment and the strategy under way of execution. Two types of behaviour can then be extracted:

- A purely opportunist behaviour, consisting in a direct answer to stimuli. This behaviour follows the reactive model.
- An intelligent behaviour computing a plan of actions in order to score at the adverse goal. This behaviour follows the deliberative model .

We present hereafter the B.O.C. model for the implementation of the control architecture of soccer playing robots.

3. BOC Architecture (Behaviour Oriented Commands)

3.1. The BOC Model

The BOC architecture is a hybrid architecture which combines reactive and deliberative reasoning by the distribution of the knowledge system into modules called behaviours. An important feature of this model is that it is easily and directly translated into a real time application.

Our architecture supplies independent entities capable of reacting directly to stimuli and also endowed with an inference system allowing deliberative planning. The knowledge of each module is encapsulated in a set of rules describing its desired behaviour. The reasoning module of each behaviour appeals to a production rules system (expert system of order 0+ or a finite state automaton) that allows to describe in a simple way complex behaviours.

Our design methodology is supported by two classical notions: abstraction and decomposition. Abstraction allows us to define a general commands without knowing its implementation while decomposition permits to map a complex set of rules into a group of less complex entities. This decomposition of the system allows the simultaneous representation of both the temporal evolution and the parallel relation of the treatments. The knowledge of the robot is represented on two levels:

- The encapsulation of a set of related rules into independent entities called behaviours.
- The establishment of association links between these entities, in order to carry out a behaviour–oriented–command (BOC) by the execution of co–operative actions.

A BOC is like a service that must be requested, executed and acquitted. It is carried out by a set of associated behaviours which can execute at the same time. Each behaviour groups a set of rules in order to achieve and maintain its own goal. It can be seen as an agent whose actions are either direct interactions with the sensors and actuators or requested services to other BOCs. The co–operative work of these Associated Behaviours (ABs) allows to solve problems beyond the scope of each one independently.

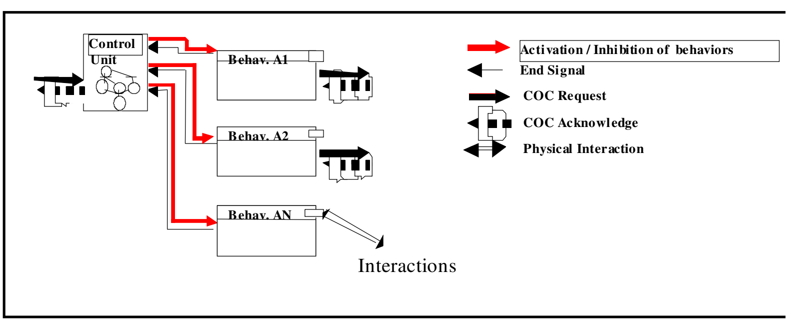

Fig. 2. Representation of a COC

A BOC is composed of a control unit and a set of associated connected behaviours which carry out a service respecting a specific set of co–ordination rules. The control unit associated to a BOC (fig. 2) includes an input port for the BOC's request, an output port for the BOC's acknowledgement and three control links between the BOC and its associated behaviours (AB): activation, inhibition and end signals.

These signals allow the co–ordination and synchronisation of the ABs. Activation signals are used to activate the necessary ABs for the correct execution of the requested BOC. End signals are sent from an AB to its BOC's control unit when it finds a BOC ending condition. Once the control unit receives the end signal, it sends inhibition signals to all the ABs which were activated. Finally, the control unit sends an acknowledgement to the behaviour that requested the BOC.

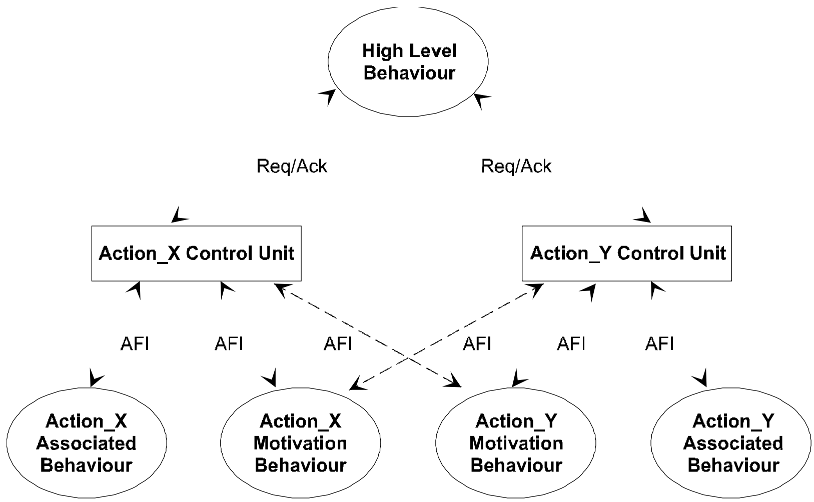

Fig. 3. Managing of concurrent actions using COC architecture

Figure 3 illustrates how the BOC architecture is used to manage two concurrent actions. Each action is modelled as a BOC which is carried out by a set of ABs. Generally, there is at least one behaviour that verifies that the conditions required to continue the execution of the BOC are valid. In a decision system, this behaviour can be seen as an entity monitoring the motivation of the action. Motivation allows to manage in a simple way the activation of concurrent actions (solid lines fig. 3).
When the motivation decreases the behaviour sends an end signal to the control unit to indicate that the action must be stopped. When the end signal arrives, the BOC is finished and acquitted. A behaviour of higher level receives this acknowledgement and uses its reasoning system to determine the next action to perform. The rules describing the behaviour and co–ordination of activities are distributed into modules, each one with a specific role.
In order to improve the performance of our decision system, the unit control of a BOC can also activate the motivation behaviour of a concurrent BOC (dotted lines in figure 3). This behaviour measures the relevance of starting the concurrent action, thus acting as an anti–motivation monitoring entity. A behaviour used as an anti–motivation sends an end signal when the motivation to perform its associated action is high enough. The trigger values are dynamically set by the high level behaviour and include an hysteresis to avoid undesirable oscillations when motivation is close to the transition value. Remark that since concurrent control units are never simultaneously active, a motivation behaviour will never be working at the same time for different control units.
Our robots use the BOC model. As a matter of fact, this model allows to easily separate into different units the robot's behaviours. It is then possible to construct the set of roles the robot can play by considering them as behaviours. For instance the attackers can be considered as an aggressive reactive behaviour and an

aggressive deliberative behaviour. For instance, when the ball is found in position allowing for goal kicking, motivation of the aggressive reactive behaviour will be at its highest. On the contrary, when the ball interception is found to be impossible without obstacle avoidance the motivation of the deliberative behaviour will take over.

Currently BOCs are implemented as fully reentrant functions. Each BOC shares the source code in charge of the real time issues (communication, synchronisation, priority) allowing the programmer to concentrate in the behaviours themselves. In order to achieve even more transparency we are currently planning a future fully object oriented BOC version.

3.2. Reactive and Deliberative BOCs

The game tactics are implemented as a set of behaviours of different types.

Reflex Behaviours

This behaviours will accomplish very simple actions, as a direct response to a reduced set of pre–recorded stimuli. This allows fast "reflexive" actions. By definition, reflexive actions are simple, use only the most recently available information and have small, unambiguous rules.

Reflex behaviours have very high motivation values when their pre–conditions are met. Motivation however decreases rapidly as the expected condition is not found, allowing other more deliberative behaviours to be considered.

Deliberative Behaviours

These behaviours execute actions with a significant degree of complexity (plans). This includes a long set of movements which should finish in a better global position of each robot. No negotiation between the robots (like which one should go for an equidistant ball) has been implemented until now. Communication and synchronisation are performed through the environmental perception. Team play is mainly achieved through coherent motivation equations.

3.3. Strategy

The different roles and sub–roles are also a result of behaviour motivation evaluations.

Attack Behaviour

This is the aggressive robot’s personality. Its goal is to find and keep track of kick solution.

Defence Behaviour
This is a more conservative behaviour. It looks for a backup or ball recovering position. This is achieved by trying to fill the gaps while remaining behind the ball and adverse robots.

Individual Strategy Behaviour
This is the highest level behaviour inside any single robot. It is charged of maintaining the robots global behaviour. It will therefore evaluate the motivations to play defensively or aggressively. This behaviour is activated at the beginning of the game with the parameters of initial robots position and playing attitude.

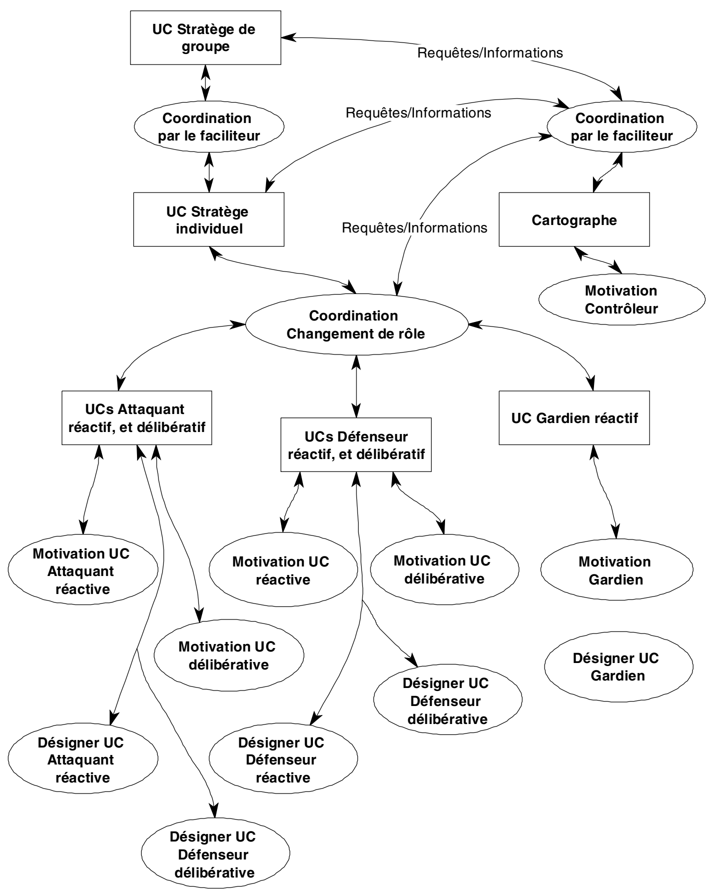

Fig. 4. C.O.C. model of each robot's role : Attacker – Defender – Keeper

Group Strategy Behaviour

This behaviour is in charge of collective play. It is the root behaviour. As such, it will start the individual strategy behaviour of all robots. It is also in charge of stopping any or all robots if necessary (dysfunction, goals, end of match…). The physical location of this behaviour can be a fixed post (used also for supervision) or any of the robots (actually the goalkeeper which has a much simpler behaviour hierarchy). Due to the network transparency of the real time kernel chosen (QNX), this can be done with no extra code. However, the physical position of this behaviour cannot be changed after starting. We intend to pass through this behaviour in order to improve the role swap of robots.

The Cartography Behaviour

Each robot saves the information gathered from different sources (camera, odometer, communications) and keeps track of a local model of the world. This allows to calculate object's future positions and to continue to evolve even when an object is not directly in sight.

A global cartography is also constructed, which takes into account information produced by several different robots. This model has to take into account local uncertainties of the different robots as well as temporal aspects. It is still in development and therefore unused for any strategic purpose. A blackboard approach is used which must fusion:

- The ball position
- The adverse goal position (Although the goal itself is a static object, we've created a dynamic goal which corresponds to the biggest goal acces and "moves away" from the opponent's goalkeeper.
- The team robots positions
- The opponents robots positions

The Blackboard control unit chooses a map building domain related with the strategic needs. Hence, if the ball's co–ordinates are necessary and a priority, they will be the first chosen domain. To illustrate the control unit's functioning let's consider the following instance :

The ball's co–ordinates belong to the designated domain, the control must choose the knowledge sources enabling it to determine them:

- The data coming from the vision modules directly from each robot;
- The data coming from the vision modules from each robot associated to make a triangulation ;
- The former ball's co–ordinates if none of the robots see the ball.

The control unit then determines the most precise source, which is to say the information coming from the closest robot to the ball or a triangulation constituted thanks to information issued from several robots. The triangulation may be chosen if the information coming from the robots is synchronous during a short time interval determined before hand according to the maximum estimated ball speed, for instance.

4. Conclusion

We have presented in this article an extension to the BOC model. This allows to endow each robot with a deliberative dimension oriented towards group strategy, while maintaining reactivity. One of our main objectives is to study and improve the emergent collective behaviours.
The notions of role and formation as well as a role classification are introduced. Defensive, offensive and individual strategy behaviours are oriented towards the role concept with a group strategy behaviour co–ordinating the whole. The entire architecture is implemented within the BOC paradigm. We currently work in an object oriented BOC implementation.

Acknoledgement

Vincent & Jean Moritz for their inconditional cooperation.

References

[BROOKS 1986]	R. A. BROOKS, A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation,* Vol. 2, No. 1, pp 535–539, 1986.
[CHATILA 1985]	R. CHATILA, J. LAUMOND, Position Referencing and consistent World Modeling for Mobile Robots, *IEEE ICRA,* 4 pp 138–145, 1985.
[CONNEL 1990]	J.H. CONNEL. A Colony Architecture for an Artificial Creature, *MIT A.I. lab Technical* Report 1151, 1990.
[CUERVO 96]	J. CUERVO, E. GONZALEZ, A. SUAREZ, C. MORENO, Behavior–Oriented Commands: From Distributed Kwoledge Representation to Real Time Implementation, *Proc.Euromicro Workshop on Real–Time Systems*, pp. 151–156, Jun. 1996.
[LOAIZA 98]	H. LOAIZA, A. SUAREZ, E. GONZALEZ, S. LELANDAIS, C. MORENO, Real MagiCol: Complex Behaviour througth simpler Behaviour Oriented Commands, *Proceedings of the second RobotCup Workshop*. pp. 475–482, Paris 1998.
[MORAVEC 1989]	H. MORAVEC. D. CHO., A Bayesian Method for Certainty Grids, *Proceedings AAAI Spring Symposium on Robot Navigation,* pp 57–60, 1989 .
[Realismo Màgico]	http://artcon.rutgers.edu/artists/magicrealism/magic.html
[SIMMONS 1994]	R SIMMONS, Structured Control for Autonomous Robots, *IEEE Trans. on Rob. and Automation,* Vol. 10, No. 1, 1994.