A Hybrid Agent Model, Mixing Short Term and Long Term Memory Abilities

An Application to RoboCup Competition

Fausto Torterolo[1] and Catherine Garbay[2]

[1] DIE, Dipartimento di Ingegneria Elettrica, Universitá di Palermo Viale delle Scienze, 90128 Palermo, Italy.
[2] TIMC/IMAG, Institut Albert Bonniot, Domaine de La Merci 38706 La Tronche Cedex, France.

Abstract. We present in this paper a novel approach for the modeling of agents able to react and reason under highly dynamic environments. A hybrid agent architecture is described, which allows to integrate the capacity to react rapidly to instantaneous changes in the environment with the capacity to reason more thoroughly about perceptions and actions. These capacities are implemented as independent processes running concurrently, and exploiting different memorizing abilities. Only a short-term memory is made available to reactive agents, whilst long-term memorizing abilities together with the possibility to reason about incomplete information is provided to cognitive agents. This model is currently experimented and tested under the framework of the RoboCup competition. An application example is provided to support the discussion.

1 Introduction

The recent development of robotic soccer competition has resulted in the designing of a variety of player models and architectures. Central to these developments is the specification of adapted, robust and efficient perception-decision-action cycle. Pure cognitive agent architectures have been proposed, like in Gaglio [3], where perception is rather approached in terms of static representations. Reactive agent architectures have been conversely designed, like the subsumption architecture of Brooks [1], where the emphasis on the contrary is on the notion of behavior. Most of these approaches, however, fail to consider the specificity of reasoning under dynamic environments, when the sensory information can be imprecise, uncertain and incomplete. Indeed, it has to be considered that the degree of veracity of sensory information decreases rapidly with time, since the environment is continuously evolving : consequently, what one agent sees at one time-instant may be not valid at the next time-instant. Most environments in the real world are of this kind, and it appears difficult to perform reasoning and interpretation with standard artificial intelligence techniques. The problem of dynamic environment was for example investigated in the path-planning domain where the changing of the environment with time has been shown to transform

M. Asada and H. Kitano (Eds.): RoboCup-98, LNAI 1604, pp. 246-260, 1999.
© Springer-Verlag Heidelberg Berlin 1999

the correctly found plan to something totally unusable [5]. To solve this specific problem some authors have tried to model the incertitude of the information by stochastic methods. Also to be considered, and even a more complex problem, is the incompleteness of sensory information, that makes the agent uncertain about its current world model, a difficulty that is rarely tackled in front [12]. By incompleteness, we mean the fact that only some part of the world is perceived at each time by a given agent, this perception being in turn subject to incompleteness, depending on the distance and occlusions between objects. We present in this paper an approach to this problem, which has been developed in the framework of RoboCup, a competition between multi-agent soccer systems. In the soccer domain, the agents have different and contrasting goals. The agents in a team are supposed to act in a collaborative way, while the two teams compete together. They evolve in a highly dynamic environment comprising the ball, the other agents, and a few static elements like the goal-area. Different types of challenge are the objectives of this competition, as debated by Itsuiki et alt. in [10]. Our work takes place in the simulation section, more precisely in the cooperative and opponent-modeling challenge. Agents are simulations of robots in the soccer domain. They have limited sensory abilities and a small visioning cone. This application therefore appears as an excellent domain for the study and development of new reasoning models able to cope with the dynamicity and incompleteness of information. This kind of problem was already investigated In the work of Stone et al. [13], but to a limited extent. We have extended and deepened this approach by analyzing (i) how to represent and maintain knowledge about the world and (ii) how to use this knowledge in presence of incomplete information, in order to increase their completeness. Specific inference schemes have been developed for this purpose. Also specific to our approach is the definition of (i) a reference system to simplify the complexity of the computation, and (ii) a short term and a long term memory (respectively ST_memory and LT_memory). A hybrid agent model is finally proposed, which allows a mixture of reactive and cognitive behaviors [11, 9, 14]. That architecture realizes the Turing Machine concepts proposed by Ferguson [2], and has been inspired by a previous paper by B. Hayes-Roth [4].

2 Motivations

We try to clarify in this section the definition and use of basic notions like the ones of reactive and cognitive agents, or short-term and long-term memory. Some emphasis is put on the notions of world model and incompleteness of information.

Pure Reactive Player. A pure reactive player makes use of the mere direct information coming through its input sensors to react, i.e. to perform actions. There is no need to register any past sensory information, nor to perform any integration within an internally built word model. New incoming information is the only information used, and it is used instantaneously to decide for the next action. A pure reactive player "reacts" at the time of perception and all the

available sensory information is used until new sensory information is available. Such agent therefore possesses a short term memory (ST_memory) that holds information valid only for the time interval between two perceptions.

Full Cognitive Player. A full cognitive player is able to elaborate an internal model of the world, based on the past and present sensory information. This dynamic model in turn may be used to maintain information about moving objects, and to estimate the certainty of this information. This knowledge is used to deliberate and decide about new actions. Such agent therefore possesses a long term memory (LT_memory) that holds information allowing the development of long term goals.

To illustrate these notions, lets take the example of a simple task like the one consisting in following another player (FP). This task will be performed differently, depending if the player is designed as a reactive or a cognitive agent.

For a pure reactive player a pseudo-code for this task can be:

1. Wait for new sensory information (copy it in ST_memory)
2. Look for the X_player position in the ST_memory
3. (a) IF {found} THEN {go into the corresponding direction for one time interval}
   (b) ELSE {move at a new position to look for X_player and go to 1}
4. go to 1

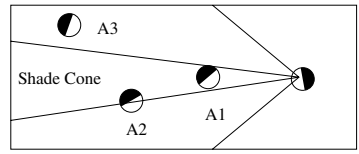
For a full cognitive player the pseudo-code to perform the FP task can be:

1. Wait for new sensory information (copy it in ST_memory)
2. Look for the X_player position in the ST_memory
3. (a) IF {not found} THEN {go to 4}
   (b) ELSE {follow the X_player for one time interval and go to 1}
4. estimate the X_player position (LT_memory)
5. (a) IF {estimation fails} THEN {go to 1}
   (b) ELSE {(go into the computed direction for one time interval and go to 1}

Conceptually speaking, the two behaviors are very similar, since they are made of the same basic actions. However, the way information is processed, and more precisely the coupling between information processing and action is different, since in the first case, deciding for a new action is the only way to react to the absence of information, whilst reasoning (a mental, or epistemic action) is used in the second case to try and recover more information. In fact, the two abilities should be seen as complementary, since the reactive player will behave in a faster way, and be able to react to small instantaneous changes in the environment, whilst the cognitive player may spend most time reasoning without having to think many decisions about action. To face this complementarity, we have developed a hybrid player model Fig.2 by mixing reactive and cognitive

features in the same internal structure. Reactive and cognitive processes run concurrently in this model, under different priorities of execution : a high priority will be given to cognitive processes in case enough time is available, whilst the reactive part will drive the agent otherwise. The ST_memory is rather used by the reactive part of the agent, while the LT_memory is rather used by the cognitive part of the agent. In addition, some restricted exchange of information is allowed to take place between the two processes, when needed by the reactive part of the agent.

The Sensory Information. We have focused our attention on the visual information, because it represents what the player sees. Visual information is provided to the agent as a list of objects describing the relative distance, angle, distance variation, and angle variation, as shown in Fig. 1. To be noticed is the fact that the soccer simulator introduces noise, imprecision and incompleteness, as illustrated in Fig. 1. As shown in this figure, the information about the three "observed" players is transmitted to the "observing" player with different levels of accuracy and completion. For A1, full information about team name, player number and position is given with high precision. For A2 on the contrary, its relative distance and occluded position results in less accurate and complete information : no information about the team number is available, only its position and velocity are transmitted by the server to the "observing" agent. For an even farer player like A3, the only information is position and relative velocity, and its accuracy is not very high.



Visual information:
For A1: ((player team_name uniform_number) distance angle distance_variation anglevariation)
For A2: ((player team_name) distance angle)
For A3: ((player) distance angle)

Fig. 1. Accuracy and completeness of perception

As a consequence, the accuracy and completeness of the information available to a given player depends on its relative position and distance to others. It may happen in fact that it is impossible to guess whether another player belongs to the same team or not, depending on its distance from the observer, or whether he is moving or not. As a consequence, some processing must be included, in order to integrate the client information with the time and to build a consistent model of the world, thus allowing the estimation of lacking information and the solving of possible conflicts.

3 A Hybrid Agent Architecture

The agent architecture that we propose in this section is born from the necessity to integrate two complementary ways to process information in a single player : a cognitive one and a reactive one. A logical view of the proposed agent architecture (LA) is given in Fig.2, in terms of functional units. The functional units are designed as processes running in parallel and communicating by message passing to improve the global performances.

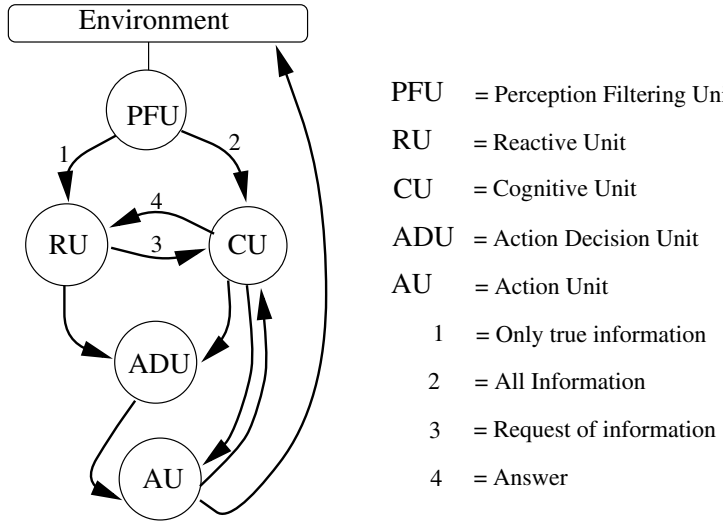


Fig. 2. Player Logic Architecture

Five main functional units may be distinguished in the proposed LA. New information is received at any time interval by the *Perception and Filtering Unit (PFU)*, from the simulator, in order to be processed. The role of the PFU is to split incomplete information from complete information, based on a straightforward analysis of the corresponding objects (see Fig. 1). The sole complete information is then transmitted to the *Reactive Unit (RU)* by the PFU, whilst all information is simultaneously sent to the *Cognitive Unit (CU)*. The RU then starts processing the transmitted information, and if a decision can be made sends a request to the *Action Decision Unit (ADU)*. The CU may concurrently infer a new action and send a request to the ADU. The role of the ADU is then to decide which action to perform from a list of actions, coming from both the RU and the CU, according to some a priori provided criteria. The decision is then transmitted to the *Action Unit (AU)* whose role is to perform the action by sending a command to the simulator. A trace of the performed action is also transmitted to the CU.

3.1 The Reactive Unit and its ST_memory

The reactive unit (see Fig.3), as told in previous sections, is a unit that gives the player the ability to react to new stimuli coming from the environment via the PFU. The RU comprises one *Short Term memory (ST_memory)*, one *Communication Unit (ComU)* and many different *Reactive Processes (RPs)*. The RPs work in parallel and share the common ST_memory. This memory is merely designed as the hard copy of the information coming from the PFU. It may also happen, as already mentioned, that one RP is lacking some information to perform its task. It is then given the possibility to access the LT_memory, which may then become a shared resource for all RPs. Such access is performed via the ComU. All decisions of action coming from the RU are sent to the ADU. The global processing of the RU is a cyclic process of the type "(i) read information (ii) decide action". The LT_memory is designed as a simple blackboard where information is maintained by the PFU, and in read-only access for the RPs. Communication via the ComU should be performed under a specific language (for example the KQML [15] language based on the KIF [8] format, developed at Stanford), but in the present implementation, it is reduced to a direct access to the LT_memory with read-only permission.

3.2 The Cognitive Unit and its LT_memory

The cognitive and reactive units work in parallel. The role of the cognitive unit is to develop more complex behaviors, like planning, or cooperation with others. It has to be noticed however, that the CU functional architecture (see Fig. 4) is more complex than the RUs one, which may increase the time spent in reasoning before reacting to changes in the environment. It may happen therefore, if the environment is changing rapidly, that the CU is not given sufficient time to decide for the next action (the decisions are then taken from the RU) or even that the decisions are inconsistent with respect to the evolution of the environment. Conversely, when the environment is changing less rapidly, no urgent decisions are taken by the RUs, and more time is given to the CU to reason, so that the overall behavior will reveal more elaborated. The proposed approach is therefore consistent with the generic idea of anytime execution, where more accurate processing is performed when more time is available. The interesting feature of this approach also lies in the fact that pure reactive behavior is performed when sufficient information about the world is available, thus allowing a flexible and dynamically determined alternation between more reactive and more cognitive behaviors.

The CU is composed of the following modules: New Information, LT_memory, World Model, Action Model, Cognitive Inference System, and Cognitive Action Decision. The New Information modules entails the information that is perceived by the agent at each time interval. It is the only interface between the external world and the Cognitive Inference System. The role of the World Model is to store knowledge about how the world is built, in terms of rules and constraints. It is composed of two parts: the Dynamic Models describe for all objects how

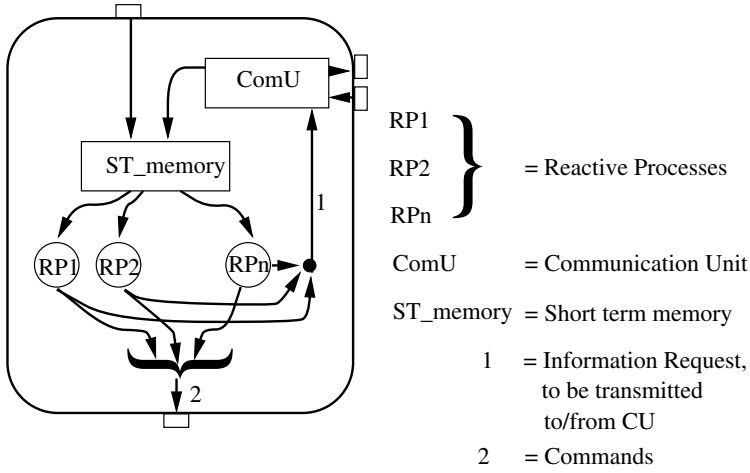


Fig. 3. Reactive Unit (RU) functional architecture

their status may change as a function of their own characteristics, while the Linguistic Models describe how to derive new object descriptions. The Action Model comprises a list of actions and models describing, for any action, its influence on the environment. The role of the Cognitive Inference System is to evaluate and modify the LT_memory at each time interval. The LT_memory is the repository of the knowledge gained by the agent about the world, either by direct external perception, or by internal reasoning. It is the central data base for the cognitive component of the agent. The refreshment cycle of the LT_memory depends on the events that can change its status. These events may be either new information or new actions. When new information is received via the inference system, the LT_memory is updated, based on the current LT_memory and World Model rules and constraints. When new actions are performed by the agent, the LT_memory is also updated, depending on the Action Model and the current LT_memory. The updating process of the LT_memory is a kind of non monotonic reasoning, with some restrictions benefiting from the peculiarities of this application (a priori knowledge of whether some information of the external world is complete or not) to allow a fast processing.

3.3 The Way Information Is Routed

The information is divided by the PFU into two categories: complete and incomplete. An information is said to be complete when each element in the object list is given a description, thus allowing the agent to behave in a perfectly (locally) known environment. A reactive behavior is launched in this case (i.e. the information is transmitted to the RU). An information is said to be incomplete when some elements in the object list are lacking. A cognitive behavior is launched in this case (i.e. the information is transmitted to the CU). In fact, some more rea-

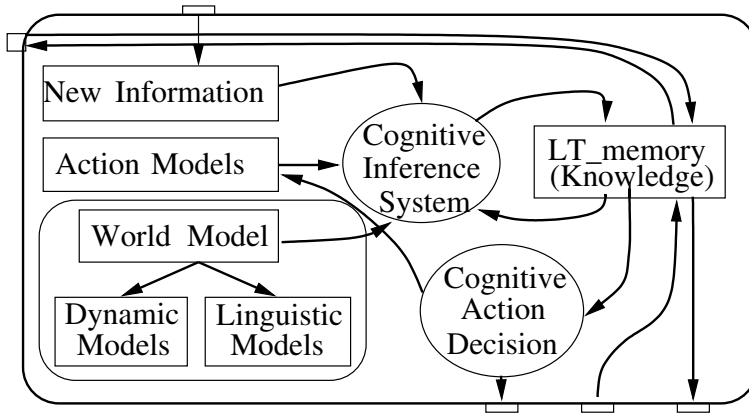


Fig. 4. Cognitive Unit (CU) functional architecture

soning is necessary in this case to ensure that the agent behaves in a consistent way.

Consider for example the two following information messages:

- Information 1: ((player team_1 uniform_10) 10 20 3 5)
- Information 2: ((player) 10 30)

In the first case all information about the player is available whilst much less is known in the second case. More information is needed in this case before deciding for any action. Due to the possibility given to the RU to communicate with the CU to obtain more information, a wide range of situations are in fact considered, ranking from pure reactive situations, where the RU works as a stand alone module, to pure cognitive situations.

3.4 Processing Incomplete Information via the CU

Three different cases have to be distinguished, as regards the completeness of the information received by the CU:

- *Case 1:* Every object is correctly and completely perceived;
- *Case 2:* Some information is lacking, because of the limited perception abilities of the player (distant objects, objects lying out of the agent vision cone);
- *Case 3:* What is perceived is incomplete and imprecise, due to environmental conditions : objects lying within the vision cone may be incompletely seen or even lacking, if occluded for example by another object.

No specific problem is encountered in the first case: the LT_memory can be directly updated with the new incoming information. Some reasoning has to be performed in the second case to retrieve lacking information : past information may be used for this purpose, and provide estimates for lacking objects, based on

the dynamic models available (the new position of the ball for example may be estimated based on past information about its position and velocity). In the third case, incomplete information may be retrieved by means of a matching process under which the effective and estimated information about the world, stored in the LT_memory, is matched against the incomplete descriptions available.

The whole information management process may be summarized as follows:

- *Case 1:* Update the LT_memory based on complete object descriptions;
- *Case 2:* Try and estimate lacking information by simulating the evolution of previously seen objects, if currently out of scope;
- *Case 3:*
  1. Update the LT_memory with the estimated information.
  2. When information is incomplete, try and find matching descriptions in the available object lists, then complete this information;
  3. Update the LT_memory with the completed information.

4 The Application

The proposed agent architecture, its various processing units and modules, have been designed as general purpose components. We have tested the validity and performance of these components by building dedicated agents for the RoboCup competition.

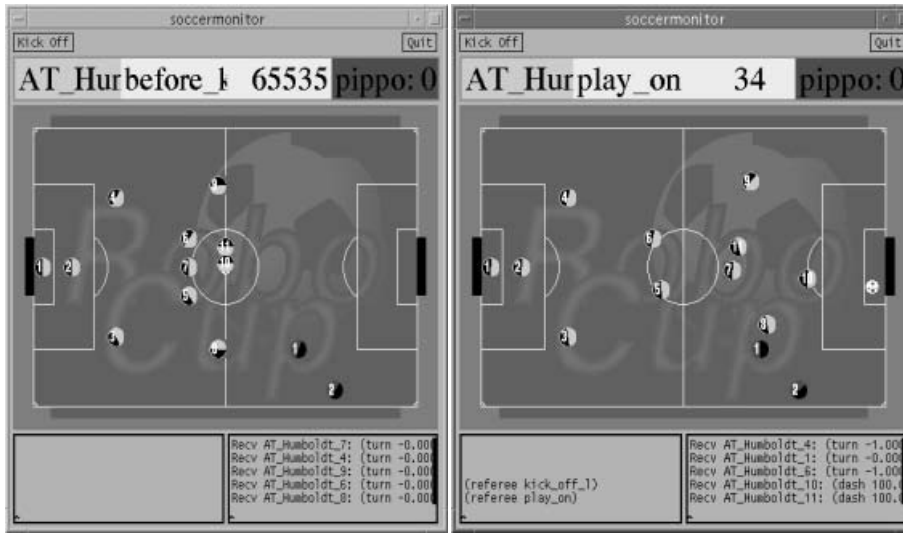
The system is currently under implementation; it is running under UNIX and implemented in C++.

We focalize in what follows on the way reactive and cognitive players differently process information, due to their distinct memory abilities. We propose to analyze the two initial and final situations depicted in Fig.5. The opposing team is figured out in white on this display, it is based on the Humboldt University program team, winner of RoboCup 97. Our team, figured out in black, is comprising only two players, in order that each of them has maximal sensory information available.

Player 1 in our team is designed as a pure reactive player, and denoted as R_Player, whilst player 2 in our team is designed as a hybrid player, and denoted as H_Player. R_Player only uses reactive components and the ST_memory. H_Player, on the contrary, is designed according to the proposed hybrid architecture: it may use reactive as well as cognitive components, and makes use of a LT_memory. As may be observed from Fig.5, these two players are inactive, and do not move from the initial to the final situation, whilst the opponent players and the ball are actively moving : the objective of the following sections is to show how the knowledge and representations of R_Player and H_Player evolve under these conditions.

4.1 Memory Structure

Every sensory information is transmitted in the form of a list of object descriptors starting with *(see x ....)* where x is the referred time for the action of seeing. The



Inital Situation *Final situation*

Fig. 5. An example game situation

objects are represented by their name and their features. As already exemplified, *((player) 73.7 1)* means *(object = player, distance = 73.7, angle = 1)*. A more detailed explanation for the grammar can be found in the soccer server manual [6].

Whatever the kind of memory possessed by the agent, information is stored as a list of object descriptors. For each object a list of features is reported in the corresponding object descriptor array. The meaning of each element is as follows:

obj : object identification number, used to find the object in the object descriptor array;
type : object type identification number, used to represent an object type (for example, $0 \equiv$ ball, $2 \equiv$ opponent player, etc.);
id : internal identification (for internal use);
see : number used to represent the time at which the player has seen the object;
sim : number used to represent the amount of simulations for the dynamic object (used only in the LT_memory);
ux,uy,px,py,vx,vy,ax,ay,dv : array used to represent various geometric features (position, velocity and so on.);
ϑ : flag (for internal use);

4.2 Reactive vs Cognitive Player in the Initial Situation

In this section we analyze how the two players handle the information available in the initial situation shown in Fig.5. The two players see the opponent players,

the ball as well as other information. The corresponding agents are launched for the very initial time at time 0, so they have no memory of the past (if any).

Below is the information available to R_Player at time 0.

R_PLAYER sensory information at time 0. " (see 0 ((goal l) 75.2 15) ((flag c b) 24.5 -34) ((flag l t) 90 36) ((flag l b) 73.7 -10) ((flag p l t) 68.7 35) ((flag p l c) 59.7 19) ((flag p l b) 56.3 0) ((player) 73.7 15) ((player) 66.7 17) ((player) 49.4 3) ((player) 60.3 36) ((player AT_Humboldt 5) 33.1 23 -0 0) ((player AT_Humboldt) 40.4 41) ((player AT_Humboldt 7) 36.6 33 -0 0) ((player AT_Humboldt 8) 22.2 0 0 -0) ((line l) 72.2 -90)) ".

This sensory information is processed by the R_player reactive units. According to the current design, only information about the players and the ball is considered. From this set of information, only complete descriptions are used, like : *(player AT_Humboldt 5) 33.1 23 -0 0)*, where the team name *(AT_Humboldt)*, the player number *(5)*, the distance, the angle, the distance variation and the angle variation *(33.1 23 -0)* are known. Information like *((player) 60.3 36)* is discarded, because incomplete.

The processing of this information by the R_player produces a new representation in ST_memory reported in Tab.1.

Table 1. State of the R_player ST_memory at time 0

Object	obje	type	id	see	sim	ux	uy	px	py	vx	vy	ax	ay	dv	ϑ
Humboldt 5	16	2	0	0	0	0	0	10.7	−6.5	0	0	0	0	0	1
Humboldt 7	18	2	0	0	0	0	0	10.9	0.4	0	0	0	0	0	1
Humboldt 8	19	2	0	0	0	0	0	2.5	−19.4	0	0	0	0	0	1

The information reported in Tab.1 is the result of a rough filtering and processing of the sensory information transmitted by the server. Only three objects are correctly seen : the opponent player 5,7 and 8, respectively represented under identifiers 16, 18 and 19 in the ST_memory. Information about eight players in total has in fact been transmitted, the other players being out of viewing (like *opponent 11* for example). Among these eight players, five of them are too far away or partially occluded (like players 6,4 and 2), which results in the transmission of partial information which is not considered by the R_player.

Below is the information available to H_Player at time 0.

H_PLAYER sensory information at time 0. "(see 0 ((goal l) 87.4 -25) ((flag c t) 70.8 19) ((flag l t) 104.6 -7) ((flag p l t) 83.1 -8) ((flag p l c) 72.2 -20) ((flag p l b) 66.7 -36) ((ball) 40.4 0) ((player) 81.5 -24) ((player) 81.5 -22) ((player) 60.3 -33) ((player) 73.7 -7) ((player AT_Humboldt) 44.7 -15) ((player AT_Humboldt) 54.6 -2) ((player) 49.4 -8) ((player AT_Humboldt 8) 33.1 -27 -0 0) ((player)

60.3 12) ((player) 44.7 0) ((player AT_Humboldt) 44.7 4) ((player pippo 1) 13.5 0 -0 0) ((line t) 90 -45))"

The processing of this information by the H_player produces a new representation in LT_memory reported in Tab.4.

Table 2. State of the H_player LT_memory at time 0

Object	obj	type	id	see	sim	ux	uy	px	py	vx	vy	ax	ay	dv	ϑ
Ball	0	4	0	0	0	0	0	−1.4	−1.0	0	0	0	0	0	1
Pippo 1	1	3	0	0	0	0	0	−20.4	−20.0	0	0	0	0	0	1
Humboldt 8	19	2	0	0	0	0	0	1.4	−19.4	0	0	0	0	0	1

As may be seen from Tab.4, only three objects are correctly seen : the opponent player 8, the ball and player 1.

To be noticed is the fact that the information available to the R_Player and the H_Player is very similar, when initially launched. At this very initial time in fact, there is no way for the cognitive player to complete the available information by himself, since there is no past information available.

4.3 Reactive vs Cognitive Players in the Final Situation

The two players are now in the final situation depicted in Fig.5. They have the possibility, if given this capacity, to handle past as well as present information. Below is the information available respectively to R_Player and H_Player, together with the state of their respective memories. Note that the information has been gained at different times by the two players, due to the effective independence between the two players, which run concurrently.

R_PLAYER sensory information at time 34 "(see 34 ((goal l) 75.2 15) ((flag c b) 24.5 -34 0 0) ((flag l t) 90 36) ((flag l b) 73.7 -10) ((flag p l t) 68.7 35) ((flag p l c) 59.7 19) ((flag p l b) 56.3 0) ((player) 73.7 15) ((player) 66.7 17) ((player AT_Humboldt) 49.4 3) ((player) 60.3 36) ((player AT_Humboldt 5) 30 29 -0 0) ((player AT_Humboldt) 40.4 44) ((line l) 72.2 -90))"

Table 3. State of the R_player ST_memory at time 34

Object	obj	type	id	see	sim	ux	uy	px	py	vx	vy	ax	ay	dv	ϑ
Humboldt 5	16	2	0	34	34	0	0	6.5	−4.8	0	0	0	0	0	1

H_PLAYER sensory information at time 36 "Buff:(see 36 ((goal l) 87.4 -25) ((flag c t) 70.8 19) ((flag l t) 104.6 -7) ((flag p l t) 83.1 -8) ((flag p l c) 72.2 -20) ((flag p l b) 66.7 -36) ((player) 81.5 -24) ((player) 81.5 -22) ((player) 60.3 -33) ((player) 73.7 -7) ((player AT_Humboldt) 44.7 -10) ((player AT_Humboldt) 54.6 0) ((player AT_Humboldt) 33.1 14) ((player AT_Humboldt 8) 18.2 17 0.364 -0.1) ((player) 54.6 31) ((player AT_Humboldt 11) 36.6 24 -0 1) ((player pippo 1) 13.5 0 -0 0) ((line t) 90 -45))

Table 4. State of the H_player LT_memory at time 36

Object	obj	type	id	see	sim	ux	uy	px	py	vx	vy	ax	ay	dv	β
Ball	0	4	0	17	41	0	0	−46.0	−2.6	−1.3	−0.01	0	0	0	1
Pippo 1	1	3	0	36	36	0	0	−20.4	−20.0	0.0	0.0	0	0	0	1
Humboldt 7	18	2	0	33	37	0	0	−12.7	−1.3	−0.1	0.01	0	0	0	1
Humboldt 8	19	2	0	36	36	0	0	−21.4	−13.5	0.1	0.3	0	0	0	1
Humboldt 10	21	2	0	23	39	0	0	−20.1	−3.4	−0.7	−0.3	0	0	0	1
Humboldt 11	22	2	0	36	36	0	0	−16.9	4.5	−0.5	0.2	0	0	0	1

We have seen that in the initial situation (time 0), the memorizing behaviors remain substantially similar. In the final situation on the contrary, the dynamism of the environment result into more substantial changes. For the R_player in fact, the environment is formed by only one object (object 16). This player has no possibility to reason about changes in the environment, and only possesses an instantaneous representation of the world state. The H_player on the contrary takes advantage of a more complete representation of the environment state and history. To be noticed is the fact that only objects seen at time 36 are directly perceived. Information about other objects (e.g. previously seen objects 0, 18 and 21) has been estimated in the course of a simulation process. The matching process is not implemented yet and therefore not available to the H_player.

Whereas R_player is only given the possibility to react to instantaneous information about the world, H_player is given the possibility to exploit a more complete range of past and present information, thus enlarging his a priori restricted vision abilities by reasoning about non directly perceived objects.

5 Team Description

Our approach to the task of Soccer Competition have been inspired by real soccer competition.[7] In the real soccer the field is divided in three longitudinal zones (A,B,C) and three vertical zones (1,2,3) (See Fig.6). The type of game changes in function of the zone in which the ball and the players are situated. Zones A and C (lateral corridors) are extern and favorable to fast game. The central zone, (zone B), is a zone full of players and very busy. Any player in this zone

has the objective of passing the ball to the external zone to support game in zone C and A. Conversely at the zones 1,2,3 are associated different game spirit. Zone 1 is the defensive zone. Players situated in this zone have the intention to avoid goal and to capture the ball. Zone 2 is an intermediate zone. In this zone players prepare and control the game. The zone 3 is an offensive zone, where all must be done to realize the goal. Player in this zone must be very reactive and opportunist.

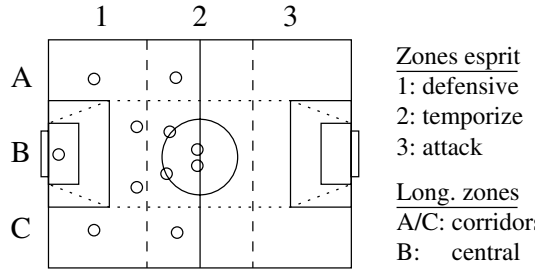


Fig. 6. The field partition, and the team distribution

A formation consists of eleven players, each with a role. The following are distinguish in: one Goalie, 2 Defenders, 4 Midfielders, 2 Wing-Forwards, 2 Attachers. The behavior and the position in the field is determined by the player's role. Players have an adaptation to the field position and changes the spirit of game from attack to defensive following the zone spirit. The role is characterized by some special behaviors. For example Goalie has the behavior of catch the ball. So, our team structure has the follows features: (i) dynamic behavior specialization, (ii) flexible positioning and game orientation, (ii) flexible and dynamic game spirit.

6 Conclusion

We have proposed a hybrid agent architecture as being built from a combination of reactive and cognitive behaviors running in a concurrent way. The core difference between these two kinds of behaviors lies in the associated memory abilities : only short-term memory abilities are provided to the reactive behaviors, whilst full long-term memorizing capacities are provided to the cognitive one. The short-term memory is the mere replication of the initial sensory information that is made available to the agent by the simulator. To be noticed is the fact that parts of this information may furthermore be discarded, if incomplete. The long-term memory on the contrary holds parts of the initial sensory information, completed with information estimated in an autonomous way, based on the reasoning abilities provided to the cognitive behavior. A varied range of decisions may therefore be taken by the ADU, from pure reactive ones in case

sufficient information is available to full cognitive ones when sufficient time is available. Specific to our approach indeed is the fact that reactive and cognitive processes are considered as separate paths, each processing information in an autonomous way, rather than designed in a hierarchical way, as in traditional approaches, where the cognitive processes are meant to control the activation of reactive processes.

References

1. R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.
2. Innes A. Ferguson. Toward an architecture for adaptative rational mobile agents. In *Decentralized A.I. 3*, 1992.
3. S. Gaglio, A. Chella, and M. Frixione. A cognitive architecture for artificial vision. *Artificial Intelligence*, 89:73–112, 1997.
4. B. Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72:329–365, 1995.
5. Housheng Hu and Michael Brady. Dynamic global path planning with uncertainty for mobile robots in manufacturing. *IEEE Transaction on robotics and automation*, 13(5), 1997.
6. N. Itsuki. Soccer server: a simulator of robocup. In *Proceedings of AI symposium '95*, pages 29–34. Japanese Society for Artificial Intelligence, December 1995.
7. A. Laurier. *Football. Culture Tactique et Principes de Jeu.* Chiron Sport, 1985.
8. R. Fikes M. Genesereth and alt. Knowledge interchange format, version 3.0 reference manual. Technical report, Computer Since Department, Stanford University, 1992.
9. N. R. Jennings M. Wooldridge. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10:2:115–152, 1995.
10. Itsuki Noda and al. The robocup synthetic agent challenge 97. Available as:http://www.cs.cmu.edu/afs/cs/usr/pstone/public/papers/97syntetic-challenge/synthetic-challenge.html, 1997.
11. M.J.Wooldridge N.R.Jennings. Agent theories, architectures, and languages: A survey. In Springer-Verlag, editor, *Lecture Note in Artificial Intelligence*, volume 890. 1995.
12. M. Veloso P. Stone. Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer System*, 1997.
13. Manuela Veloso Peter Stone, Mike Bowling. Predictive memory for an inaccessible environment. In *International Conference on Intelligent Robots and Systems (IROS'96)*, 1996.
14. Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.
15. Y. Labrou T. Finin. A semantic approach for kqml, a general purpose communication language for software agents. In *Third International Conference on Information and Knowledge Management (CIKM'94)*, November 1994.