Simulation League, Team *PaSo-Team99*, pages 98–102
http://www.ep.liu.se/ea/cis/1999/007/22/

Balancing Reasoning and Reaction in a Multi-Agent Architecture for RoboCup Competition

PaSo-Team99

Carlo Ferrari, Francesco Garelli, Stefano Griggio, Enrico Pagello

CF, FG, SG, EP: University of Padua (Italy)

1 Introduction

In the past years, the RoboCup competition has been demonstrated to be a very good research platform for studies on multi-agent systems. The availability of a Soccer Server let the people to test their systems acting in a dynamically changing environment, with real-time constraints under limited sensorial perceptions. Proper team-playing strategies can be designed and loaded onto each team-player and experiments can be devised for comparing different models for representing these high-level strategies. The University of Padua has been active in RoboCup since two years ago, developing the PaSo-Team (The University of PAdua Simulated Robot SOccer Team), a Multi-Agent System able to play soccer game for participating to the Simulator League of the RoboCup competition [PagelloÊ1997], [Pagello1998]. One of the big scientific challenge of this project, is to resolve a dilemma between two basic paradigms that marked the past and current history of Artificial Intelligence and Robotics, i.e. planning vs. reaction. In a soccer game the reactive aspects look to be predominant respect with the reasoning and planning aspects of the game. In fact, because each single player shares the global goal of the system (score a goal), he can immediately verify whether he can reach the goal (score the goal), by using the Soccerserver information. That means that at some extent, the decisions of each player do not need any further reasoning involving his team-mates. The experience done at previous competitions seems to confirm the better quality of reactive-based systems. But it also demonstrated that a pure reactive schema suffers some major problems. In fact, it must be considered that the player is not alone in the environment. Thus, the agent cannot be recommended to execute only pure lowest-level reactive actions, due to the presence of obstacles such as other players, (team-mates or opponents) that surely interfere with the development of its strategy to score the goal. Then, obstacle avoidance routines against the opponents agents, and cooperative routines to get collaboration with the team-mates, are necessary. The necessity to overcome the major limitation of a pure reactive system is discussed in literature and some solutions have been proposed. In [Stone

1998], a multi-agent reinforcement learning is introduced in order to partition the task to be performed among all the agents, allowing them to simultaneously learn collaborative policies by observing the long-term effects of their actions. In [Burkhard 1998] the agents of the system maintain a model of their world (belief) and the way from belief to action is guided by the desires of the agent, that can be better expressed by commitment to intentions. In [Pagello1999], the emphasis is about the way to obtain an appropriate global team behaviour, without via implicit communication. In the current version of PaSo-Team (PaSo-Team'99), we inherited from the past versions, the model for the low-level skills, commonly used by each player. Moreover we confirmed the idea of not realizing the coordination via explicit communication. In the current project the emphasis is about finding a good balance between reaction and reasoning in the design of a player. In the next paragraph we will show how it is possible to use sensed data after they are filtered through a proper mechanism (the Synthetic Visual Maps), that builds a world model able to support fast and efficient procedure for motion planning. In the third paragraph we will present the main structure of the PaSo-Team'99 client, with major emphasis on how the obstacle avoidance and motion reasoning can be performed at the behaviour level, while decision and learning are performed through the use of an utility function.

2 Reasoning About Visual Information

The reactive approach uses the information coming from sensors to immediately activate some action properly designed to deal with the sensed data. Complex data fusion procedures do not help reaction because their major role is to build a (maybe local) world model that represent a situation in a more compact form and with a focus on some aspects that are dependent from the actual task. Moreover, the continuos updating of the world model can be better performed if the validity of the acquired information can be maintained along more than one sensing cycle, trying to compute the evolution over time of the model itself. In simulated soccer game, each player is given with different information about its team-mate and opponents, as well as about ball dynamics and about the game field. Because of the nature of the problem (a team game), each single player cannot move simply reacting from raw positions and velocities information of the opponent players (like in a cat & mouse play), both because the presence of its team-mates can produce a variety of physical arrangements that it is real hard to classify, and also because its global attitude can switch very quickly from being a defender or an attacker. Moreover its focus of attention can change as well. In fact a player that has the ball maybe want to defend it against one ore more opponent or it want to drop it in the adversary net. Run-time motion planning and reasoning is the main intelligent activity at an intermediate level, for a single player. This activity is greatly based on obstacle avoidance that becomes a primary tasks in the most game situations. As obstacle avoidance is a primary task, the player world model must easily support a fast kind of obstacle avoidance reasoning. Moreover we want a model that can be produced and updated in a standard way, and that it is applicable in all the game situations. With this major requirements in mind, we developed a model that we call the SVM model, where SVM stands for Synthetic Visual Maps. SVM are a concise representation of the free space around the player. The SVM maps each movement direction of the player with a boolean value that says whether that direction is free or prohibited. Only the nearest elements are considered while building the SVM. In fact the influence on a SVM due to a game element (team-mate, opponent or field element) becomes as less important as its distance of the element itself is increasing. The SVM can be seen as a polar representation of the free space in a proper disk centered in the player. This representation can be easily updated at each sensing cycle, in order to consider new game elements that become important, either because they are moving towards the player or because the player itself is moving towards them. The SVM are computed from the sensed position of those game elements, that are directly reachable from the current player position. These elements form the in-focus set. The SVM for a single player is obtained as a composition of the SVM due to each element in the in-focus set. While building a SVM it is possible to explicitly take into account some correction factors that can enlarge the actual dimensions of an opponent player, in order to be able to make a choice about the movement direction that shows to be robust in spite of all the possible movement of the opponent player itself.

3 The PaSo-Team (The University Of Padua Soccer Team)

As each team is formed by 11 players, we need 11 client programs to compete for a match in RoboCup simulator. In our implementation we have developed in C++ language a single program that contains the features of all kind of players. This is because the simulation context is different from real soccer, where the human players have different physical abilities. In our team all the players are similar to clones of an individual player with certain characteristics. The only thing that distinguishes one player from another is the role, that slightly makes changes in the way of reasoning of the players. The technical structure of each player is based on encapsulating the interface with the Soccerserver in order to obtain a more robust code with respect to changes in the Soccerserver itself, and to represent the different player capabilities at different levels of abstraction. At the lowest level the PaSo-Team'99 clients act via B-Actions (Basic-Action), that code the information to be sent to the Soccerserver. A proper "talker" module is devoted to explicitly send information to the server, using the information prepared by the B-Actions. B-Actions form a catalogue of primitives that can be used to model a kind of more complex action: the Skills. Skills are used to model atomic actions, like kicking the ball or looking around. Using Skill and B- Action let possible to separate the logical capabilities of a player from its particular implementation due to a specific server. Information from the server are handled by the "listener" module. The listener module solves the synchronization problem with the server and it receives the data items related to the game development. The data interpretation activity is devoted to a "parser" module, that can update the player local status, that is, the player "memory". The memory module is used as a data container able to answer to every request from the higher reasoning level modules. Among these modules there are the behaviour modules (or simply Behaviours). They realize the intelligent manoeuvring of a player, according to the different game situations. Behaviours are coded using Skills, and they also compute the SVM for motion reasoning. Some typical behaviours are devoted to control the dribbling, or to stop an opponent player, or to kick the ball away. While playing, every player can activate a particular behaviour (its running behaviour) at a time, and, usually, all the player of a team do not have the same running behaviour. In this way it is possible to have a global team emergent behaviour, that it is due to the compound effects of all the players behaviours. In the current approach, the PaSo-Team player do not code explicitly the coordination procedures for obtaining complementary actions, that arise, instead, because of a proper interpretation of the perceptions sent by the soccerserver. The high level reasoning is devoted to select the proper behaviour, that becomes the running behaviour. This kind of reasoning requires to solve a two-fold problem: the choice & evaluate problem. At each game cycle each player must have a running behaviour, that is, each player must either confirm its running behaviour or decide to change it with another one that is more appropriate to the game development. Moreover, it should maintain and update the evaluation of its past behaviour choices, with respect to the different game situations. In fact, the success of the action of a player depends also on the actions of the opponent players, and it is necessary to deal with the failure of the running behaviour goal. Recognizing failures, and learning from them is a key point for the high level reasoning. So far, the choice & evaluate problem is solved first looking for the most promising behaviour, that is, the behaviour that maximize an utility function over a proper coding of the game situations, and then updating this utility function measuring how good the behaviour has been in reaching its goal.

4 Conclusions

In the PaSo-Team'99 project we are experimentally investigating how much the reaction schema for intelligent agents team must be integrated with some kind of high level reasoning. In PaSo-Team'99, sensed data are used in the decision process after they are filtered through a proper mechanism (the Synthetic Visual Maps), that builds a world model to support fast and efficient procedure for motion planning. Obstacle avoidance and motion reasoning are encapsulated at the behaviour level, while higher levels deal with decision and learning. The choice & evaluate problem is performed separately by each player, using an utility function over a proper coding of the game situations

5 Acknowledgements

This research work could not be done without the enthusiastic participation of the students of Electronics and Computer Engineering Undergraduate Division of Padua University Engineering School. Financial support has been provided by both CNR, under the Special Research Project on "Real-Time Computing for Real-World" and Murst, under the 60

6 References

[Burkhard 1998] H-D. Burkhard, M. Hannebauer, J.Wendler, "Belief-Desire-Intention Deliberation in Artificial Soccer", AI Magazine, Fall 1998.

[Pagello 1997] E. Pagello, F. Montesello, A. D'Angelo, C. Ferrari, "A Reactive Architecture for RoboCup Competition", Proc. First Int. Workshop on RoboCup, H.Kitano ed., Lecture Notes in Artificial Intelligence, Springer -Verlag, 1997.

[Pagello 1998] E. Pagello, F. Montesello, F.Candon, P.Chioetto, F.Garelli, S.Griggio, A.F. Grisotto, "Learning the when in RoboCup Competition", Proc. Second Int. Workshop on RoboCup, Lecture Notes in Artificial Intelligence, Springer -Verlag, 1998.

[Pagello 1999] E. Pagello, A. D'Angelo, F. Montesello, F. Garelli, C. Ferrari, "Cooperative behaviors in multi-robot systems through implicit communication". Robotics and Autonomous Systems, to appear, Spring 1999.

[Stone] P.Stone, "Layered Learning in Multi-Agent Systems", Ph.D. Thesis, School of