PaSo-Team'99

Carlo Ferrari, Francesco Garelli, Enrico Pagello

Department of Electronics and Informatics, The University of Padua, Italy
Via Gradenigo 6a. I-35131 Padova, Italy
{carlo,pascal,epv}@dei.unipd.it

Abstract. PaSo-Team is a Multi-Agent system for playing soccer game in the Simulation League of the RoboCup competition. This paper describes the ideas and the technical structure of PaSo-Team'99, that played at RoboCup-99, in Stockholm during IJCAI'99. The main goal of the 1999 project was about the integration of a reactive model with some kind of high-level reasoning. Obstacle avoidance and motion reasoning are encapsulated at the behavior level. They use a proper world model (built from the sensed data), that focus on the relevant objects. The choice&evaluate problem is performed through an utility function over a proper coding of the prototypical game arrangements.

1. Introduction

Following the experiences done in 1997, and 1998 competitions, [2], [3], it has been developed the '99 release of *PaSo-Team* (The University of PAdua Simulated Robot SOccer Team), namely PaSo-Team'99. The PaSo-Team project has been conceived as a Multi-Agent System that must be able to play soccer game. While maintaining its historical origins of being a multi-agent reactive software architecture based on Brooks' Subsumption Architecture, PaSo-Team'99 tries to overcome the major limitations of pure reactive systems, introducing a more abstract level of learning and reasoning used to proper differentiate the current behaviour of the various component agents of the system, according to the actual phase of the game.

One of the big scientific challenge of this project, is to resolve a dilemma between two basic paradigms that mark the past and current history of Artificial Intelligence and Robotics, i.e. planning vs. reaction. The experience done at previous competitions seems to confirm the better quality of reactive-based systems, for soccer games. 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. In the current project the emphasis is about finding a good balance between reaction and reasoning in the design of a player.

M. Veloso, E. Pagello, and H. Kitano (Eds.): RoboCup-99, LNAI 1856, pp. 618–622, 2000.

2. Team Development

Team Leader: Enrico Pagello (Associate Professor of Computer Science)
epv@dei.unipd.it
Dept. of Electronics and Informatics, The University of Padua,
Via Gradenigo 6a, 35131 Padua, Italy
and
Institute LADSEB of CNR, C.so Stati Uniti 4, 35100 Padua, Italy,
Team Member: Carlo Ferrari (Assistant Professor of Computer Science)
Francesco Garelli (Undergraduate Student)
Stefano Griggio (Undergraduate Student)
Andrea Sivieri (Undergraduate Student)

Dept. of Electronics and Informatics, The University of Padua,
Via Gradenigo 6a, 35131 Padua, Italy and
{carlo, pascal, btenia, tigre}@dei.unipd.it

WEB page: http://www.dei.unipd.it/~robocup

3. The World Model

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. Figure a,b and c show how to use the SVM for a dribbling action.

4. Communication

In the current PaSo-Team project we confirmed the idea of not realizing the coordination via explicit communication. In [4], it has been shown how it is possible to obtain an appropriate global team behaviour, via implicit communication., for a generic multirobot system. With implicit communication the agents can communicate looking at the current status of the environment. The interpretation procedure has been tuned to measure how the team is carrying its global task.

5. Skills

In PaSo-Team'99 we 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 clones of a prototype player. The only thing that distinguishes one player from another is the role, that slightly makes changes in its reasoning strategy. The technical structure of each player is based on encapsulating the interface with the Soccerserver, 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.

6. Strategy

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 either confirm its running behaviour or decide to change it with another one that looks 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. Recognizing failures, and learning from them is a key point for 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.

7. Conclusions

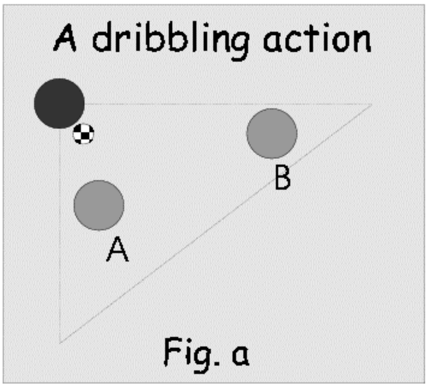
In the PaSo-Team'99 project we experimentally investigate how much the reaction schema for intelligent agents team must be integrated with some kind of high level reasoning. In PaSo-Team'99, sensed data are used in the decision process after they are filtered through a proper mechanism (the Synthetic 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

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% and 40% Grants.

References

1. Burkhard, H.D, Hannebauer, M., Wendler, J.: Belief-Desire-Intention Deliberation in Artificial Soccer. AI Magazine, Fall 1998.
2. Pagello, E., Montesello, F., D'Angelo, A., Ferrari, C.: A Reactive Architecture for RoboCup Competition. In Kitano H. (ed.): Proc. First Int. Workshop on RoboCup, Lecture Notes in Artificial Intelligence, Springer -Verlag, 1997.
3. Pagello, E., Montesello, F., Candon, F., Chioetto, P., Garelli, F., Griggio, S., Grisotto, A.F.,: Learning the when in RoboCup Competition. Proc. Second Int. Workshop on RoboCup, Lecture Notes in Artificial Intelligence, Springer -Verlag, 1998.
4. Pagello, E., D'Angelo, A., Montesello, F., Garelli, F., Ferrari, C.: Cooperative behaviors in multi-robot systems through implicit communication. Robotics and Autonomous Systems, 29 (1999), 65-77
5. Stone, P.: Layered Learning in Multi-Agent Systems". Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, December 1998.



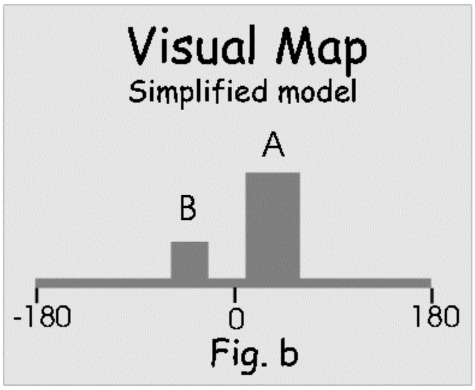


Fig. a. A typical action of dribbling: our player has the ball, A and B are opponents

Fig. b. A simplified map for situation in fig. a without filters and complex shape maps

Fig. c. The **real** map created by our client; note the effects of filters in promoting the forward direction.