VitoriaRC Team Description

Pedro Manuel Pinto Ribeiro

Computer Science Department / Faculty of Science / University of Porto
pribeiro@dcc.fc.up.pt

Abstract. This paper presents an overview of VitoriaRC 2D simulation league team. Its main research focus is on the use of Inductive Logic Programming in such a complex domain. It presents a description of the team multi-threaded architecture and of the high-level reasoning used. It also gives motivation for the use of Inductive Logic Programming, as well as future research directions for the team.

1 Introduction

Robocup is an international research and education initiative that attempts to promote and develop artificial intelligence and robotics. Its main contribution is to provide a standard problem were a wide number of technologies can be integrated and examined [1]. For that, it focuses on soccer domain, and it promotes competitions and scientific conferences. It has served to aggregate efforts of hundreds of researchers.

VitoriaRC was born precisely with the motivation of contributing to such a great initiative. The author always enjoyed soccer and Artificial Intelligence, and being able to merge these two felt almost like being in paradise. The base of the team was made in 2002, in the final project of an undergraduate course. All the code was made completely from zero, in order to better study and understand the functioning of the simulation, and the result was a fully functional team. The development was then discontinued for some time, but is now being continued. Never leaving the subject, the author is now doing a Master's Thesis in Artificial Intelligence and Computation, and VitoriaRC will be used as a platform to study the use of Inductive Logic Programming in Multi-Agent Systems, since this area still provides a strong personal motivation.

This paper is organized as follows. Section 2 presents a brief overview of the team architecture, describing the synchronization methods and the flow of data. Section 3 gives a view on the current high-level functioning of the team, in order to understand its global behavior. Section 4 presents motivation for the use of Inductive Logic Programming in Robocup, giving some research directions. Section 5 ends the paper with concluding remarks.

2 Team Low-Level Design

Although being made completely from zero, the team also "drinks" some inspiration from existent teams. In particular, UvaTrilearn [2] proved to be very good example of how a team could be made. Like that team, VitoriaRC is made in C++, with a lot of concern for software engineering aspects, being extensively documented, in order to facilitate its development by several persons at the same time.

To tackle with the asynchronous nature of the simulation, a multi-threaded approach was taken, as detailed in figure 1.

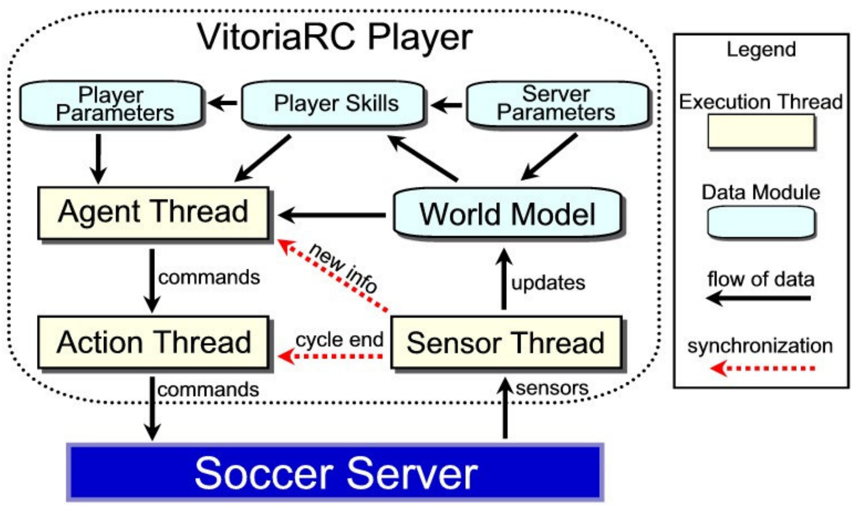


Fig. 1. VitoriaRC Agent Architecture

Each player has 3 execution threads. The sensor thread is responsible for receiving all sensorial messages from the server, no matter at what time they arrive. Whenever it receives new info, it updates the World Model accordingly. The Action Thread is responsible for all the communication from the team to the server, this is, gives all basic commands like dash and kick to the server. In order to know when it is the best time to send the messages, it uses signals from the Sensor Thread, based on the messages arrival time. Agent Thread represents the player's brain, and it is activated each time new sensorial info is received, trying to know what the player should do based on its surroundings. It is important to say that the high-level decisions of the agent thread have persistency, this is, low-level commands are calculated for more than one cycle, and are all stacked up in a queue in Action Thread. In this manner, even when the Action Thread takes more than one cycle to decide, the agent can continue to behave rationally. This approach effectively makes the team adapt to a continuous simulation, if that would be the path to follow.

In what refers to data modules, the World Model has information on all game info, including the ball, players, flags, result, game mode and time. Server Parameters contains detailed information on all server-side parameters that were communicated on the connection to the server. Player Parameters stores all parameters that affect the team behavior (and can be changed by a human). Player Skills provides the Action Thread with a more sophisticated set of commands, being responsible to translate them in primitive commands understood by the soccer server. This set includes simple things like moving to a position or aligning the neck with the body, but also contains more complicated things like passing the ball to a teammate, shooting to the goal, intercepting the ball or dribbling.

From the beginning it also seemed that effective debugging could be crucial on a domain like this. Therefore, a logging scheme was adopted, and a visual debugger was created, trying to visually perceive if the team functioned as expected, understanding why it behaved in a way instead of another. Figure 2 shows a screenshot of the debugger, showing the range of the visual sensor and the perceived players. The logs, in conjunction with the debugger, were used to manually tune the player skills.

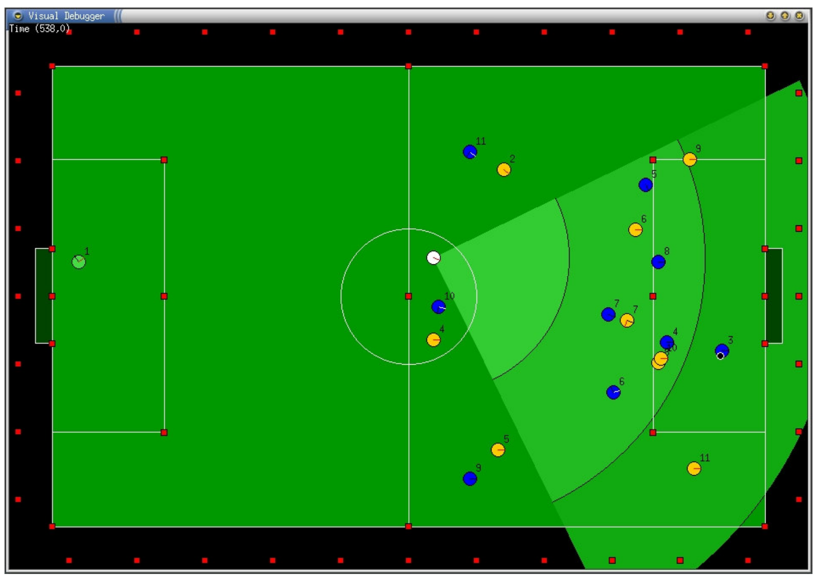
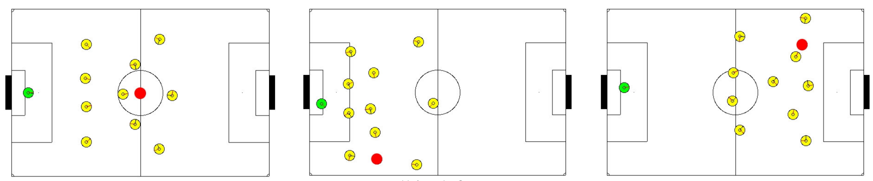


Fig. 2. VitoriaRC Visual Debugger

3 High-Level Functioning

So far, only the individual behavior of the players was described. This section shows how the team is organized is order to function as a whole. Any real soccer team uses tactics and formations to decide the positioning of the players in the field. In that way, vital zones of the field can be occupied and when the ball is recovered new pass lines can be created. Having as basis the SBPS [3], a formation scheme was created, and several typical formations were created. Having in account the ball and opponents position, the formation scheme provides an estimate of where each player should be. Figure 3 shows how an example 4-3-3 formation and the respective positioning of the players, having in account the ball position.

The use of formation already provides some coordination in the team, but it needs to know the ball position, probably the most important info in the game. Having that in account, when a player knows with some confidence the ball position and velocity, it communicates that to the rest of the team. The information is encoded in 6 bytes, leaving the other 4 for coordination aspects. In the current implementation, the other 4 bytes are not used, but plans are already made in how to use them.



Ball in the middle Ball in defense Ball in attack

Fig. 3. VitoriaRC example 4-3-3 formation (the ball is the red circle)

In order for the team to reach its objectives, each agent must select at each time what action to make. This is not an easy task and of course is one of the most important research issues in the simulation league. However, this is currently the Aquiles Heel of VitoriaRC. The team mainly uses decision trees, having a different tree for the goalie and the rest of the players. Then, a different ramification of the tree exists for each of the following four main situations: the player has the ball, the player is the team closest player to the ball, the team is attacking (has possession of the ball) or the team is defending (the opponents have the ball). Using the Player Skills and Parameters, each player attributes points to possible actions and chooses the best one to execute. For example, priority for the player with the ball is given to a goal chance. If that is not possible the player can pass to a teammate, dribble with the ball, just shoot the ball to front or simply wait for the team to position better in the field. All this process has been manually tuned, but one must admit that a lot of work is already waiting to be done.

4 Inductive Logic Programming

Inductive Logic Programming (ILP) is a branch of machine learning built on the junction of two "roads": logic programming, which provides the formalism for the training examples and theories learned, and induction, which gives us the learning method. The main research focus of VitoriaRC is precisely how to integrate ILP in the team, using it to improve the team. One may ask why should ILP be used, when several other learning techniques were already used with different levels of success (ex: reinforcement learning, neural networks or genetic programming). This is a good question and this section tries to give the necessary motivation.

One of the main advantages of ILP is the fact that it is a declarative learning method, which surely gives a more human understandable representation of the knowledge acquired. And in some domains, the "why" of the solution can be as important as the solution itself. Other important advantage is the ability to incorporate background knowledge, which can be crucial in complex and very dynamic environments, giving a higher degree of quality to the learning and decision process. Another notable aspect is the possible recycling of learned knowledge, when other machine learning methods require a complete new retraining phase, even when a slight change on the goal is made.

If we cross these characteristics with learning on multi-agent systems, we see lots of potential. However, multi-agent inductive logic learning is still a relatively young research domain. Some attempts were already made to summarize potential gains and downfalls, such as [4], were issues like learning pure logic programs vs decisions lists, eager ILP vs analogical prediction or single-predicate vs multi-predicate learning were discussed. But a lot of work is yet to be made, and this is still an open research direction. It should also be noticed that the ILP community seems eager to demonstrate that ILP can be used on a much broader sense than the typical classification and description tasks, trying to embrace artificial intelligence in all its glory. In [5], this is suggested as a research direction to be taken and Robocup, with all its public exposure, seems the perfect domain for that to happen.

ILP was already used in Robocup domain. Leuven applied it to verification and validation of their team [6]. Other was more focused on the soccer playing activity and decision making. A framework for inductive learning soccer agents was proposed, in order to give them the possibility to acquire knowledge based on past behavior [7]. This research focused on action checking and has made some progress but not enough to use it on a real simulation team. To my best knowledge, these are the only two applications of ILP in the simulation league.

We can see that there is a lot to research in this area. Currently, we are researching the kind of decision framework that could incorporate ILP. We are researching mainly off-line learning, but some on-line learning ILP concepts are being considers. Some experiences are being made with simplified situations (ex: goalie defending ball, player trying to score a goal against a goalie or keepaway soccer), but no definite direction has already been chosen. What is already decided is that ILP should focus on high-level decisions, precisely because of its potential on human understandable models and potential use of symbolical data. One other direction being considered is the conjunction of ILP with other more conventional machine learning techniques, like the new relational reinforcement learning paradigm [8].

5 Conclusion

VitoriaRC was the fruit of a passion of a single person and is still a work in progress. Currently, the human team behind it is being augmented (all the team was until now a solo effort). It is true that right now the team is barely competitive, and certainly not able to win against the best teams, but is has potential. It should be noticed that the present binary does not reflect all the work that has been done (and has not been incorporated in the real simulation team).

This team contributes to Robocup, researching the possible use of ILP in it. We are confident that some results will be achieved in a short period of time, and that the team has a big progress margin, being able to be competitive by the time Robocup 2004 competition takes place. And if a good framework for ILP can be created, than a solid and original contribution will be made. But can ILP be applied with success to the simulation league? The future will tell…

References

1. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. *RoboCup: The Robot World Cup Initiative*. In Proceedings of IJCAI-95, Workshop on Entertainment and AI/ALife, 1995.
2. R. de Boer and J. Kok. *The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team.*. Master's Thesis. Universiteit van Amsterdam. 2001**.**
3. Luis P. Reis, Nuno Lau, Eugénio C. Oliveira. *Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents*. Balancing Reactivity and Social Deliberation in Multi-Agent Systems, M. Hannebauer, J. Wendler, E. Pagello, Eds. LNCS 2103, pages 175-197, Springer. 2001.
4. D. Kazakov and D. Kudenko. *Machine Learning and Inductive Logic Programming for Multi-Agent Systems*. In M. Luck, V. Marik, O. Stepankova, R. Trappl, Eds. Multi-Agent Systems and Applications, pages 246-270. Springer, 2001.
5. L. De Raedt. *A perspective on Inductive Logic Programming*. In K.R Apt, V.W. Marek, M. Truszczynski, D.Warren, Eds. The Logic Programming Paradigm, Springer, 1999.
6. K. Driessens, N. Jacobs, N. Cossement, P. Monsieur and L. De Raedt. *Inductive Verification and Validation of the KULRot RoboCup Team*. Proceedings of the Second RoboCup Workshop, 1998.
7. T. Matsui, N. Inuzuka, and H. Seki. *A Proposal for Inductive Learning Agent Using First--Order Logic*. In James Cussens and Alan Frisch, editors, Work-in-Progress Reports of ILP-2000, pages 180-193, UK, 2000.
8. S. Dzeroski, L. De Raedt and K. Driessens. *Relational Reinforcement Learning*. MLJ, 43:7-52, 2001.