

Essex Wizards 2002 Team Description

Huosheng Hu, Nikolaos Kalyviotis, Matthew Hunter

Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, UK
{hhu, nkalyv, mchunt}@essex.ac.uk

Abstract. This paper overviews the Essex Wizards 2002 team participated in the RoboCup 2002 simulator league. Three major issues have been addressed, namely adaptive strategic planning and communication, a generalized approach to position selection, as well as a real-time agent architecture.

1 Introduction

Up to now the simulation league continues to be the most popular event in RoboCup, in terms of the number of teams participated annually and the team strategies being adopted. In general, all teams competed in this league are faced with several major research challenges: multi-agent coordination, agent modelling, real-time performance and learning. In order to satisfy all the necessary timing constraints for simulated soccer agents, a multi-threaded implementation has been adopted in the Essex Wizards team so that the agents can perform various computations concurrently and avoid waiting for the slow I/O operations. Moreover the behavior-based approach plays a key role in building the Essex Wizards team [1]. A decision-making mechanism based on reinforcement learning enables co-operation among multiple agents by distributing the responsibilities within the team [3].

The focus of our Essex Wizards 2002 team is on adaptive strategic planning, generalized position selection and real-time agent architectures. We briefly outline our research focus here in terms of these three aspects.

2 Adaptive Strategic Plans and Communication

In the RoboCup domain, co-operation is the key to success. However, in order to achieve co-operation between the agents, planning has a key role to play. We have focused on the framework and design issues regarding strategic planning. The framework for our strategic plans consists of three major components, namely *Triggers*, *Actions* and *Abort* conditions [4]. The *Triggers* are used as signals to allow or forbid the actions or plans that are predefined. The *Actions* are a combination of low-level and high-level behaviors that are executed sequentially. The *Abort* condition is a safeguard to ensure the conditions of the environment are suitable for the strategic plan (SP) being executed.

Strategic planning is a relatively simple but very useful method. Although it has been used successfully only in the RoboCup environment so far, there is no reason why it can not be used in the other domains. Given that adequate knowledge of the domain is acquired and careful design and implementation of the strategic plans has been done, then the performance of a MAS (Multi-agent System) that uses those strategic plans can be increased significantly. Having a SP for a situation that happens frequently is a good idea. Having more than one SP for the same situation is even better. The only problem is that the agent should choose which one to execute. For this reason, how to implement more flexible and adaptive strategic planning has been investigated in our team [5]. More specifically, a set of SPs that are assigned to a strategic situation as Global Strategic Plan (GSP). A strategic situation can have a number of GSPs assigned to it, and a number of different SPs can belong to each one of the GSPs. For example, a strategic situation can have three GSPs (GSP1, GSP2, GSP3) and each one of them has three SPs.

Every GSP is associated to an execution time. When a GSP is decided at time T then we say that the decided GSP is in progress until $T + \text{execution time}$. For example if the execution time = 30 for GSP1, then if GSP1 starts (i.e. is decided) at time 100, it will be in progress until time 130. When a GSP is in progress then the agents can execute only the SPs that are members of that GSP. The SPs that are not members of any GSP can be executed only if no GSP is in progress.

A very important module of the GSP framework is the decision mechanism. The effectiveness of the Strategic Planning is directly related to the ability of the decision mechanism to make the right decisions. In other words if the performance of the decision mechanism is poor, so will be the performance of the Adaptive Strategic Planning. There is no single decision algorithm that can perform well in any environment. The framework of Adaptive Strategic Planning does not apply any constraints to the use of a specific decision mechanism algorithm. Therefore the issue of the algorithm is left open, since it depends on the environment that Adaptive Strategic Planning is used.

In RoboCup, communication plays a very important role since it can enhance dramatically the world model of each player. Therefore the more information is communicated between the players the better the team performance. The use of an effective communication model is imperative, however the effectiveness of any communication model in RoboCup is limited by the size of the message that can be sent (512 bytes). We have demonstrated that the use of encoding can maximize the amount of information and increase the capacity of each message at least 2 times. Moreover encoding also results in hiding the actual information contained in each message from opponent agents.

Since communication is not reliable in the RoboCup environment, a message can be lost and never be delivered to some or all the agents that it was sent for. A simple way to account for that is the repetition of the message. For example the agent that makes a decision regarding the execution of a GSP, can repeat the message a few times in order to make sure that the intended recipients have

indeed received the message. The period of time that the agent can repeat a message about the execution of a GSP is called "repetition period". Therefore if a decision is made at time T , then the agent can repeat the message until time $T + \text{repetition period}$. Another important point here is that the agent should include in the message the actual time that the decision was made. By doing that, the agents that receive a message will be able to understand if that message refers to a previous decision or to a new one.

3 Real-time Agent Architecture

To achieve real-time performance, we have adopted a modular approach in the overall agent implementation [3]. In such a design, there are five function modules, namely *Sensors*, *Behaviors*, *Actuators*, *World Model* and *Parameters*. Based on information from the *Sensors*, *Parameters* and *World Model* modules, the *Behaviors* module in each agent decides on the best course of action. This involves both low-level behaviors such as *moving* and *kicking*, and high-level ones such as selecting where to move to and which teammate to pass to.

At the lowest level any decisions made by the agent must be reduced to the core primitive actions provided by the server, i.e. *Kick*, *Turn* and *Dash*. In order to provide the options for high-level behaviors, extended primitives have been implemented such as *Advanced Kick* that moves the ball to a position where the desired kick can be made; *Move* that mixes turns and dashes to reach the desired location.

The high-level tactical behaviors are built on top of low-level primitive behaviors, and are currently implemented as a hybrid of Q-learning and rule-based decision-making, including *Intercept* that involves predicting the location of the ball for interception and moving to that location; *Clear Ball* that involves kicking the ball, using the *Advanced Kick* behavior; *Send Ball* that occurs when the agent attempts to get the ball to a position from which a team-mate can score; *Pass Ball* that generates a good pass based on the locations of teammates and opponents on the field; and *Position Selection* that examines the current view of the pitch and suggests a good place to move to, which is a non-trivial task, requiring information about the current role of the agent and the state of the pitch.

4 Generalized Approach to Position Selection

In the Robotic Soccer domain, position selection is often seen as a baseline case since it is often the last resort. If a player is positioning then that is usually because it has nothing better to do. Some teams use position selection in specific situations such as ball tracking and marking. This results in the position selection being tightly coupled to the rest of the agent, making it more difficult to experiment with. Our approach has been to provide a general interface for selecting a position whatever the situation while still allowing specific cases to be exploited where appropriate.

Three issues have been addressed in order to improve the position selection mechanism for the Essex Wizards 2002 team. The first issue was to improve individual behavior classes and reduce them to more general components. The second was to provide more flexible control within position selection. Finally the interface between the soccer agent and the position selection mechanism has been modified. In fact, a generalized behavior-based approach has been deployed to tackling the problem and the practical implementation of these ideas has been integrated into a working system [2]. This system uses a small number of interchangeable behaviors that are combined to perform rule-based position selection in real time.

The current positioning is purely rule-based, which can be tricked by opponent strategies that are different from those the rules were designed to deal with. If both teams use similar strategies they can interfere with each other in unexpected ways. More rules can be added to deal with different strategies, although detecting which strategy an opponent is using may be difficult. Although individual behaviors could be augmented with learning modules, a better alternative may be to learn the behavior tree, requiring that the tree be dynamically changeable. We have considered the position selection mechanism as a specialized programming language, and if it can be modified to allow the program to be altered while it is running it could find uses in other domains such as planning.

5 Conclusions and Future Work

The main features of the multi-agent system of our Essex Wizards 2002 for the simulated RoboCup competition are presented in this paper. The three major research issues of our team are addressed, namely adaptive strategic planning, generalized position selection, and real-time architectures. In the next stage of our research, we will investigate how to improve these features in order to maximize the team's performance and achieve efficient team collaboration.

References

1. C. Candea, H. Hu, L. Iocchi, M. Piaggio and D. Nardi, Coordination and Behavioural Issues in RoboCup, *International Journal of Robotics and Autonomous Systems*, Vol. 35, No. 2-3, pages 67-87, August 2001
2. M. Hunter and H. Hu, A Generalised Approach to Position Selection for Simulated Soccer Agents, *Proc. of the RoboCup 2001 International Symposium*, Seattle, Washington, 4-10 August 2001
3. K. Kostiadis, M. Hunter and H. Hu, The Use of Design Patterns for Building Multi-Agent Systems, *IEEE Int. Conf. on Systems, Man and Cybernetics*, Nashville, Tennessee, USA, 8-11 October 2000
4. N. Kalyviotis and H. Hu, A Co-operative Framework for Strategic Planning, *Proceedings Towards Intelligent Mobile Robots (TIMR'01)*, Manchester, 2001.
5. N. Kalyviotis and H. Hu, ASPF: an Adaptive Strategic Planning Framework in RoboCup, *Proc. of IASTED International Conference on Robotics and Applications*, Clearwater, Florida, USA, 19-22 November 2001.