Essex Wizards 2001 Team Description

Huosheng Hu, Kostas Kostiadis, Matthew Hunter, and Nikolaos Kalyviotis

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

Abstract. This article presents an overview of the Essex Wizards 2001 team participated in the RoboCup 2001 simulator league. Four major issues have been addressed, namely a generalized approach to position selection, strategic planning and encoded communication, reinforcement learning (RL) and Kanerva-based generalization, as well as the agent architecture and agent behaviours.

1 Introduction

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 [4]. Moreover the behaviour-based approach plays a key role in building the Essex Wizards team. A decision-making mechanism based on reinforcement learning enables co-operation among multiple agents by distributing the responsibilities within the team. The focus of our Essex Wizards 2001 team is on adaptive position selection, flexible strategic planning, multi-agent learning and real-time agent architectures. We briefly outline our research focus here in terms of these four aspects.

2 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.

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 511–514, 2002.

Three issues have been addressed in order to improve the position selection mechanism for the Essex Wizards 2001 team. The first issue was to improve individual behaviour 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 behaviour-based approach has been deployed to tackling the problem and the practical implementation of these ideas has been integrated into a working system [1]. This system uses a small number of interchangeable behaviours 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 behaviours could be augmented with learning modules, a better alternative may be to learn the behaviour 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.

3 Strategic Plans and Encoded 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 [2]. 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 behaviours 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 [3].

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.

4 RL and Kanerva-Based Generalization

RL (reinforcement learning) has been adopted in our behaviour-based decision making process since it provides a way to program agents by reward and punishment without needing to specify how a task is to be achieved [4]. Each time the agent receives sensory inputs, it determines the state of the environment and then chooses an action to execute. The action itself changes the state of the environment and also provides the agent with either a reward if it does well or a punishment if it does badly. The agent should choose actions that maximize the long-term sum of rewards. It should be noticed that the agents in our implementation not only have different roles and responsibilities, but also have different sub-goals in the team. Hence, every individual agent tries to reach its own goal state, and cooperation emerges when the goals of all agents are linked together. The ultimate goal, scoring against the opposition, becomes a joint effort that is distributed among team members.

The complexity of most multi-agent systems prohibits a hand-coded approach to decision making. The problem of learning in large spaces is tackled through generalization techniques, which allow compact representation of learned information and transfer of knowledge between similar states and actions. In large smooth state spaces, it is expected that similar states will have similar values and similar optimal actions. Therefore it should be possible to use experience gathered through interaction with a limited subset of the state space and produce a good approximation over a much larger subset. The combination of Kanerva coding and reinforcement learning has been investigated in order to build the K-RL multi-stage decision-making module [5]. The purpose of K-RL is twofold. Firstly, Kanerva coding is used as a generalization method to produce a feature vector from the raw sensory input. Secondly, the reinforcement learning component receives this feature vector and learns to choose a desired action.

5 Agent Architecture

To achieve real-time performance, we have adopted a modular approach in the overall agent implementation [4]. In such a design, there are five function modules, namely *Sensors*, *Behaviours*, *Actuators*, *World Model* and *Parameters*. Based on information from the *Sensors*, *Parameters* and *World Model* modules, the *Behaviours* module in each agent decides on the best course of action. This involves both low-level behaviours 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 behaviours, 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 behaviours are built on top of low-level primitive behaviours, 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* behaviour; *Send Ball* that occurs when the agent attempts to get the ball to a position from which a teammate 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.

6 Conclusions and Future Work

This article presented the main features of the Essex Wizards 2001 multi-agent system for the simulated RoboCup competition. The four major research issues of our team are addressed, namely adaptive position selection, flexible strategic planning, multi-agent learning and real-time architectures. In the next stage of our research, we will investigate how to improve these features in order to maximise the team's performance.

Acknowledgements This research is financially supported by the Essex University Research Promotion Fund DDP940.

References

1. 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
2. N. Kalyviotis and H. Hu, A Co-operative Framework for Strategic Planning, Proceedings Towards Intelligent Mobile Robots (TIMR) '01, Manchester, 2001.
3. 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.
4. 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
5. K. Kostiadis and H. Hu, K-RL: Kanerva-based Generalisation and Reinforcement Learning for Possession Football, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Hawaii, USA, November 2001