A Generalised Approach to Position Selection for Simulated Soccer Agents

Matthew Hunter and Huosheng Hu

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

Abstract. Position selection is a key task that must be carried out by a soccer-playing agent, but is often overlooked in favour of the more active tasks such as ball control. This paper examines the position selection implemented by the Essex Wizards team in the RoboCup Simulator league in recent competitions. The initial approach using task specific behaviours is firstly reviewed. The new approach is then addressed based on modular decomposition for flexibility. Implementation results are given to show the applicability.

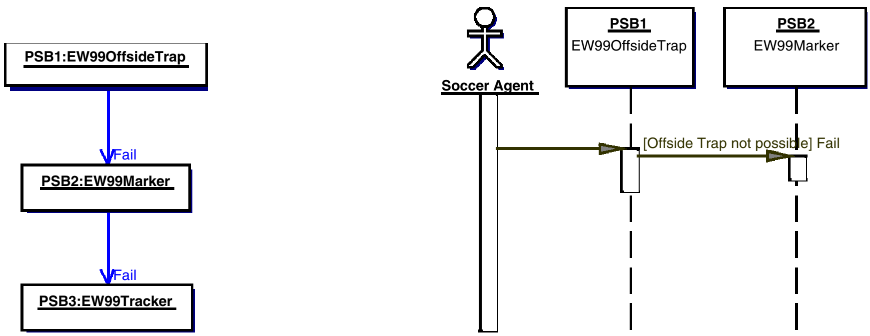
1 Introduction

In multi-agent systems, position selection plays a key role for achieving collaborative actions and undertaking complex tasks. The implementation of such systems is frequently focused on a specific domain. In other words, the selection of a good location is domain dependent in a multi-agent system. In the domain of puck gathering by a team of agents [6], for example, position selection involved picking a search area, and sticking to it. If one of the agents fails, a new working area is selected. Within a chosen area the agent is free to move in task specific ways, another layer of positioning. At the other end of the scale position selection may choose very specific locations that are only valid for short periods of time. For example, flocking, following and other team formations require frequent small changes in position [1][2][6].

Position selection is a general problem and therefore any approach should attempt to maintain that generality, while allowing modification for a particular domain. In order to achieve this, it is assumed that the action of position selection is independent of the action of moving. This abstracts away the details of the underlying agent movement and it gives the agent more control over how the position is interpreted. Position selection can present a new target as soon as the environment dictates, there is no need to explicity abort the previous movement.

Currently in the Robotic Soccer domain [5] position selection is done by a player when it has nothing better to do. Considerable effort is put into ball control, but good positioning can maximise chances for getting close to the ball. For example in [8] some mention is made of position selection for specific situations and solutions such as tracking and marking. This could result in the

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 380–386, 2002.



(a) (b)

Fig. 1. The failsafe mechanism allows several position selection behaviours to be chained together to obtain more complex behaviour.

position selection being tightly coupled to the rest of the agent making it more difficult to experiment with.

This paper examines the position selection implementation used by the Essex Wizards team in the RoboCup Simulator league in recent competitions. Section 2 reviews the initial approach using task specific behaviours. Section 3 examine the current approach and show the team in action. Conclusions are then drawn and future work discussed.

2 Task-Specific Decomposition

The version used in the RoboCup‘99 competition in Stockholm [3] introduced the concept of a Position Selection Behaviour (PSB), i.e. an entity responsible for establishing a target position. The term *behaviour* is used here to reflect the fact that externally each PSB is a sealed unit that simply provides a target position to the user. The soccer agent works at a more abstract level, for example by *defending* rather than specifically *marking*. Seven high-level behaviour classes (BC) were actually implemented for this version, each targeted towards a specific task[3]. A PSB is an instance of a behaviour class. The most important of the BC were the tracker group (Tracker, Ball Tracker and Marker) used defensively, and the support behaviour, based on SPAR [8]. Additional behaviours were used for Offside Traps and moving to fixed locations on the pitch.

All behaviours have an identical external interface so that they can be transparently exchanged for the soccer agent. This in turn lead to the failsafe mechanism which gave the behaviours some control. A chain of behaviours is constructed so that a result can be returned from one of the behaviours in the chain in case some of them fail. In the example shown in Fig. 1(a) a chain of three behaviours is constructed for the defenders, as soon as one PSB finds a result it is returned and the other behaviours are never activated as in Fig. 1(b). Similar chains are needed for the midfielders and forwards. The soccer agent carries out initial configuration of the chains, and in play selects the position using the chain for the current player role based on team formation and player number.

The position selection mechanism used in the Essex Wizards'99 team demonstrated that a relatively simple rule-based system could perform well in robotic soccer. However the task-specific nature of the behaviours themselves and the limited control made available through the failsafe mechanism still required that soccer agents take active part in position selection.

3 Modular Decomposition for Flexibility

To generalise the behaviours and improve the level of control provided, the new version of position selection has been developed to provide more behaviours, more flexibility and reduced commonality between behaviours.

3.1 Behaviour Generalisation

Common components were identified to form the basis of the first two groups of new BC, *Basic* and *Calculated*. In addition the failsafe mechanism was replaced by the *Control* BC group. The relationships are shown in Fig. 2.

The *Basic* BC group consists of, *Ball*, *Fixed*, *Goal*, *Home* and *MyPos*, which represent significant objects in the environment.

The *Calculated* BC group, is responsible for processing the results of one or more behaviours to produce a target location. *Combine*, mixes two positions. *Constrain* limits the result of a behaviour to a given area. *Interpolate* calculates a point that lies a given distance along the line between two positions. *Offset* relocates a position by a given offset. *Protected* restricts a position to be the home area of the player. The *RightAngle* BC was suggested as an advanced marking method [9], the object is to select a target position (T) such that the angle OTB is 90 ° where O is the opponent being marked and B is the ball. *SelectOpponent* uses a series of rules to find the position of an opponent relevant to positioning. This involves finding nearby opponents that are not near teammates. The final behaviour class in the calculated group is *Support*, which attempts to find a good place to receive a pass from the team member with the ball.

The *Control* BC group replaces the previous failsafe mechanism, which was only appropriate where a clear fail condition existed, e.g *SelectOpponent* cannot succeed if no opponents are nearby. On some occasions a behaviour may find a valid result, but an alternative would be better. For this reason control behaviours were introduced. Control behaviours select between one or more positions but do not change the selected position. *Congestion* tests the relative numbers of opponents and team members in a region. *Near* tests two positions to see if they are within a given distance of each other. *OurBall* tests which team has possession of the ball. *TestPosition* tests whether a position is within a particular area of the pitch. The most complex control behaviour class is *Role*, which selects between seven possible roles based on the player's home position.

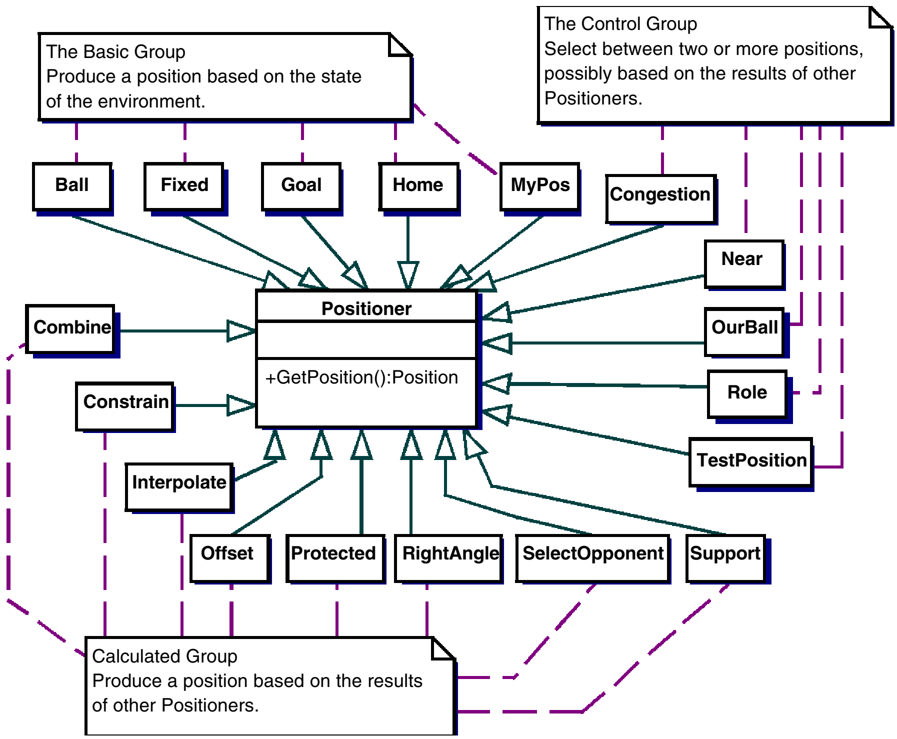


Fig. 2. There are three groups of Behaviour Class, Basic, Calculated and Control.

3.2 Behaviour DAGs

In order to carry out useful tasks it is necessary to combine the individual behaviours described above to form a Behaviour DAG. Figure 3 shows an object diagram of a simple behaviour DAG, rooted at the Position behaviour. Initially the Behaviour Registry contains a prototypical instance of each BC. When a behaviour is requested by name, it will be returned if it is present, otherwise it is generated recursively in terms of other behaviours from the configuration rules read in from a file. As each behaviour is generated, it is added to the registry for future use.

If a behaviour is created from another it inherits the behaviour class and current configuration. If a behaviour is configured to use another behaviour class then it will call on the sub behaviour as needed, for example an Combine behaviour needs two sub behaviours to mix. In this way the Behaviour Graph is built up,possibly consisting of many sub-behaviours.

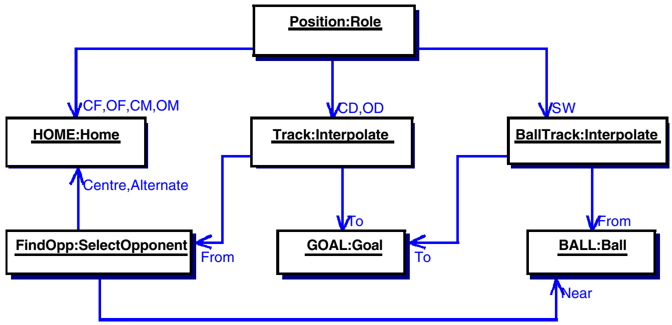


Fig. 3. Simple position selection DAG, rooted at *Position*. Forwards and midfielders simply move to their home location. Defenders track opponents and sweepers the ball.

3.3 Using Position Selection Behaviours

For positioning the team is split into three main groups, namely the *defenders*, *midfielders* and *forwards*. All the *defenders* work as one group, operating anywhere in the defensive zone, near the home goal and over the full width of the pitch. They try to get between the ball, opponents and the goal. Midfielders use the centre or one of the sides of the pitch but can move over much of the length, supporting defenders of forwards as needed. Forwards look for space away from each other and opponents, to create passing and shooting opportunities.

When the fixed plan mechanism was added [4] it became obvious that moving to locations was part of many plans. By having Fixed plans use Positioning they can take advantage of the positioning behaviours to adjust to an opponent.

4 Competition Results

The performance of our team has shown considerable improvement since the changes described, particularly defensively. An example is shown in Fig. 4 where the attacking team is preparing to take a kick in. This situation was initially intended to be a fixed plan, but players were more effective using the standard positioning and so the fixed plan was disabled.

A particular difficulty occurs with the attacking team's players 9 and 10 due to their closeness. Previously this lead to problems when the two attackers separated, only one of them would be marked. As it is even if the attacking players do get the ball none of them will have a good shot on goal.

5 Conclusions

This paper has considered the key role of position selection. A generalised behaviour-based approach to tackling the problem has been described along

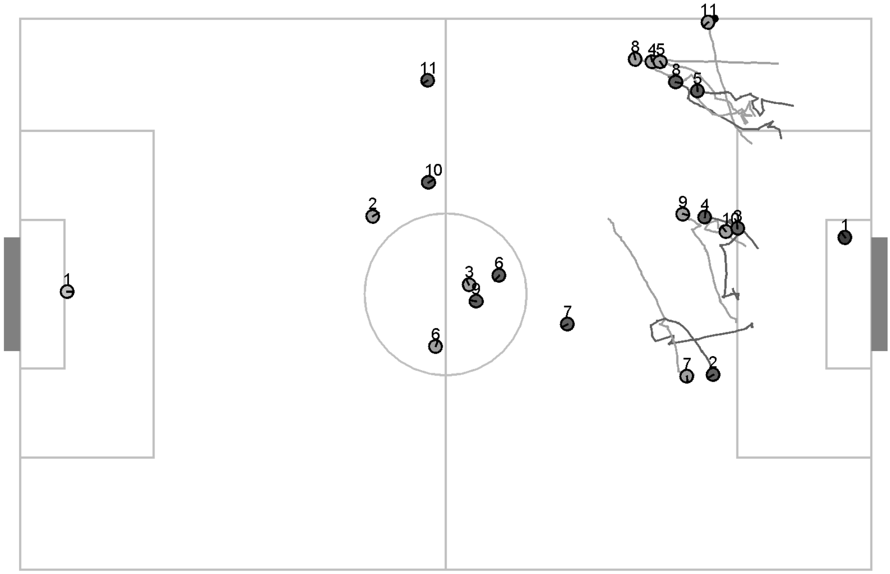


Fig. 4. Essex Wizards 2000 defensive marking in action.

with the practical implementation of these ideas into a working system. This system uses a small number of interchangeable behaviours that are combined to perform rule based position selection in real time. It is anticipated that the mechanism can be directly translated to other leagues with minimal changes.

Acknowledgements We would like to thank Kostas Kostiadis and Nikos Kalyviotis who have made contributions to this project. This project has been supported by RPF grant DDP940.

References

1. Desai J., Kumar V., Ostrowski J.: *Control of changes in formation for a team of mobile robots.* Proc. of 1999 IEEE International Conference on Robotics and Automation, 1556-1561, May 1999
2. Hu H., Kelly I., Keating D. and Vinagre D.: *Coordination of multiple robots via communication.* Proc. of SPIE'98 - Mobile Robots XIII, Boston, 1998, 94-103.
3. Hunter M., Kostiadis K., Hu H.: *A Behaviour-based Approach to Position Selection for Simulated Soccer Agents.* Proceedings 1st European RoboCup 2000.
4. Kalyviotis N. and Hu H.: *A Cooperative Framework for Strategic Planning.* Proc. of the 3rd British Conference on Autonomous Mobile Robotics and Autonomous Systems, Manchester, England, 5th April 2001.
5. Kitano H., Tambe M., Stone P., Veloso M., Coradeschi S., Osawa E., Matsubara H., Noda I., and Asada M.: *The RoboCup Synthetic Agent Challenge.* Proc. of International Joint Conference on Artificial Intelligence (IJCAI97), 1997.

6. Mataric M.J.: *Issues and Approaches in the Design of Collective Autonomous Agents.* Robotics and Autonomous Systems 16:(2-4) 321-331 DEC 1995
7. Schneider-Fontán M. and Mataric M.: *Territorial Multi-Robot Task Division.* IEEE Transactions on Robotics and Automation, 14:(5) 815-822 OCT 1998
8. Stone P.: *Layered Learning in Multi-Agent Systems PhD Thesis.* Carnegie Mellon University 1998
9. Williams D.: *University of Essex Football Club Coach, personal correspondence.*