Description of the Team
Mainz Rolling Brains
for the Simulation League of RoboCup 2000

T.Uthmann, C.Meyer, B.Schappel, F.Schulz
Institut für Informatik, Johannes Gutenberg-Universität,
D-55099 Mainz, Germany

Abstract

After reaching the semifinals in the 1998 and 1999 RoboCup competitions the Mainz Rolling Brains 2000 players are based on the 1999 ones but will include some new features such as different player types, the use of player models and more options for the players to decide how to work with the given data which makes players really different in mind.

1 Introduction

Real life soccerplayers are different from each other, both physically and intellectually, which either increases or decreases their effectivity concerning the different positions or roles they could hold during a match. Difference in physical variables like strength, speed or maximum power of kick is not (yet) implemented in the soccer simulator setting. So we try to make our players "different in mind" instead, i.e. we offer them a number of techniques to use and they pick the most useful ones, depending on their role and position. There are, of course, certain (low-level) skills that each player uses in the same way. We describe these first, then we give a list of the (high-level) techniques mentioned above and sketch how a player might choose from them.

One important innovation is the use of player models for the other players on the field which help the player to 'understand' his teammates and rank his opponents.

We do not give a detailed description of each module as the MRB (Mainz Rolling Brains) team for 2000 builds on the successful MRB 1999 implementation (We made the 1999 RoboCup ([4]) semifinals). We concentrate on this year's significant changes instead.

2 Player structure

This year's MRB soccer players are based on a three-layer concept as shown in figure 1. The lowest layer (called technical layer) is the same for each player. It deals with purely technical matters only (like server communication) and so provides an abstract interface for the layers lying above. The transformation layer is the collection of all skills and tools the player might use. It works with data provided by the technical layer and is influenced by the decision layer which represents the player's "brain".

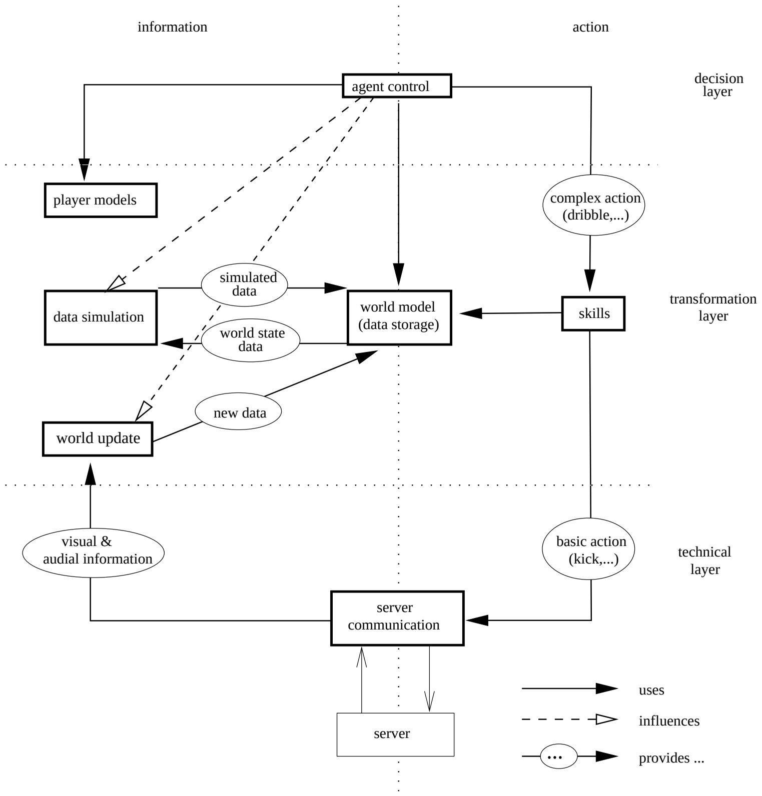

Figure 1: Layer concept of player structure

2.1 Worldmodel

MRB players of past years had a worldmodel that both stored data and updated with new information (cf. [1]). This year's philosophy claims strict distinction between these two matters. The only task left for the worldmodel is data storing. Data manipulation is done in two different modules as there are two situations when it has to be done: The player receives new information and carries out an UPDATE or the player has to decide what to do in the future which requires data SIMULATION. For both update and simulation there are different ways how to do it, and it should be up to the player to decide, depending on variables like his role, his position and the situation of the match.

Data update can be based either on visual information exclusively or can include information gained by other channels such as communication with the team members (e.g. a player who does not see the ball will listen closely to his teammates but a player dribbling towards the opposite goal might concentrate on what he is seeing (at least as long as data about the ball is concerned)).

Data simulation could make use of different algorithms for the calculation of variables which are difficult to control (like ball position and ball speed) or could make different assumptions on the movement of players.

2.2 Skills

The player has the possibility to choose among two skill levels. Low level skills correspond with the basic player commands like turn, kick, dash. High level skills are for example interception of the ball, dribbling or handling standard situations. Last year's player skills were coded by hand and made use of numerical algorithms for nonlinear optimization. These calculations are now to be improved by reinforcement learning techniques, e.g. find optimal parameters for dribbling to decide whether to trace the ball or to kick it and find an optimal position for kick off resp. kick in.

2.3 Player control

A hierarchical rule mechanism provides the skeleton for the player control. We started using such techniques in 1998 and improved them in 1999 (cf. [2]). In these years we concentrated on both stability and flexibility so that we don't have to deal with technical matters anymore but can include our new ideas like

- creating different rule trees for different player types like offender, midfield, defender, goalie
- taking into account player models of the team members and the opponents
- creating subtrees for standard situations
- reconstructing trees by AI algorithms (cf. [2])

The idea of giving players different abilities may be useless if there is no way of knowing if a decision for or against a certain technique makes sense. Basicly, there are two things a player should try to find out after having made a decision: Has it been successful or not? If the first question can not yet be answered: Will it be successful with a certain probability or not?

Answer to these questions is found via a learning model, based on the principle of the carrot and the stick. The main problem is to find out which things to reward (e.g. a goal, a successful pass or dribbling) and which not (e.g. a goal for the opponent, a pass intercepted by the opponent, bad positioning on the field). These observations will be done by the coach as he is the one with the best overview and the coach will pass on his results to the players.

3 Player models

Soccer is a team sport, and exact knowledge of one's teammates is essential for successful play. This year's MRBs will have the possibility to use player models for their teammates. These models are similar to the model the player uses for his own decisions. They are filled with data by visual information and player communication.

Player models for opponent players will also be implemented but are much more difficult to handle because there are less ways to get information and the opponents change from match to match (and the matches are short!). So these models will be of a much more sketchy type, trying to recognize general patterns of behaviour (like 'they prefer dribbling to passing' or 'they have a powerful kick routine so they try to score from a distance') rather than model single players in detail. A similar approach has been suggested by CMUnited ([3]).

4 Engineering of a multi agent system

Our experience of the last two years led us to the development of different agent types for special purposes. The 1999 team consists of two player types: goalie and field player. This year we are thinking about splitting up the field player into three parts: offensive, midfield and defensive player. The advantages are twofold. First of all the code of a specialist is not as complex as that of an universalist. This makes it easier to debug it. Secondly it is advantageous if different programmers are developing their own code. The varying approaches for similar problems produce competing solutions. Starting from a fixed platform and design one can choose from these solutions and integrate the best into the final agent.

To make debugging and evaluation easier we will develop a tool for replaying games in slow motion, combining the log player, our own world model visualization and evaluation tools.

Acknowledgements We wish to mention all the developers of the team (in alphabetical order): Axel Arnold, Alexander Fluhr, Manuel Gauer, Tobias Jung, Erich Kutschinski, Christian Meyer, Sebastian Oehm, Birgit Schappel, Ralf Schmitt, Holger Schneider, Frank Schulz, Götz Schwandtner and Patrick Wolf. In addition we wish to express or gratitude to F^3G.

References

[1] D.Polani, S.Weber, T.Uthmann. *A Direct Approach to Robot Soccer Agents.* In: M.Asada and H.Kitano, editors, *RoboCup-98: Robot Soccer World Cup II.* Springer , Berlin, 1999.

[2] D.Polani, T.Uthmann. *Between Teaching and Learning: Development of the Team 'Mainz Rolling Brains' for the Simulation League of RoboCup '99.* `http://www.ida.liu.se/ext/robocup/simul/Mainz-Rolling-Brains/teampage.html`

[3] P.Stone, M.Veloso, P.Riley. *The CMUnited-99 Simulator Team.* `http://www.ida.liu.se/ext/robocup/simul/CMUnited99/teampage.html`

[4] H.Kitano, Y.Kuniyoshi, I.Noda, M.Asada, H.Matsubara, E.Osawa. *RoboCup: A Challenge Problem for AI.* AI Magazine, 18(1):73-85, 1997.