

Team Description

Mainz Rolling Brains 2004 - 2D

Axel Arnold, Felix Flentge

Department of Computer Science
Johannes Gutenberg-University Mainz
D-55099 Mainz, Germany

1 Introduction

The Mainz Rolling Brains attend to the RoboCup Simulation League since 1998. Besides the changes in the SoccerServer since that time, our overall three-layer agent design (technical layer, transformation layer, decision layer, for details see [MRB00b]) remained roughly the same. Notably, the concept to combine various low level actions to so-called *Advanced Skills* [MRB00a] has proven itself very useful. Furthermore, we could keep the flexible, but simple design of our decision layer (introduced in 2000, see [MRB00a], [MRB01b]) over the last years. Because we always had about six up to ten developers, one main concern in all design decision was to have independent modules which could be handled separately by one or two developers. That also means to have reliable mechanisms to coordinate these different modules. Since we want to carry over these concepts to the 3D world, we are going to describe the main principles in the following. For details on other issues, e.g. world model [MRB02], self-localization [MRB01a], positioning [MRB02] [MRB01a], or communication [MRB00a], please have a look at our other papers.

2 Decision Layer Design

The decision layer currently consists of six modules and the *MasterControl* (see figure 2). The tasks of most modules should be clear by their name. *Ballcontrol* is responsible for dribbling and for getting the ball back from opponents. The *tactical module* is used for the administration of tactical information, e.g. player types and player roles (teammates and opponents) and for the decision where to look to. The *MasterControl* is responsible for choosing an action using the modules. The *MasterControl* decision cycle is:

1. *call feedback procedure for all modules*
The feedback procedures should allow the modules to judge the effects of their last decisions and actions and to adapt their evaluation scheme to the opponents.
2. *call evaluate procedure for all modules except tactical module*
The evaluate function returns a grade from a set of six discrete values which

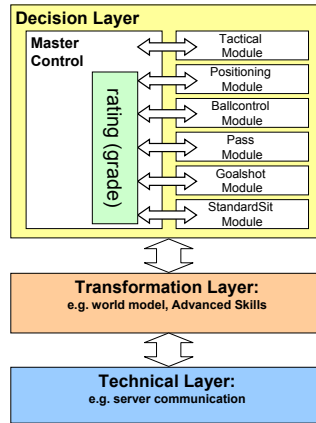


Fig. 1. Agent Design Overview

reflects the urge of the module to determine the action. For actions that takes more than one cycle there is the value *absolute* which ensures that the module can continue with its action.

3. call *act procedure of the module returning the best grade*
The module returning the best grade is called to send actions using the *Advanced Skills*. Ties are solved on a fixed policy basis.
4. call *turnHeadEvaluate procedure for all modules except tactical module*
The modules return where they want to look to (e.g. the ball or a particular player) and how urgent this is.
5. call *tactical module to decide on turn head*
The tactical module is called to decide where to turn the head to based on the wishes of the various modules.

The main problem with this approach is to balance the gradings of the modules, e.g. to balance the decision between dribbling and passing. Therefore we limit ourself to only six different grades, which should be easy interpretable. Currently these grading decisions are fine-tuned by hand. In principle it is possible to change some parameters effecting these decisions inside the module (in the *feedback* procedure) to adapt to an opponent during a game. This modular design makes it easy to exchange a single module (e.g. for using a different passing policy) or to split some of the current modules if they get to complex (e.g. dividing the *ballcontrol* into two modules, one for attacking the opponent, one for dribbling). It would be even possible to have different modules for the same task and to choose which one to use during the game depending on the opponent.

3 Advanced Skills

The *Skills* provide the agent with actions such as intercepting the ball or kicking it, as well as more complex combinations, e.g. dribbling. All skills are split up in two parts: The first one calculates the necessary low level (i.e. kick or dash) parameters to achieve the desired result. While doing that, additional information is generated like the actual ball velocity that will be obtained or, of more interest, how much cycles an intercept will need. The action is *not* yet sent to the server. Therefore, a module can test a number of different actions before deciding on which one to use. For example, it could test some desired positions to see which one could be reached first, or whether the player can reach the ball faster than an opponent. After the *MasterControl* has decided on the module to use the *Skills* are called by the respective module again to perform the action.

The *Skills* are not stateful, i.e. they do not use results from previous time steps, but always recalculate an action completely. This makes them robust against sensor errors. Internally, the Skills use a discrete optimization. As the physics of the 2D soccer simulation is discrete and all constraints are circles (e.g. dash range or kick range), finding the optimal kick or dash vector is an optimization problem on a convex area with finitely many corners, resulting in a finite set of possible optimal solutions which can be solved trivially.

4 Conclusion

Our modular design has turned out to be very useful and flexible to develop software agents for an environment with always growing requirements. The design also allows steps towards a more adaptive agent behavior as suggested above. Our future work will concentrate on the 3D-environment and we are looking forward to carry over our concepts to the 3D-soccer-simulation (for details, see the description of our 3D team).

References

- [MRB02] Flentge, F., Schneider, Ch., Wache, M.: Team Description. Mainz Rolling Brains 2002.
<http://www.rollingbrains.de/MRB2002.ps>
- [MRB01a] Arnold, A., Flentge, F., Schneider, Ch., Schwandtner, G., Uthmann, Th., Wache, M.: Team Description. Mainz Rolling Brains 2001. In: RoboCup-01. Robot Soccer World Cup V, Springer, Berlin Heidelberg New York (2002)
- [MRB01b] Flentge, F., Meyer, C., Schappel, B., Uthmann, Th.: Enhancing the Adaptive Abilities. Mainz Rolling Brains 2001.
<http://www.rollingbrains.de/MRB01-2.ps>
- [MRB00a] Schappel, B., Schulz, F.: Mainz Rolling Brains 2000. In: Stone, P., Balch, T., Kraetzschmar, G. (ed.): RoboCup 2000: Robot Soccer. World Cup IV. Lecture Notes in Computer Science, Vol. 2019. Springer, Berlin Heidelberg New York (2001)
- [MRB00b] Uthmann, Th., Meyer, C., Schappel, B., Schulz, F.: Description of the Team Mainz Rolling Brains for the Simulation League of RoboCup 2000.
<http://www.rollingbrains.de/MRB00.ps>