

# Virtual Werder 3D 2006 Team Description

Andreas D. Lattner, Steffen Planthaber, Carsten Rachuy, Arne Stahlbock,  
Ubbo Visser, and Tobias Warden

TZI, Center for Computing Technologies, Universität Bremen,  
PO Box 330 440, D-28334 Bremen, Germany  
`grp-vw3d@tzi.de`

**Abstract** This paper describes the current status of the Virtual Werder 3D team. Since last year's competition many efforts have been initiated by our team. A selection thereof is presented in this paper. Major changes include a new timing model, the use of a particle filter for the estimation of positions and velocities of moving objects at arbitrary times, and a new dynamic reposition skill based on Voronoi cells.

## 1 Introduction

This paper describes the current status of the Virtual Werder 3D soccer agent which is compliant with the RoboCup 3D simulator [13, 1]. Virtual Werder 3D participated in the previous two RoboCup competitions in Lisbon and Osaka [4, 8]. Since last year's competition many efforts have been initiated by our team and a selection of the changes is presented in this paper. As it has been identified that our previous agent did not exploit all possible options of the provided soccer server framework some fundamental changes w.r.t. effector control and world model updates have been accomplished. These changes have been necessary in order to obtain exact positional information about all dynamic objects and to allow for acting at a finer granularity between sensation messages sent by the server. Our system of behaviors and skills has seen only minor changes in comparison with the previous agent version.

The paper is organized as follows: The next section gives an overview of the basic architecture of the Virtual Werder 3D soccer agent. The subsequent sections present the timing model and the ball position prediction. The different behaviors and skills are then introduced before the paper closes with a brief overview of further efforts of our team.

## 2 Architecture

The world model – containing all relevant information for the agent's decisions – is stored in the knowledge base. Figure 1 shows the several components of the knowledge base and their interactions. The knowledge base gets its input from mapper and injector modules, which e.g. parse and filter the sensation messages or keep track of statistical information. A set of extractors provides functions

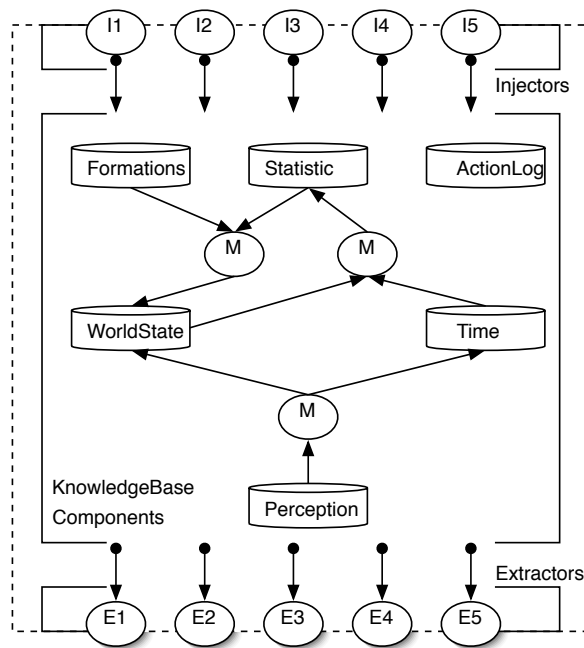


Figure 1. Knowledge base

for the different behaviors and skills in order to query the knowledge base for making decisions and identifying parameters of the skills. In the behavior module it is selected which skills should be used. A modular design for different types of behaviors allows for an easy adaption of the agent's behavior. The program's main loop runs in the **Agent** object and is mainly used to initiate the perception process and invoke the different behaviors.

The **KnowledgeBase** holds all relevant information about the current state, a number of previous and predictions of future world states. Here, we make a distinction between moving objects and static marks (such as flags, goalposts). Moving objects are the ball and the players of the different teams including the agent itself. The world model also provides information about both teams and the setup of the soccer field.

Every time a new sensation is received a particle filter (see e.g., [6]) computes values for different past, future and the current time steps by filtering a number of previously sensed positions. Besides handling noisy perception data the prediction of future positions can be used to estimate the actual positions (incorporating the delay) and future positions. Interpolation is used in order to compute positions between perception time points.

Basic strategies of the team involves the basic system of play (e.g. 4-4-2) and kickoff positions for the complete formation. As mentioned above statistical information about the evolution of the current game can be stored in the

knowledge base. A dynamic selection of the strategy allows for changing the strategy in a running match if certain conditions are satisfied (e.g., change to a more defensive formation if the opponent has scored more than two goals in succession).

### **3 Timing Model**

The realization of proper behaviors and skills in order to deal with various situations is an important aspect of a soccer playing agent. However, it is necessary to have accurate positional information of the dynamic objects in the scene in order to initiate actions at the appropriate time points. The agent receives delayed sensation messages every 200 ms, i.e., the time the sensation is received the information is not up to date any more. Another aspect is the possibility to place more than one action between two perceptions. SPADES allows the agent to request time notifies every 10 ms which can be responded by further action commands. In order to take advantage of the finer granularity for effector control it is necessary to provide the agent with (predicted) positions at the different time points.

The timing model of our agent updates the knowledge base every time a sensation is received. Here, the particle filter is used as mentioned above. After having processed the sensation the remaining time is used to run as many act cycles as possible based on predicted (and partly interpolated) positions of the dynamic objects. The delay of the sensation and action is incorporated in order to choose the activity on information which should be as close as possible to the real situation when the action command is executed in the effector.

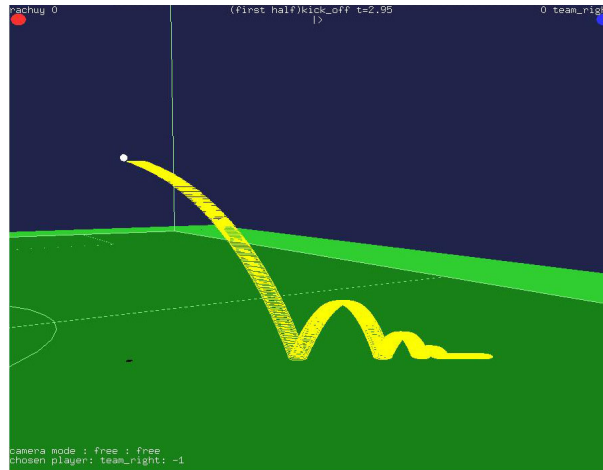
If it is expected that the next act cycle would overlap with a forthcoming sensation no further time notify is requested. In this case the agent waits for the next sensation message.

### **4 Ball Position Prediction**

As interaction with the ball is of special interest for different skills (e.g., kick and intercept in our case) and the ball also is the dynamic object that leaves the ground repeatedly it is very important to have accurate information about its position. In order to compute the ball's position various physical properties are taken into account, e.g., friction, gravity, loss of force when hitting the ground. The position for a designated time point is computed by an iterative calculation under consideration of the different forces. Figure 2 shows the predicted path of the ball with our modified log player.

### **5 Behaviors and Skills**

Behaviors of the agents are currently divided into keeper, defender, midfielder, and forward behavior. All these behaviors provide different behavior methods

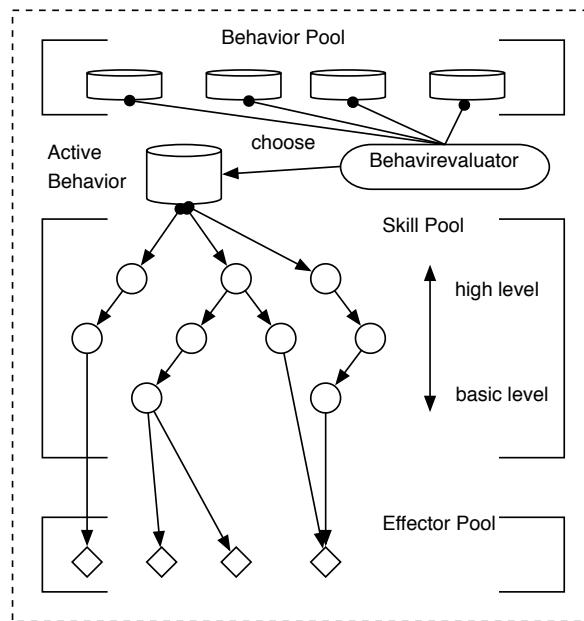


**Figure 2.** Prediction of the ball position

that can be used for the different situations in the game like corner kick, regular play, etc. The selection of the actual behavior is done by querying the world model using the extractors. Depending on the current situation the skill to be applied is selected. The skills themselves are responsible for getting information how to act in detail, e.g., where to move for repositioning. Figure 3 illustrates the interaction of behaviors, skills, and effectors.

The skills are divided into low-level and high-level skills. Low-level skills are the **Move**, **Kick**, **Beam** and – recently since communication is supported in version 0.4 of the 3D soccer server – the **Say** skill, and are motivated by the corresponding effectors of the agent. The **Move** skill enables the agent to move to a certain point. The travel speed and the speed at the goal coordinates can be given as parameters. The **Kick** skill allows the agent to kick the ball to a certain coordinate. Here, it is also possible to use different parameters how to perform the kick (speed and height). The **Beam** skill is just used off the regular game, e.g., in order to move the agents to their positions before the game. The **Say** effector has been recently implemented in order to enable communication between the players. Further details about how and what to communicate among the agents are currently under discussion.

The high-level skills are: **Cover**, **Score**, **Intercept**, **Pass**, **Reposition** and **Goalkick** and are based on the aforementioned low-level skills. **Cover** is used to stay close to an enemy agent, i.e., it tries to stay in a good position for intercepting the ball if the ball is passed to the cover target. **Score** selects the most promising position in the opponent’s goal and performs a kick to this position. The **Intercept** skill computes the best position where to catch the ball (“interception point”) and moves to this position. The **Pass** skill can be used to kick the ball to a team-mate or an area which is expected to be beneficial in the



**Figure 3.** Behaviors, Skills, and Effectors

current situation. **Pass** also provides the agents with the capability to dribble the ball given appropriate circumstances during a game.

**Reposition** is used to optimize the agent's position if he does not possess the ball. For repositioning a Voronoi-based segmentation of the field is performed (see illustration in Figure 4) [5]. Depending on the roles of the agent different responsibility areas are defined. Within each of these areas the corresponding agents try to optimally spread out in the space. Additional aspects like attraction or repelling of opponents or positions have influence on the overall positioning.

In order to avoid unsteadiness in the agent's behavior we implemented a double-layered commitment for decisions within the skills at the lower level and for behavior decisions in the higher level. This leads to faster actions of the agent and prevents the agent from oscillating between alternative actions.

A feature of our implementation is the possibility to dynamically adapt the agent. On the one hand this can be done via a configuration file in order to avoid a recompilation for applying different variants, e.g., for the formation and roles of agents, and on the other hand it allows for dynamically adapting certain aspects of our agent to the current situation like e.g., game status and the opponent's formation and strategy.

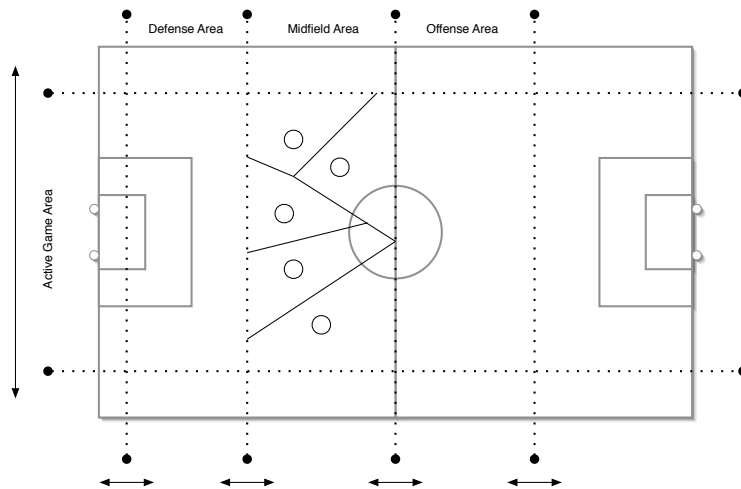


Figure 4. Voronoi-based repositioning

## 6 Further Efforts

There is a number of further efforts we are working on which are related to the RoboCup simulation league. We have integrated a framework for reinforcement learning into our agent where different variants like Q-Learning and SARSA can be used [16, 17, 14]. In our new agent framework learnable skills can be used like any other skill and thus, manually implemented and learned skills can be easily exchanged or combined. Reinforcement learning enables an optimization of skills and a more comfortable adoption to changes in the agent’s environment. It has been applied successfully in robotic soccer before by other teams (e.g., [11, 7]).

The application of a temporal pattern mining algorithm to data of RoboCup simulation matches is another research direction we are currently working on [9, 10]. The basic idea is to learn prediction rules from previous matches and to take advantage of such predictions in the behavior decision process.

Integrating the opponent’s intentions into the behavior decision process is the goal of another project that is funded by the German Research Council (DFG). It runs in the special program “Co-operative teams of mobile robots in dynamic environments” (SPP-1125) and focuses on the development of methods that enable agents to recognize and predict primitive and complex actions of opponent agents (e.g., [12, 2, 15, 3]).

An extension of the existing monitor to a more sophisticated log player with various useful functions for debugging (e.g., drawing circles, lines, and text on the field, rewinding the match, selecting different play speeds) is described in a separate paper for the development competition. A screenshot can be seen in Figure 2.

The work on further research topics has recently been started in the context of different diploma theses. Among others, ongoing research addresses the real-time analysis of soccer games, learning of evaluation functions, and creating statistical models of opponent's actions.

## Acknowledgment

The work presented here was partially funded by the *Senator für Bildung und Wissenschaft, Freie Hansestadt Bremen* ("FIP RoboCup", Forschungsinfrastrukturplan).

## References

1. Simulation league: The next generation. In D. Polani, A. Bonarini, B. Browning, and K. Yoshida, editors, *RoboCup 2003: Robot Soccer World Cup VII, LNAI 3020*, pages 458–469. Springer, 2004.
2. An egocentric qualitative spatial knowledge representation based on ordering information for physical robot navigation. In *RoboCup 2004: Robot Soccer World Cup VIII, LNAI 3276*, pages 134–149. Springer, 2005.
3. Towards a league-independent qualitative soccer theory for RoboCup. In *RoboCup 2004: Robot Soccer World Cup VIII, LNAI 3276*, pages 611–618, 2005.
4. T. Bogon, M. Kuhlmann, C. Niehaus, S. Planthaber, C. Rachuy, A. Stahlbock, U. Visser, and T. Warden. Description of the team virtual werder 3d 2004. Team Description Paper for the RoboCup-2004 in Lisbon, Portugal, 2004.
5. H. Dashti, Aghaeepour, S. Asadi, et al. Dynamic positioning based on voronoi cells (dpvc). In *RoboCup 2005 - Proceedings of the International Symposium Lecture Notes in Artificial Intelligence*. Springer, 2006, to appear.
6. D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
7. G. Kuhlmann and P. Stone. Progress in 3 vs. 2 keepaway. In *RoboCup-2003: Robot Soccer World Cup VII*, pages 694 – 702. Springer Verlag, Berlin, 2004.
8. M. Kuhlmann, A. D. Lattner, S. Planthaber, C. Rachuy, A. Stahlbock, U. Visser, and T. Warden. Virtual werder 3d 2005 team description. Team Description Paper for the RoboCup-2005 in Osaka, Japan, 2005.
9. A. D. Lattner and O. Herzog. Mining temporal patterns from relational data. pages 184–189, Saarbrücken, Germany, October 10 - 12 2005.
10. A. D. Lattner, A. Miene, U. Visser, and O. Herzog. Sequential pattern mining for situation and behavior prediction in simulated robotic soccer. In *RoboCup-2005: Robot Soccer World Cup VIII*. Springer Verlag, Berlin, 2006. To appear.
11. A. Merke and M. Riedmiller. Karlsruhe Brainstormers - a reinforcement learning way to robotic soccer. In *RoboCup 2001: Robot Soccer World Cup V*, pages 435–440. Springer, Berlin, 2002.
12. A. Miene, U. Visser, and O. Herzog. Recognition and prediction of motion situations based on a qualitative motion description. In D. Polani, B. Browning, A. Bonarini, and K. Yoshida, editors, *RoboCup 2003: Robot Soccer World Cup VII, LNCS 3020*, pages 77–88. Springer, 2004.

13. O. Obst and M. Rollmann. Spark – a generic simulator for physical multiagent simulations. In *Multiagent System Technologies – Proceedings of the MATES 2004, LNAI 3187*, pages 243–257. Springer, September 2004.
14. G. A. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical report, Engineering Department, Cambridge University, 1994.
15. T. Wagner, U. Visser, and O. Herzog. Egocentric qualitative knowledge representation for physical robots. *Journal for Robotics and Autonomous Systems*, 2005. To appear.
16. C. J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge, England, 1989.
17. C. J. C. H. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8(3-4):279–292, May 1992.