Virtual Werder 3D 2005 Team Description

Mirco Kuhlmann, 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 architecture, functionality, and the general research idea of the Virtual Werder 3D team. Since the RoboCup competition in 2004 a redesign of the soccer agent has been performed which lead to an improved world model and a modular structure for easy extensions and adaption of the agent's behavior. The work in progress addresses the learning of skills and the prediction of the enemy's intentions.

1 Introduction

This paper describes the current status of the Virtual Werder 3D soccer agent which is compliant with the RoboCup 3D simulator [7, 4]. Virtual Werder 3D participated in the RoboCup competition in Lisbon, Portugal, last year [1]. Since this event a redesign and refactorization of the soccer agent has been performed. This includes changes in the world model, low-level and high-level skills, and the basic behavior of the agent.

The paper is organized as follows: The next section gives an overview of the basic architecture of Virtual Werder 3D. The subsequent sections present the different skills and the behavior decision in the soccer agents. The paper closes with a section about the work in progress.

2 Architecture

The basic architecture of our system is shown in Figure 1. The communication to the soccer server is encapsulated by the `Messenger` class. Here, messages can be received from and send to the server. The control of the different effectors (currently `Beam`, `Drive`, and `Kick`) is done by separate delegates in the `Effector` module. Queries can be sent to the `WorldModelQuery` module. This module provides a number of useful queries which can be used by the different skills and behaviors. The skills module consists of low-level and high-level skills like moving, kicking, passing, etc. More details about the skills can be found in Section 4. The `Behavior` module uses world model queries and different skills for behavior decision and control. A modular design for different types of behavior allows for an easy adaption of the agent's behavior. All behaviors are managed by the `TacticRepository`. The `Agent` object itself is mainly used to invoke the different behaviors.

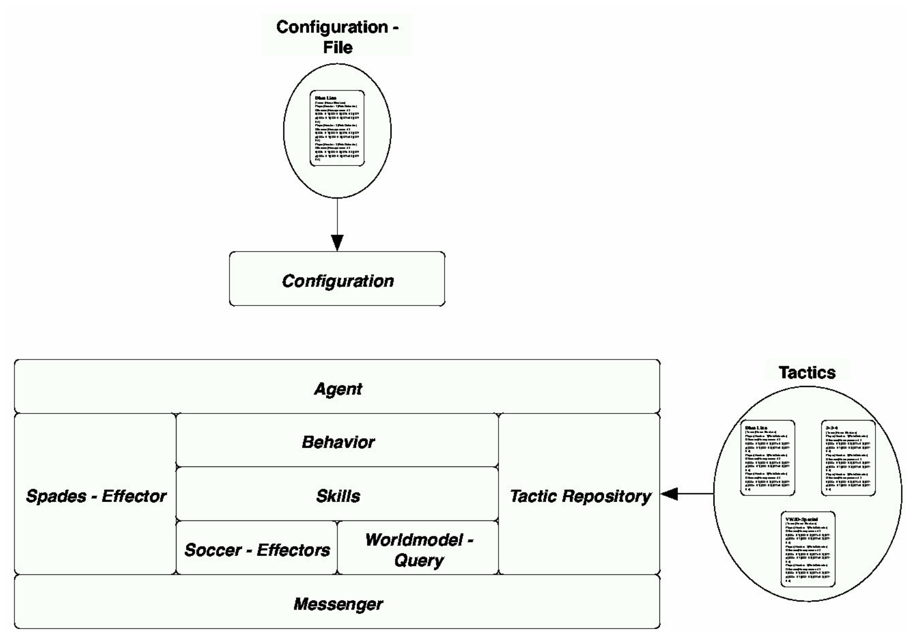

Figure 1. Virtual Werder 3D basic architecture

3 World Model

The `WorldModel` holds all relevant information about the current and a number of previous world states. Here, it is distinguished between moving objects and static marks. Moving objects are the ball, the players of the different teams, and, as a special case of a player, an object representing the current agent itself (called `Myself`). The world model also provides information about both teams and the soccer field.

A number of previous world states is kept in the memory of the world model. This is for instance useful for applying filtering techniques in order to compute the actual speed of objects. The computation of possible future positions of moving objects is also done here. Monte Carlo localization [8] can be activated for identifying the current position if needed.

As mentioned above the world model is queried by by the `WorldModelQuery` module. Queries can be rather simple, e.g., giving position or speed information for ball and agents, or more complex like identifying potential pass partners.

4 Skills

The skills are divided into low-level and high-level skills. The following two subsections give a short introduction.

4.1 Low-level Skills

Low-level skills are the `Move`, `Kick`, and `Beam` skills, 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 in order to perform the kick. The `Beam` skill is just used off the regular game, e.g., in order to move the agents to their positions before the game.

4.2 High-level Skills

The high-level skills are: `Clear`, `Cover`, `Dribble`, `GoalKick`, `Intercept`, `Pass`, and `Reposition`. These skills are based on the low-level skills introduced in the previous section.

`Clear` just kicks the ball into a direction which seems to be least harmful for the own team. `Cover` is used to stay close to an enemy agent. The `Dribble` skill is used if the agent should move on with the ball. `GoalKick` performs a kick to the enemy goal. 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. `Reposition` is used to optimize the agent's position if he does not possess the ball.

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.

5 Behavior

Behaviors of the agents are currently divided into `KeeperBehavior`, `DefenseBehavior`, and `OffenseBehavior`. All these behaviors provide different behavior methods 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 different world model queries in a decision tree. Depending on the current situation (identified by world model queries) the skill to be applied is selected.

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.

6 Work in Progress

Currently there are two major directions we are working on: Integrating the opponent's intentions into the behavior decision process and learning different skills. In the first case we co-operate with another project in our institute 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., [6, 9, 10, 2]). In the second case we integrate reinforcement learning approaches in order to learn different skills. This enables an optimization of skills and a more comfortable adaption to changes in the agent's environment. Reinforcement learning has been applied successfully in robotic soccer (e.g., [5, 3]).

Acknowledgment

The work presented here was partially funded by the *Senator für Bildung und Wissenschaft, Freie Hansestadt Bremen* ("FIP RoboCup", Forschungsinfrastrukturplan). We would like to thank our colleagues at the TZI for fruitful discussions. Especially, we want to express our gratitude to Tjorben Bogon, Cord Niehaus, and Thomas Wagner.

References

1. 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.
2. F. Dylla, A. Ferrein, G. Lakemeyer, J. Murray, O. Obst, T. Röfer, F. Stolzenburg, U. Visser, and T. Wagner. Towards a league-independent qualitative soccer theory for RoboCup. In *RoboCup 2004: Robot Soccer World Cup VIII, LNAI 3276*, pages 611–618, 2005.
3. 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.
4. M. Kögler and O. Obst. 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.
5. 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.
6. 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.
7. O. Obst and M. Rollmann. Spark – a generic simulator for physical multiagent simulations. In G. Lindemann, J. Denzinger, I. J. Timm, and R. Unland, editors, *Multiagent System Technologies – Proceedings of the MATES 2004, LNAI 3187*, pages 243–257. Springer, September 2004.
8. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.
9. T. Wagner and K. Hübner. 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.

10. T. Wagner, U. Visser, and O. Herzog. Egocentric qualitative knowledge representation for physical robots. *Journal for Robotics and Autonomous Systems*, 2005. To appear.