AT Humboldt & AT Humboldt 3D Team Description 2006 ⋆

Ralf Berger and Daniel Hein and Hans-Dieter Burkhard

Humboldt University Berlin, Department of Computer Science,
simteam@informatik.hu-berlin.de,
http://www.robocup.de/AT-Humboldt/index.shtml

Abstract. Since RoboCup 2004 we participate in both, the "classical" 2D-Simulation League as well as in the new 3D-competition. Last year, our team "*AT Humboldt*", placed second at German Open while "*AT Humboldt 3D*" placed 8th at RoboCup World Championship in Osaka.
For ten years we use our soccer agents as a research testbed for long-term deliberation and realtime reasoning, cooperation and coordination in multi agent systems, case based reasoning (CBR) aided decision making and evaluation of different methods of machine learning. In addition to participating in RoboCup competitions, we are successfully using our agent system in education for various aspects of multi agent issues.
In 2006 we again intend participate in both competitions. For the 2D Soccer Simulation we want to further extend the *double pass architecture* (DPA), evaluate new improvements in the RL-subsystem and increase the use of Case Based Reasoning support in different levels of the behavior hierarchy. The issues for the 3D agent will be implementation of biped robot models, further evaluation of different learning approaches for motion control, finalize the port of our *double pass architecture* into the new agent and the development of an integrated design and debug-tool.

1 Introduction

Our group's general research focus encompasses agent-oriented techniques, case-based reasoning, knowledge management, intelligent robotics, cognitive science and socionics. We investigate agent architectures and deliberation concepts that allow to build rational, scalable, dynamic and cooperative multi agent systems, with applications in e-commerce, medicine and industry. [Mün00,Min01,Hüb00]
We have found RoboCup to be an interesting and challenging domain for the development of new techniques and we participate in different leagues: the Soccer Simulation League (since 2004 in both, 2D and 3D competition), the Sony Four-Legged Robots League and the Humanoid League. Our main objective is the development of universal behavior architectures and concepts that are applicable to a variety of platforms in spite of their different specific demands. We

⋆ Our work is granted by the German Research Foundation (DFG) in the main research program 1125 "Cooperating teams of mobile robots in dynamic environments".

are members of the DFG (German research foundation) program "Cooperating teams of mobile robots in dynamic environments" [DFG-SPP-1125], and we focus on agent architectures.
A lot of our work in RoboCup is strongly linked to other research fields, for instance: Case Retrieval Nets [Len96] as a means for efficient and flexible retrieval in Case Based Reasoning, social modeling with multi agent systems or Belief-Desire-Intention models (BDI [Bra87,Rao91]) to go beyond emergent and allow cooperative behavior.
Ideas from soccer team development have often been fruitful for projects outside RoboCup, for example in cognitive robotics. Furthermore our work in RoboCup is a great teaching platform for practical exercises in our AI and robotics courses.
At present, the specification of the new simulation environment (3D simulator) for 2006 is not yet released and much work is still in progress. In this report we will give an overview of the most important concepts and interesting work that has been done till now. A detailed description of new issues and results will appear in the final team description paper.

2 3D Soccer Simulation

By introducing the new SoccerServer [Kög04], the Simulation League shifts its focus more towards real robot applications. Although in RoboCup 2006 one last time the 3D simulation will work with spheres, we begin to investigate biped motion control (fig. 1). The roadmap of this study consists of two primary steps:

1. Developing and implementing an appropriate agent model, using the SPARK simulation framework. This includes concidering architecture, dimension and weights of the general humanoid model, furthermore joint and sensor-number and -types.
2. Implementing an useful controller, that is applicable and indepent for further behavior control.

Regarding the high level behavior, we will finish porting the (*double pass architecture* to our 3D agent. Furthermore there are two new important features this year that have to be taken into account for adapting the agent: the ability of agent communication and the implementation of the offside rule in rcssserver3D.
One important point of agent developing is implementing useful and maintainable debug and development tools. Apart from agent designing we will focus on developing such necessary tools (fig. 2).
In the second international 3D competition at RoboCup 2005 we became 8th out of 32 teams. Apart from further improving the motion control, using recurrent neural networks and evolutionary algorithms for optimizing the robots movement towards steady and moving destination we advanced the high level planing of our agent.
For 2006 we intend to explore biped motion control, using RL-methods and evolutionary gate optimization.

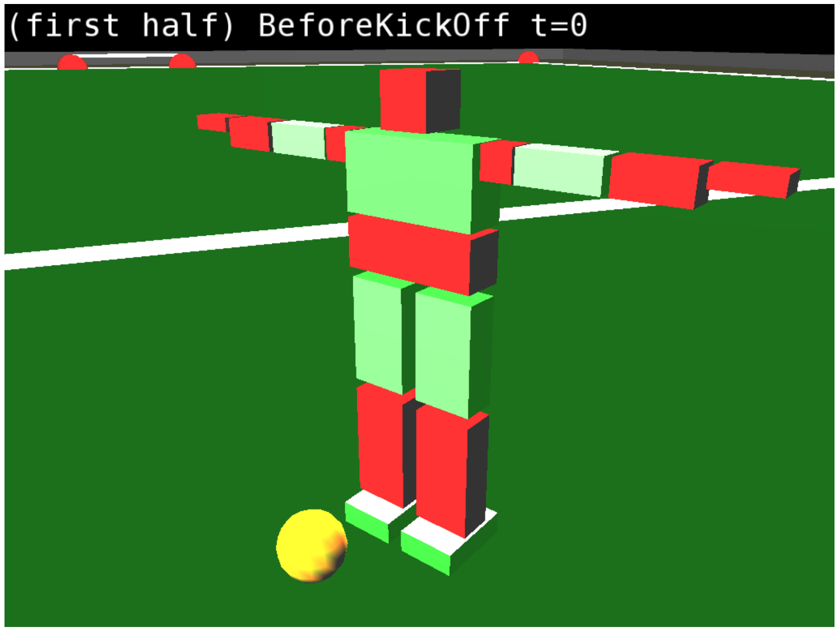

Fig. 1. Experimental humanoid robot model "Jana" running inside the SPARK simulation framework.

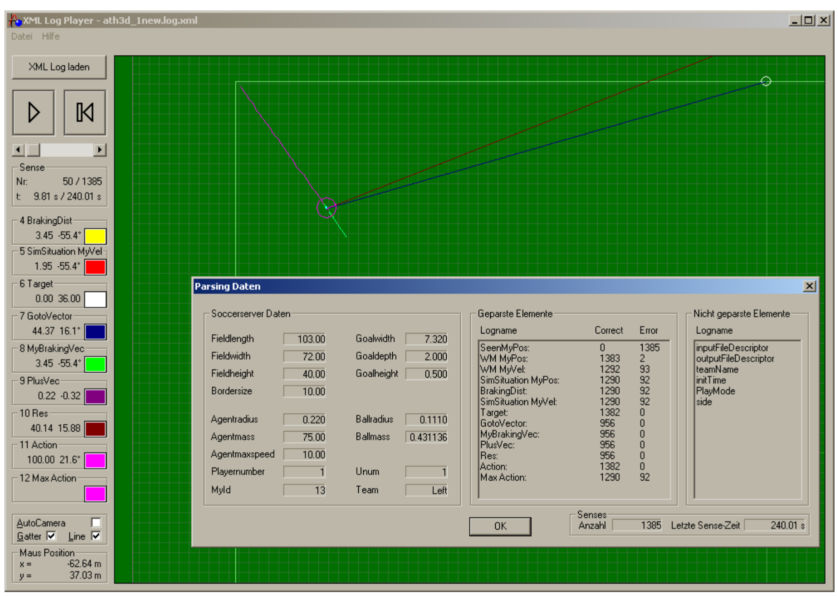

Fig. 2. Present logplayer for play back and graphical analysis of xml logfiles generated by ATH-3D.

3 Double Pass Architecture

An introducing description of the architecture is given in [Bur02,Bur05,Ber04]. Additionally there is a diploma thesis [Ber06] (in German), that includes a detailed specification and discussion of the *double pass architecture* .
Classical layered architectures show problems and limitations in highly dynamic environments which result from their fixed-layer structure and most of all from the coupling between behavior levels and the control directives. The *double pass architecture* does not have these limitations and is able to fulfill the following requirements:

- Time independent long-term deliberation of complex behavior with free time horizon
- Least commitment of all possible data to execution time
- Scalability in number of behavior options
- Scalability in the timing resolution and computational expense
- Realtime reaction capability on all behavior layers
- Persistency of high level individual and cooperative goals
- Control of coordinated behavior involving more than one player

We will only enumerate the key-concepts of our approach:

- Hierarchical organized behavior levels
  Extensive individual or cooperative actions can be modeled as a combination of more elementary actions, whereby combination can mean either choice or concatenation. All these sub-behaviors can further be described as a combination of other behaviors. The result spans a tree of options with abstract options near the root, like 'play soccer', and basic actuator commands, like 'kickToPos', at the leaves.
  Instead of a fixed number of layers, an arbitrary number of layers is permitted in the tree, whereby the different layers itself do not have a predefined abstraction model but every node reflects a certain context. This context can be determined e.g. by a class of world situations, a social goal, a local role assignment or an individual mental state of an agent.
  Evaluating only the possible actions within the context of one node instead of comparing all possibilities of a certain abstraction layer reduces the complexity of evaluation (local evaluation vs. global evaluation methods) and allows for specific decision making. Having this tree of possible behavior modes for every single agent, it can also be used by other agents as a common knowledge pool which makes cooperation without negotiation much easier.
- Decoupling of behavior and control
  Our approach is based on a strict separation between long-term deliberation and reactive execution (both considered on all levels), whereby all layers are invoked in every process cycle. For this reason the double pass architecture uses two independent top-down passes:

Deliberator-Pass
The deliberator performs long-term planning[1] to prepare and monitor behavior according to individual and social goals and persistent strategies. The deliberators main task is to choose the goals and to prepare all the context of the nodes which is necessary to decide how to realize these goals according to the current situation. Deliberation starts in the root-option and evaluates available suboptions by analyzing current or subsequent game-situations to determine their associated utilities. To avoid recursion, the behavior tree has to be organized in a way that the evaluation mechanisms can be performed locally in the context of the current node. The result is a pre-arranged partial plan – a set of evaluated options in the tree, that corresponds to desires and intentions in the BDI-methodology. This plan is continuously updated and completed as time goes on. Additionally, the deliberator provides alternative options/plans that are instantly available if an exception occurs at execution time. The deliberator is independent from the actual run-time demands; it has to be ensured however that at any time enough information is prepared for execution of sensible actions.

Executor-Pass
The executor generates the reactive actuator commands that will fulfill the goals selected by the deliberator. The main tasks are checking the options consistency and transition conditions and resolving of symbolical data based on the most recent sensory information (least commitment). The executor is called whenever a timer component decides that it is necessary to perform an action. Along this way the executor checks all the pre-, post- and break-conditions, resolves abstract parameters to actual values and collects control instructions that are generated by the deliberator or are steady components of the tree (e.g. communication strategies). Based on the preparatory work of the deliberator, the executor has to perform only a minimum of computational work (the data that has been left open for least commitment) and thus can be delayed to get the latest possible perception.

– Mental models for cooperation
  It is well known, that a kind of persistency is necessary for complex cooperation between agents which cannot negotiate their behavior because of communication constraints and which cannot deduct others behavior because of incomplete world knowledge. Explicit mental states are a particularly efficient way to achieve persistency. Normally these states include knowledge of former decisions, therefore they reduce the space and complexity of following evaluations and ensure stable decisions.
  The mental states of our agents include (apart from the worldmodel) commitments to individual or cooperative long-term and short-term behaviors and strategic modes, the global role of the agent itself and all teammates, the task association in local cooperative behaviors and the progression state of current plans.

[1] not meant as the classical term of planning or means-end reasoning, but the preparation of a set of intentions, which can generate a long-term behavior

Flow of data and control compared to classical 2–pass layered architectures is shown in figure 3.

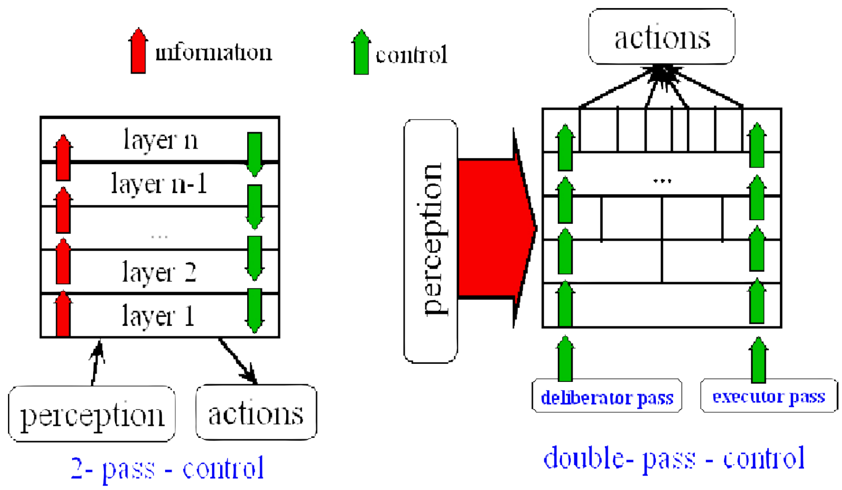

Fig. 3. flow of data and control for 2–pass and double pass architecture

An overview of the architecture components and its interactions is shown in figure 4.
Within the last years we continuously enhanced the architecture specification, added new control elements and improved the underlaying algorithms with respect to scalability and real-time capability. Furthermore we had to define an universal and context independent meaning and processing of the option concept.
At RoboCup 2005 we successfully used the current version of the double pass architecture in our soccer agent "*AT Humboldt*" and a little simplified version in "*AT Humboldt 3D*". This allowed us the flexible application of various methods for planning and selecting individual and cooperative behavior modes on different levels of abstraction. The option trees we used consisted of up to 350 options and more than 3400 control elements in at most 9 hierarchical levels for every single agent. Thus these models are about ten times as extensive then the ones we used in 2002. Nevertheless we could ensure real-time execution behavior even during time consuming deliberations. The average runtime of the executor process was about 1ms.
For 2006 we plan to improve the use of role concepts within the option-tree as well as the use of explicit cooperative commitments, Furthermore we will continue extend the option trees to allow for more specific decision making and more complex behavior. Additionally we just began to extend the possibilities of visualizing the behavior specifications.

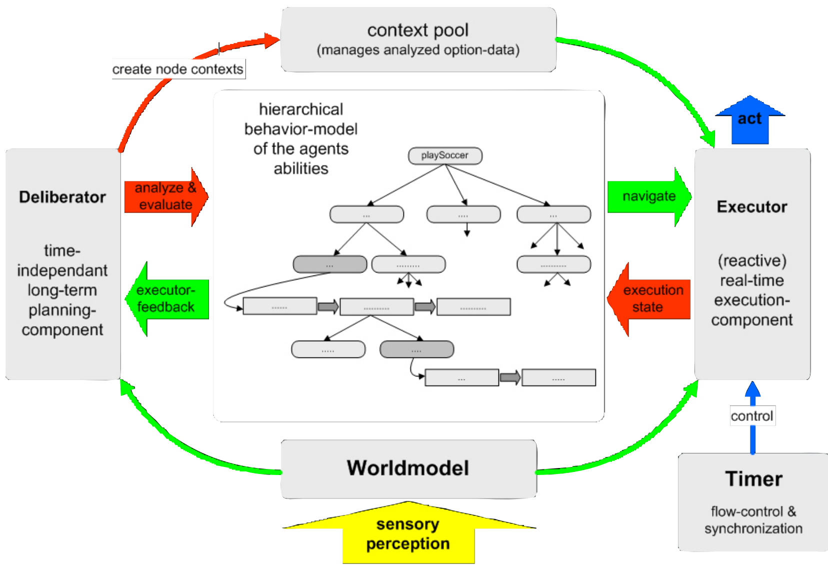

Fig. 4. basic interactions between the components of the architecture

4 Machine Learning

AT Humboldt successfully used *case based reasoning* (CBR) techniques [Ber04] in the decision making process since 2003. The case database was automatically build up with a set of logfiles and covered a relatively simple decision task for the goalie. This year we plan to integrate a CBR-based decision support system for describing and evaluating standard situations and for making strategic decisions on the upper levels of the behavior tree. The preparative theoretical work is currently done in a diploma thesis. Our first example is a one-two-pass. At the moment we build up the cases in a semi-automatic process from logfiles of previous RoboCup events. For the future we also want to use descriptive examples and training models as a kind of case base. First experiments have shown that the challenging task is finding a suitable similarity and relevance measure as well as a highly efficient case-retrieval method. Currently it seems that statistic approaches in combination with case retrieval nets will be powerful means for this problems.

In 2004 we started using methods of reinforcement learning (RL) for improving low-level and mid-level agent skills. In a diploma thesis [Gol05a], we developed an universal and comprehensive RL-library [RL++]. The agent's dribble-skill is now about twice as fast and even more failsave that the handcoded one. Interesting aspects of the actual learning method were the use of action evaluation/selection for very fast skill execution and the evolutionary selection of suitable meta-actions and tile-codings for modeling the problem. Furthermore

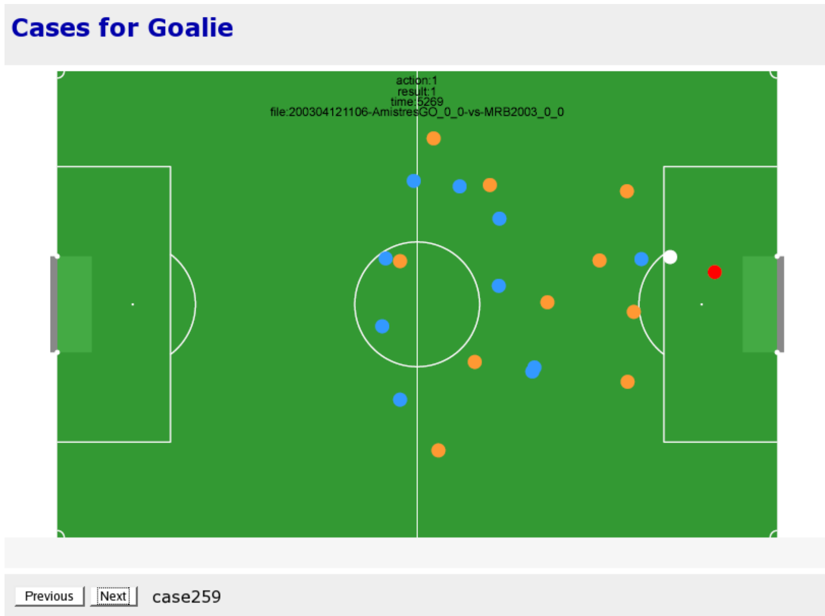

Fig. 5. graphical case representation (without similarity- and relevance grid) from CBR-supported goalie module

we've made improvements in adopting the learning scenarios to work with the heterogeneous players.
For 2006 we plan to extend the learning scenarios to have more learned agent skills. Additionally we already did and will further do experiments on using statistical methods for automatically finding near optimal RL-configurations for the action space, the state space and the state organisation (tile coding configuration). This will enable us to have fully automated runs of new learning experiments without constant supervising. The theoretical fundamentals of this approach will possibly be researched in a diploma thesis.

5 Outlook

While much work for 2006 is still in progress or in a conceptual stage, a detailed description of new results will appear in the final team description paper. For AT Humboldt 2006 / AT Humboldt 3D we plan to achieve the following:

- Optimizing (3D) robot motion with evolutionary or RL-methods
- Porting the full-featured *double pass architecture* into the 3D-agent
- Implementing a debug- and behavior development tool for the 3D-agent
- More sophisticated use of CBR supported decision making

– Describing standard situations and moves as cases (e.g. one-two)
– Extending the *double pass architecture* with capabilities for better asynchronous process management
– Enhancements to the visualization of the behavior-tree
– Automatical parameter optimization for the Reinforcement Learning Module

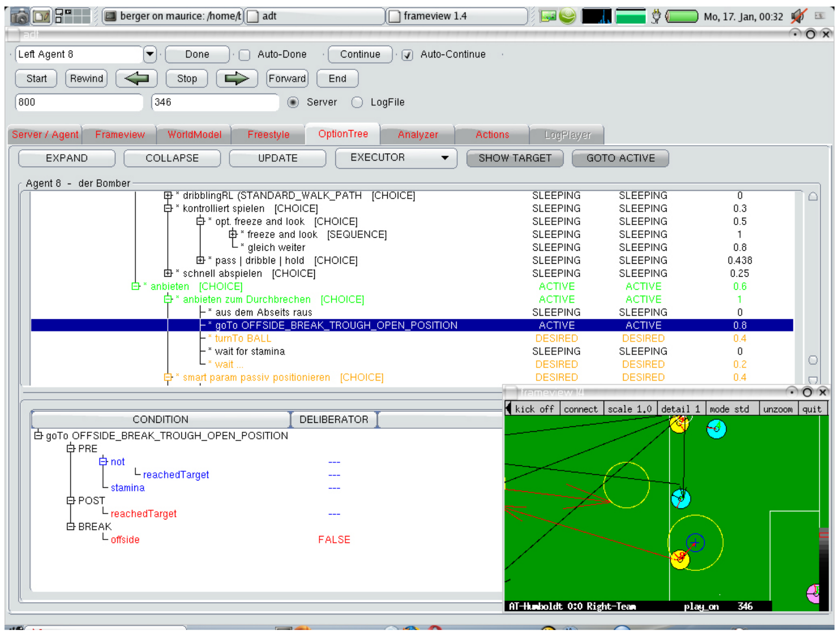

Fig. 6. development tool for the 2D-agent (ADT) showing a live fragment of the behavior tree and the internal state of the double pass architecture

References

Ber04. R. Berger, M. Gollin and H.-D. Burkhard. *AT Humboldt 2003 – Team Description*. In *RoboCup 2003 - Proceedings of the International Symposium*, Lecture Notes in Artificial Intelligence. Springer: 2004.

Ber06. R. Berger. *Die Doppelpass-Architektur – Verhaltenssteuerung autonomer Agenten in hochdynamischen Umgebungen*. Diploma Thesis, Institut für Informatik, Humboldt-Universität zu Berlin: 2006.

Bra87. M. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press: 1987.

Bur02. H.-D. Burkhard, J. Bach, R. Berger, B. Brunswiek and M. Gollin. *Mental Models for Robot Control*. In M. B. et al. (ed.), *Advances in Plan-Based Control of Robotic Agents*, vol. 2466 of *Lecture Notes in Artificial Intelligence*, pp. 71–88. Springer: 2002.

Bur05. H.-D. Burkhard. *Programming Bounded Rationality*. In *Proceedings of the International Workshop on Monitoring, Security, and Rescue Techniques in Multiagent Systems (MSRAS 2004)*, pp. 347–362. Springer: 2005. To appear.

Gol05a. M. Gollin. *Implementation einer Bibliothek für Reinforcement Learning und Anwendung in der RoboCup Simulationsliga*. Diploma Thesis, Institut für Informatik, Humboldt-Universität zu Berlin: 2005.

RL++. M. Gollin. *webpage of RL++ – open source C++ library for Reinforcement Learning* [online]: 2005. Available from: http://sourceforge.net/projects/rl-pp/.

Hüb00. A. Hübner, M. Lenz, R. Borch and M. Posthoff. *Last-Minute Travel Application. AI Magazine*, vol. 21(4):pp. 58–62: 2000.

Kög04. M. Kögler and O. Obst. *Simulation League: The Next Generation*. In D. Polani, A. Bonarini, B. Browning and K. Yoshida (eds.), *RoboCup 2003: Robot Soccer World Cup VII*, vol. 3020 of *Lecture Notes in Artificial Intelligence*, pp. 458 – 469. Springer: 2004.

Len96. M. Lenz and H.-D. Burkhard. *Case Retrieval Nets: Basic Ideas and Extensions*. In *Proceedings of the 20th Annual German Conference on Artificial Intelligence (KI '96)*, vol. 1137 of *Lecture Notes in Artificial Intelligence*, pp. 227–239. Springer: 1996. ISBN 3-540-61708-6.

Min01. M. Minor and M. Lenz. *Textual CBR im E-Commerce. Knstliche Intelligenz*, vol. 1:pp. 12–16: 2001.

Mün00. I. Münch and G. L. von Trzebiatowski. *ChariTime - Concepts of Analysis and Design of an Agent-Oriented System for Appointment Management. Fundamenta Informaticae*, vol. 43(1–4):pp. 215–226: 2000.

Rao91. A. S. Rao and M. P. Georgeff. *Modeling Rational Agents within a BDI-Architecture*. In J. Allen, R. Fikes and E. Sandewall (eds.), *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pp. 473–484. Morgan Kaufmann: 1991. ISBN 1-55860-165-1.

DFG-SPP-1125. *webpage of DFG main research program 1125* [online]. Available from: http://www.ais.fraunhofer.de/dfg-robocup/.

ATH-web. R. Berger (ed.). *webpage of AT Humboldt Soccer Simulation Team* [online]: 2006. Available from: http://www.robocup.de/AT-Humboldt/index.shtml.