AT Humboldt & AT Humboldt 3D Team Description 2005 *

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. In RoboCup 2004 we participated in the "classical" 2D-Simulation League as well as in the newly introduced 3D-competition. Our team, "*AT Humboldt*", placed 10th while "*AT Humboldt 3D*" placed second.
Like in the past years we used 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 2005 we intend to again participate in both competitions. For the 2D Soccer Simulation we want to further improve and extend the *double pass architecture* (DPA) and explore the use of Case Based Reasoning support on different levels of the behavior hierarchy. The issues for the 3D agent will be further evaluation of different learning approaches for motion control and first steps to port the *double pass architecture* into the new agent.

1 Introduction

Our groups 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 two leagues: the Soccer Simulation league (since 2004 in both, 2D and 3D competition) and the Sony four-legged robots 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 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.

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

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 simulated robots (3D simulator) for 2005 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 from 2004 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. Furthermore there are two important conceptual changes that have to be taken into account for agent design.

1. Three-dimensional world modeling
   By simulating the game of soccer three-dimensional, the set of tactic and strategic behavior possibilities will drastically increase. In the 2D-world the actual gameplay takes place within a very small area that is determined by the maximum reliable pass distance. In the 3D-world, passes over opponent players are possible, which increases the relevant playing area and the number of involved players. Exploiting this possibility for tactical moves or strategic gameplay can not be achieved by individual decisions or a global positioning method. Instead, intended cooperation and context-based goal-driven decision making become more important.
2. More realistic time modeling
   The continuous event-based timing model [Ril03] of the new SoccerServer goes far beyond the 100ms steady-state discrete old world simulation. The thinking time of each agent will determine the amount of time the world will advance. As for real robots, immediate reaction on changing situations is sometimes necessary because the world is moving on. The challenging task is that these actions should consider the committed individual and social goals as much as possible. In addition perception and acting will be possible in much shorter time-intervals than the duration of the agents reasoning-process. However, it is not necessary to shorten the agents thinking time, but to decouple perception, thinking and acting.

Our *double pass architecture* was developed with these requirements in mind, so we will use it in the new agent framework as soon as possible.

In the first international 3D competition at RoboCup 2004 we became runner-up. We achieved this due to our experience in agent design and RoboCup. We

implemented the new agent from scratch (not on top of "agenttest") based on the structural model of our 2D agent, which has turned out to be a real advantage during the competition. Furthermore we spent a lot a work for motion control. Experiments were conducted using recurrent neural networks and evolutionary algorithms for optimizing the robots movement towards steady and moving destination.
For 2005 we intend to converge the architectural design of the agent to the *double pass architecture* and do further experiments of motion learning with evolutionary algorithms and (fuzzyfied) RL-methods.

3 Double Pass Architecture

A fundamental description of the architecture is given in [Bur02,Bur05,Ber04]. 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 effi-

[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

cient 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.

Flow of data and control compared to classical 2–pass layered architectures is shown in figure 1.

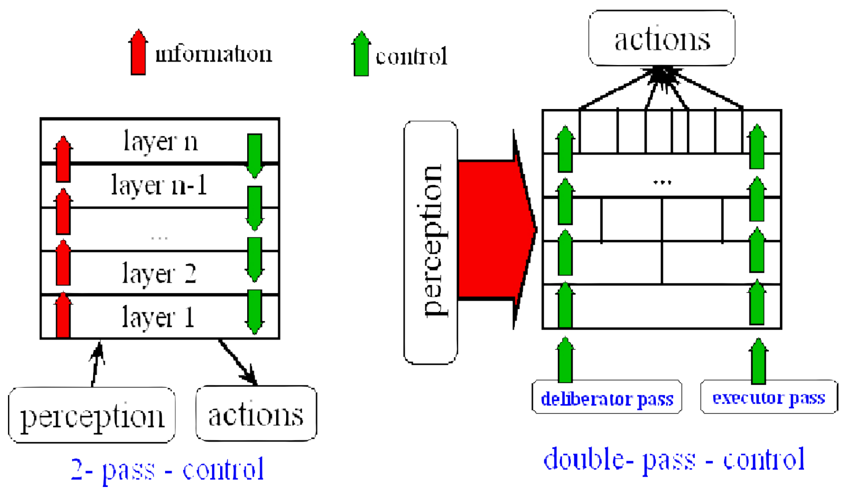

Fig. 1. flow of data and control for 2–pass and double pass architecture

An overview of the architecture components and its interactions is shown in figure 2.
Within the last year 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 2004 we used the double pass architecture successful in our soccer agent "*AT Humboldt*". 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 340 options and more than 3000 control elements in at most 9 hierarchical levels for every single agent. For the first time global roles, dynamic role assignments as well as explicit cooperative commitments were integral parts of these behavior

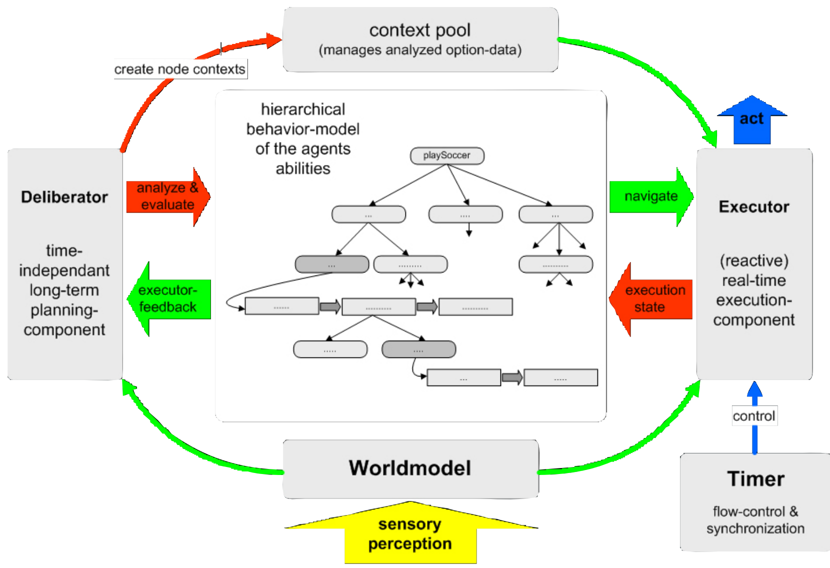

Fig. 2. basic interactions between the components of the architecture

models. 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 2005 we plan to integrate some experimental enhancements to the architecture that will allow even more complex but also more efficient pre-planning of strategic and tactical behaviors. Furthermore we will continue extending the option trees to allow for more specific decision making.

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 make use of CBR for describing and evaluating standard situations and for making strategic decisions on the upper levels of the behavior tree. Therefor we have to build up the cases manually from descriptions, examples and training models. Last years experiments have shown that the challenging task is finding a suitable similarity and relevance measure as well as a highly efficient case-retrieval method. For both tasks we think qualitative situation attributes will be an important means.

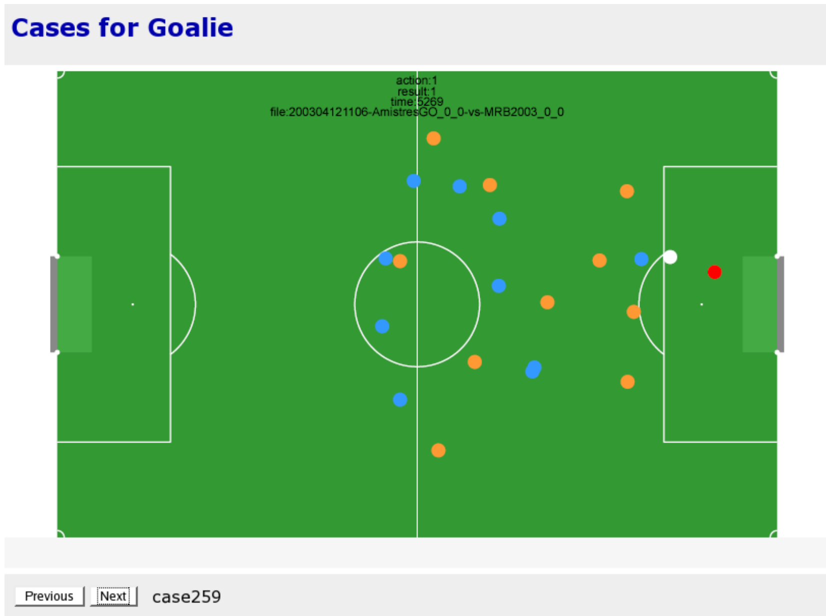

Fig. 3. graphical case representation (without similarity- and relevance grid) from CBR-supported goalie module

In 2004 we started using methods of reinforcement learning (RL) for improving mid-level agent skills. In a diploma thesis, we developed a 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.
For 2005 we plan to extend the results to more agent skills. Furthermore we will do some experiments on using reinforcement learning for optimizing evaluation functions on different levels of the behavior hierarchy.

5 Outlook

While much work for 2005 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 2005 / AT Humboldt 3D we plan to achieve the following:

- Optimizing (3D) robot motion with evolutionary or RL-methods
- Much more sophisticated use of CBR supported decision making
- Describing standard situations and moves with a uniform case format
- Integrating key aspects of the double pass architecture into the new 3D-agent
- Extending the double pass architecture with capabilities for efficient long-term pre-planning and asynchronous process management

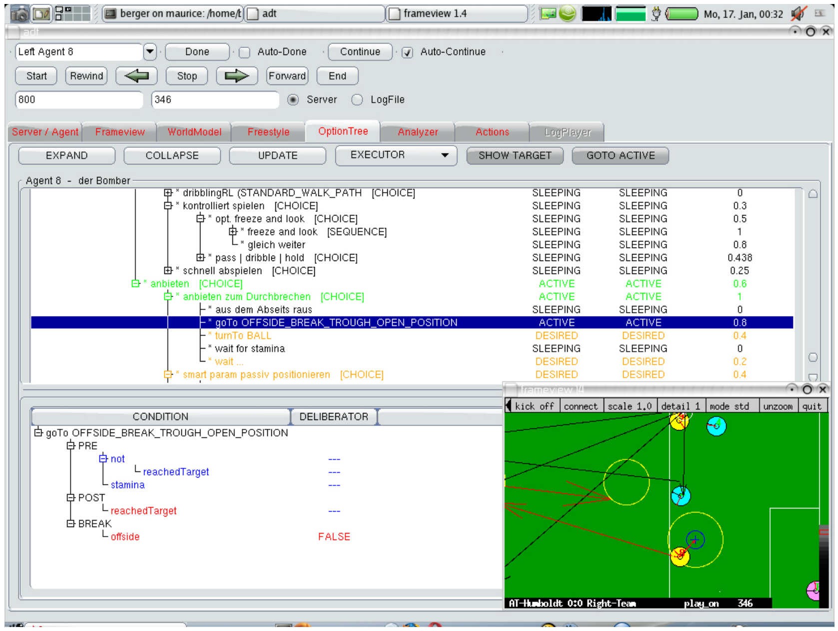

Fig. 4. development tool (ADT) showing a live fragment of behavior tree and the internal state of the double pass architecture

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.

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.

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. Padova: 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.

Ril03. P. Riley and G. Riley. *SPADES — A Distributed Agent Simulation Environment with Software-in-the-Loop Execution*. In S. Chick, P. J. Sánchez, D. Ferrin and D. J. Morrice (eds.), *Winter Simulation Conference Proceedings*, vol. 1, pp. 817–825: 2003.

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]: 2005. Available from: http://www.robocup.de/AT-Humboldt/index.shtml.