

# AT Humboldt Team Description 2008

Ralf Berger, Hans-Dieter Burkhard,  
Alexander Burger and Daniel Herfert

Department of Informatics, AI Group  
Humboldt University Berlin, Unter den Linden 6, 10099 Berlin (Germany)  
[sinteam@informatik.hu-berlin.de](mailto:sinteam@informatik.hu-berlin.de),  
<http://www.robocup.de/AT-Humboldt>

**Abstract.** *AT Humboldt* is the RoboCup Soccer Simulation team of Humboldt University Berlin. Since 1997 we are successfully using our soccer agents as a research testbed for different fields of artificial intelligence and robotics – especially long-term deliberation and realtime reasoning, cooperation and coordination in multi agent systems, case based reasoning aided decision making and machine learning. In this report we will briefly present some interesting aspects of the *AT Humboldt* agent implementation. Further we will describe our main areas of research as well as some new ideas for this years RoboCup competition to be held in Suzhu, China.

## 1 Introduction

The Soccer Simulation team *AT Humboldt*[ATH] was founded in 1997, as part of the Artificial Intelligence Lab at the Humboldt-University Berlin and participated at each of the 11 previous RoboCup world cups. Our Lab is also home of our sister projects *Aibo Team Humboldt*, *Humanoid Team Humboldt* and the new *Nao Team Humboldt*. We look back on a long and successful history in RoboCup competitions. e.g World Champion 1997, Vice World Champion 1998 and 2004 in the Simulation League, World Champion in the Four-legged League in 2004 and 2005 as part of the GermanTeam.

Our group's general research focus encompasses agent-oriented techniques, behavior modeling, intelligent robotics, cognitive science, case-based reasoning, machine learning and neuro-dynamics to name but a few.

The main objective of our simulation group is the development of an universal behavior architecture based on mental models that is applicable to a variety of platforms. As a second focus we investigate the usability of case-based reasoning and reinforcement learning techniques to complex multi-agent problems.

In this report we will give a brief overview of the most important concepts and some interesting work that has been done till now as well as an outlook to new ideas for this years RoboCup competition.

## 2 Research Topics

### 2.1 Behavior Control Architectures for (Multi)Agent Systems

It is widely accepted that a combination of both, reactive and deliberative approaches is essential for achieving complex behavior in dynamic multi-agent systems. Most of these hybrid architectures rely on a layered structure of several behavior modules. They have in common that high-level layers gain control in larger time intervals than lower layers. This principle and the mostly stack-oriented runtime organization leads to several problems in highly dynamic environments. We give a detailed analysis of the problems and limitations (namely context-problem, “failing upwards” [Mur00], least commitment, “Wesson-Oil” [Gat91]) of existing approaches in [Ber06a] and [Bur01].

As a conclusion we have developed a new architecture for plan-based control of autonomous agents (*Double Pass Architecture (DPA)*) that overcomes these problems. The DPA is based on the principles of Bratmans Belief-Desire-Intention Theory (BDI) [Bra87]. Thereby the DPA is more faithful in realization of the fundamentals of this theory than many other BDI-like architectures.

An introducing description of the architecture is given in [Bur05,Bur02,Bur01]. A detailed specification and a comprehensive discussion of the *Double Pass Architecture* can be found in a diploma thesis [Ber06a].

An overview of the architecture components and its interactions is shown in figure 1.

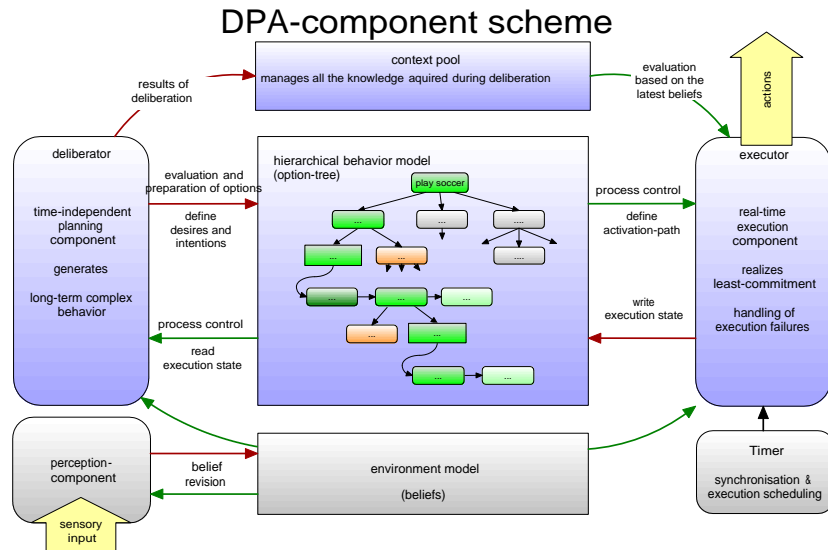


Fig. 1: basic interactions between the components of the architecture

## 2.2 Case-Based Reasoning (CBR) for Decision Support

The use of CBR techniques in this field[Bur07] was pioneered by our team already in 1997[Wen98] and amongst others continued with a case-based decision support module for the goal keeper[Ber]. We have continuously extended our work and have now developed a CBR-framework[L06] that perfectly fits into the Double Pass Architecture and that is able to select and initiate complex game plays for several players by using experience from previous situations. We have already integrated this CBR-system in our team with a first kind of game play, namely a wall pass– more game-plays will follow. The system was able to outperform the former hand-coded behavior selection for this particular tactics right from the start. The most recent recent work is presented in [Ber07b].

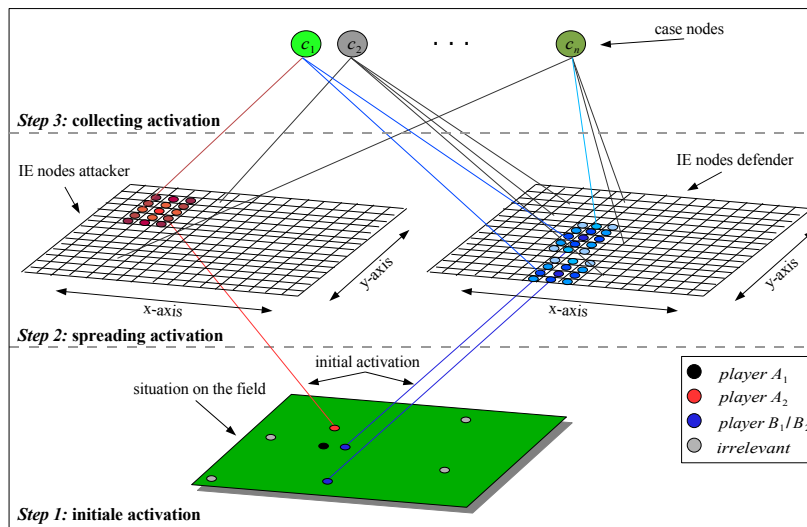


Fig. 2: Exemplary view of the Case Retrieval Net for CBR-based wall pass.

## 2.3 Reinforcement Learning in Complex Domains

In 2004 we successfully started using methods of reinforcement learning (RL) for improving low-level and mid-level skills of our agents. Furthermore we have developed a universal and comprehensive RL-library [RL++] as part of a diploma thesis [Gol05]. As one result the agent's dribble-skill is now about twice as fast and even more failsave than our former handcoded one. A particularly interesting aspect of the actual learning method is the use of evolutionary selection of suitable meta-actions and tile-coding settings for modeling the problem.

### 3 Team Details

In the following sections we want to briefly present some details of the team design that might be found interesting by other simulation teams.

#### 3.1 Localization

Our method of self-localization differs significantly from common approaches like triangulation or particle filters. Our gradient descent algorithm[Bac01] makes use of the facts, that in Simulation League the vision error model is known and that a set of seen flags corresponds to a defined localization figure with an uniform probability distribution. We could show that self-localization using this gradient descent method is not only outstandingly exact and very fast, but also yields much more consistency than other approaches.

#### 3.2 Sensor Fusion

For the problem of merging seen players into an existing situation of the world-model we borrowed a well-known algorithm from graph theory. We transform the sensor fusion problem into the problem of finding the *optimal matching in weighted bipartite graphs*. The nodes of the first partition represent the existing players while the nodes in the second partition represent the seen players. The distance between both players is used as the edge's weight. The method is based on the Dijkstra algorithm for finding shortest paths within a graph and uses Fibonacci heaps which significantly speed up the computation. We could show that this approach performs much better in a variety of situations compared to a rule-based approach[Ber07a].

#### 3.3 Behavioral Complexity

As we mentioned earlier the DPA allows for very specific decision making and intuitive modelling complex behavior. With our old architecture we were able to model and handle only about 30 different behavior options. Since we implemented the DPA within our simulation team we could easily increase the complexity of the used behavior models from year to year. The option trees to be used this year will consist of about 700 different options and more than 7.000 control elements for every single agent. Nevertheless we are still able to understand the resulting agent behavior and could also ensure its execution in real-time ( $< 1ms$ ).

Compared to classical SPA models the concurrent P-SA organization of the DPA can better handle varying computational resources with regard to the behavioral performance. Because the executor has to perform only a minimum of computational work and the deliberator is an any-time process by nature, shortages of available resources have only minor influence to the behavior[Ber06b].

### 3.4 Online Coach

One of the most important tasks for the online coach is the initial selection and substitution of the heterogenous player types. In the last years we've used a rule-based approach for determining which player-type should be used on which position within our team-formation. We found this method very inflexible and hard to maintain, therefore we developed a completely new architecture for tackling the "heterogenous player selection problem".

Here are the key ideas:

- We define a set of  $N$  useful features  $f_n(pt)$  (currently  $N = 9$ ) that vary significantly for the different playertypes (e.g. time for a 15m run, average stamina consumption, ...).
- We normalize these features based on the values for the standard player ( $f_n^*(pt) = f_n(pt)/f_0(pt)$ ).
- We define a  $10 \times N$  weight matrix ( $WP$ ), that describes the "importance" of every of the  $N$  features for every of the 10 player positions within our team. The values of this matrix are initially determined by experience and domain knowledge and can later be optimized by machine learning techniques.
- The value of a player  $p$  after assignment of playertype  $pt$  is:  

$$vp(p, pt) = \sum_{n=1}^N WP(p, n) \cdot f_n^*(pt)$$
- We define a second weight vector ( $WF$ ), that describes the "importance" of every of the 10 player positions within our team formation.
- The value of a complete player substitution is:  

$$vf(p_1, pt_1, \dots, p_{10}, pt_{10}) = \sum_{i=1}^{10} WF(i) \cdot vp(p_i, pt_i)$$
- For determining the best substitution (i.e. the substitution that maximizes  $vf$ ) we use a randomized algorithm that generates and evaluates arbitrary substitutions.
- Depending on the machine the coach can evaluate about 10 million different substitutions within the few seconds before kickoff. Even this is only a small fraction of all the 70 billion possible assignments, we reach an average performance of about 90 – 95% of the maximum possible substitution value.

### 3.5 Agent Development Tool

Since 2003 we use a very powerful tool (ADT) for supporting the process of agent development. Originally the tool was intended to visualize worldmodel-data only, in order to make the agents' decisions more transparent. By adding more and more features it has now become an indispensable implement for designing and revising the agent's behavior. Key-features of our tool are:

- Playing real games step by step
- Generate arbitrary game situations on the field
- Analyzing / visualizing all internal (mental) states of all agents
- Tracing the entire behavior control process (all decisions)
- Visualizing arbitrary data on the field

Figure 3 shows a screenshot of the ADT

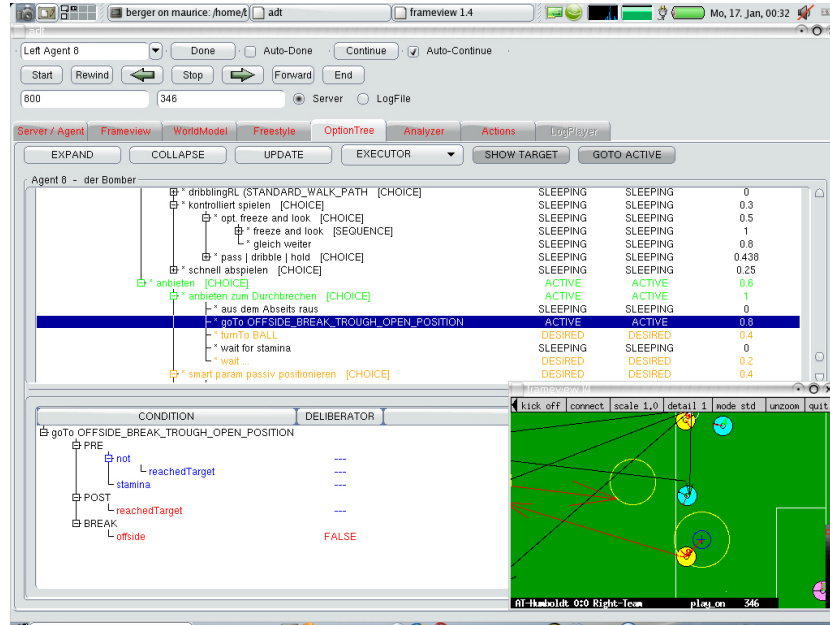


Fig. 3: Agent Development Tool (ADT) showing a live fragment of the behavior tree and the internal state of the Double Pass Architecture

## 4 Changes regarding SoccerServer v12

The SoccerServer (*rcsserver-12.x*) that will be used in this years competitions contains a couple of changes and new features. The overall goal of these changes is to emphasize the strategic aspect of the RoboCup Soccer Simulation League and to create new challenges for the next years. Due to these changes it will be necessary to reconsider and adapt many modules of our soccer agents. We will give a very brief overview on some changes and modifications as well as our ideas about their effects. Most of the needed adaptations have not been finished yet and the next competitions will show if the server changes are a step in the right direction.

### 4.1 Heterogeneous Players

The parameters of the heterogeneous player have been changed to adjust the player's physical strength and to better balance the generated player types. Furthermore the number of different player types has been increased from 7 to 18. This should foster the research on role assignment problems and strategic elements in the online coach.

Our current substitution model in the coach is already prepared for the new challenges since it has a very generic design (see 3.4). The crucial task to do now

is finding appropriate weight parameters that fits to our kind of play and our team formation.

## 4.2 Synchronous Timer

In addition to the asynchronous timer model that has been used for many years now a second timer will be introduced this year. The aim of this new timer is to simplify the synchronization between agent and server. Since both timers are not fully comparable we have to wait what turns out to be the best timer for actual competitions.

Irrespective of that *sense*, *think* and *act* are asynchronous by nature for all real robot applications. Therefore we are convinced that the asynchronous timer should be kept in order to allow research on this interesting and important topic.

## 4.3 Dynamics of the Ball and Goalie Debasement

The maximum velocity of the ball will be increased from 2.7m/step to 3.0 m/step and the catchable area of the goal keeper will be decreased to almost one third of the old value. Even this doesn't sound to be such a big change it will probably have extensive effects on the games, especially for the defense strategy.

There are pros and cons for this modifications - we have to see if the advantages prevail. In order to adapt to these changes we have to (amongst others):

- recalibrate the intercept behavior and all the passing behaviors
- relearn the dribble-skill and the outplay-skill
- recalibrate positioning algorithms and the probability estimation for scoring
- revise the complete goalie behavior

## 5 Conclusion and Outlook

For the last 11 years we successfully used our agent system *AT Humboldt* for interesting and fruitful research and as a great teaching platform resulting in many publications, diploma and PhD thesis'. While much work for 2008 is still in progress (especially due to the last server changes), a detailed description of new results will appear in a final team report.

Some still unmentioned issues we want to tackle for this years RoboCup are:

- dynamic changes of team formation and behavior directives by the coach
- improvement of standard situation behavior by on-line analyzing shortcomings in opponent behavior
- extending the use (and number) of local agent roles for specific tasks
- improving the quality and usability of our codebase
- increased use of automatically generated code

## References

- Bac01. J. Bach and M. Gollin. *Self-Localization Revisited*. In *RoboCup 2001: Robot Soccer World Cup V*, vol. 2377 of *LNAI*, pp. 251–256. Springer: 2001.
- Ber. R. Berger, M. Gollin and H.-D. Burkhard. *AT Humboldt 2003 – Team Description*. In *RoboCup 2003 - Proceedings of the International Symposium*.
- Ber06a. R. Berger. *Die Doppelpass-Architektur – Verhaltenssteuerung autonomer Agenten in hochdynamischen Umgebungen*. diploma thesis, Institut für Informatik, Humboldt-Universität zu Berlin: 2006.
- Ber06b. R. Berger and H.-D. Burkhard. *Modeling Complex Behavior of Autonomous Agents in Dynamic Environments*. In *Concurrency, Specification and Programming CS&P'2006*, vol. 1: *Concurrency of Informatik-Berichte*, p. 116125: 2006.
- Ber07a. R. Berger. *Using MWBM for Allocating Ambiguous Data in Multi Sensor Systems*. In *Concurrency, Specification and Programming CS&P'2007*: 2007.
- Ber07b. R. Berger and G. Lämmel. *Exploiting Past Experience – Using Case-Based Techniques for Decision Support in Soccer Agents*. In *RoboCup 2007 - Proceedings of the International Symposium (submitted)*: 2007.
- Bra87. M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press: 1987.
- Bur01. 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, pp. 71–88. Springer: 2001.
- Bur02. H.-D. Burkhard. *Real Time Control for Autonomous Mobile Robots*. *Fundamenta Informaticae*, vol. 51:pp. 251–270: 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.
- Bur07. H.-D. Burkhard and R. Berger. *Cases in Robotic Soccer*. In *7th International Conference on Case-Based Reasoning (ICCB'07)*: 2007.
- Gat91. E. Gat. *Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots*. *SIGART Bulletin*, vol. 2:pp. 70–74: 1991.
- Gol05. M. Gollin. *Implementation einer Bibliothek für Reinforcement Learning und Anwendung in der RoboCup Simulationsliga*. Diploma Thesis, Institut für Informatik, Humboldt-Universität zu Berlin: 2005.
- L06. G. Lämmel. *Fallbasierte Verhaltenssteuerung im RoboCup*. Diploma Thesis, Institut für Informatik, Humboldt Universität zu Berlin: 2006.
- Mur00. R. R. Murphy. *An Introduction to AI Robotics*. The MIT Press, 1st edn.: 2000.
- Wen98. J. Wendler and M. Lenz. *CBR for Dynamic Situation Assessment in an Agent-Oriented Setting*. In *Proc. AAAI-98 Workshop on CaseBased Reasoning Integrations*: 1998.
- ATH. *Website of AT Humboldt Soccer Simulation Team* [online]. Available from: <http://www.robocup.de/AT-Humboldt>.
- RL++. *Website of RL++ – Open Source C++ library for Reinforcement Learning* [online]. Available from: <http://sourceforge.net/projects/rl-pp/>.