Recognizing and Predicting Agent Behavior with Case Based Reasoning

Jan Wendler[1] and Joscha Bach[2]

[1] Zuse Institute Berlin, Takustraße 7
14195 Berlin, Germany
wendler@zib.de
[2] Institut für Informatik, Humboldt-Universität zu Berlin
Unter den Linden 6, 10099 Berlin, Germany
bach@informatik.hu-berlin.de

Abstract. Case Based Reasoning is a feasible approach for recognizing and predicting behavior of agents within the RoboCup domain. Using the method described here, on average 98.4 percent of all situations within a game of virtual robotic soccer have been successfully classified as part of a behavior pattern. Based on the assumption that similar triggering situations lead to similar behavior patterns, a prediction accuracy of up to 0.54 was possible, compared to 0.17 corresponding to random guessing. Significant differences are visible between different teams, which is dependent on the strategic approaches of these teams.

1 Introduction

Our work is concerned with the analysis of the behavior of agents in a highly dynamic and heterogeneous domain– virtual robotic soccer (RoboCup [4]). Here, two teams of 11 agents each connect to a server that simulates their environment and their 'bodies' in discrete time-steps of 100 ms [11]. The agents have a limited field of view and partially incomplete information about their surroundings. Agents within a team may communicate, but this is limited.

This paper addresses a way of automatically classifying and an attempt at predicting the behavior of a team of agents, based on external observation only. A set of conditions is used to distinguish behaviors and to partition the resulting behavior space. From observed behavior, team specific behavior models are then generated using Case Based Reasoning (CBR) [5, 6]. These models, which are derived from a number of virtual soccer games, are used to predict the behavior of a team during a new game.

This work is based on a diploma thesis by Uwe Müller [9] and a doctoral thesis by one of the authors [15].

2 Approach

There have been previous attempts at recognizing behavior within RoboCup, especially for automatically commenting on soccer games (see, for instance, André

D. Polani et al. (Eds.): RoboCup 2003, LNAI 3020, pp. 729–738, 2004.

et al. [1], Voelz et al. [14], Matsubara et al. [7], Frank et al. [2] and Raines et al. [12]). Wünstel et al. (see, for instance, [18]) use self organizing feature maps to highlight typical behavior elements of teams of agents on a global scale. In Miene and Visser [8] a universal approach for specifying and recognizing behavior is introduced based on spatial and temporal relations.

Relatively little has been published in the RoboCup domain concerning the prediction of individual moves of soccer agents, however, Riley and Veloso [13] use behavior models to predict movements of opponents in standard soccer situations by maintaining probability distributions of the positions of individual agents.

In [17], one of the authors has used a CBR based behavior model to imitate the successful behavior of agents (specifically, the passing behavior).

2.1 Terms

A behavior model is a structure $\mathfrak{M} = [\mathfrak{C}; \mathfrak{B}; f]$, where $\mathfrak{C}$ is a nonempty set of causes, $\mathfrak{B}$ is a nonempty set of behaviors, and f is a function $f : \mathfrak{C} \rightarrow \mathfrak{B}$. Behavior models may be based on actions or situations. Actions are not directly observable in our approach and can not always be determined because they can not always be distinguished by their outcomes. Therefore, we base our approach on situations and assume that similar situations correspond to similar behavior.

Behaviors may be modeled implicitly or explicitly; for instance, the agents of our team AT Humboldt 98 [3, 10] have used an implicit model based on their own interception algorithm to estimate the intercept behavior of opponents and teammates.

A situation S_t^i is what the agent i knows of a world situation W_t. This situation is generated by combining several observations. An observation O_t^i is a part of a world situation W_t, interpreted by an agent i, and may be erroneous and incomplete. A world situation consists of the features that determine the state of the agents' environment at a given time-step t, such as the positions of agents and the ball, their speed vectors, the game-state, the score and the stamina values of the agents.

An (observed) behavior is a process that extends over several adjacent situations and may be described by events. A behavior is caused by the actions of multiple agents and what follows from these actions. The part of the data that is considered to lead to a behavior is what we call a *trigger*. As trigger we apply an extract (relevant data from an observers point of view) of the situation at the start time of the according behavior.

Similar behaviors are grouped into behavior patterns, and likewise, sets of triggers that lead to the same behavior pattern are grouped into trigger patterns.

To describe behaviors, we resort to a sequence of values that we call a behavior-essence, with $\texttt{behaviors}_E$ being the set of all behavior essences. Thus, we can describe the behavior model as $\mathfrak{M} = [\texttt{situations}; \texttt{behaviors}_E; f]$, with $f : \texttt{situations} \rightarrow \texttt{behaviors}_E$.

2.2 Using CBR

f is implemented by a CBR system, which consists of a case base, a similarity measure defined between the cases, a method for updating the case base, and a method for predicting behavior.

To collect cases, behaviors needs to be specified and recognized (section 3). The behavior essences and its triggers are specified (sections 4.1, 4.2) and cases which consist of a trigger and a behavior essence are generated (section 4.3).

Because triggers and behavior essences are determined by a sequence of values, combined similarity measures are used (section 4.4). Each combined measure consists of local similarities between two individual values and a weighted sum of these local similarities. For the behavior similarities, weights and values are specified by the designer. For triggers, only local similarities are given by the designer, and weights are determined automatically.

Based on the case base and the similarity measure the methods to predict behaviors and to update the case base are introduced (section 4.5).

3 Behavior Recognition

The basic recognition process is based on behavior patterns. Behaviors are constructed from events, and they can be recognized by performing a pattern matching of these events against the observations. The general behavior recognition algorithm can be found in [15].

Whenever behaviors contain each other, the longest recognizable behavior is used (for instance, within a dribbling, further instances of dribbling may occur. Currently, we only examine behaviors involving the ball (i.e. positioning behavior such as the building of a pass chain are not considered).

3.1 Specification of Behaviors

Behaviors are specified by defining and arranging the events they are constructed of. We have specified behaviors for passing, dribbling, goal-kicking, clearing and other. We will give a short overview how the pass behavior has been specified. The exact specification can be found in [15] and [16].

For the pass behavior it is necessary that one player controls the ball exclusively during a small time interval. After that the ball mustn't be controlled by any player for some time. Finally the ball needs to be exclusively controlled by another player. Further conditions are used to distinguish a pass from a mere ball transfer, like that the ball is departing from the kicker with enough speed and other.

By adding further conditions regarding ball speed and player movement further differentiation between direct passes, indirect passes and passes with approaching movement is possible. Diagonal passes, back passes and transversal passes can be identified by taking the angle of the ball movement into account.

A behavior fails if an opponent gets into possession of the ball, or if the game-state changes in such a way that an opponent will receive ball control, for instance, when the ball has been shot out of the field. We have added conditions to classify both cases for the four behavior patterns pass, dribbling, goal-kick and clearing.

3.2 Abstraction Levels for the Description of Behaviors

A behavior instance is (from the point of view of an observer) completely determined by a sequence of world states. We discern four levels of abstraction:

- General description: on this level we use names like pass, dribbling or other.
- Manifestation description: this introduces sub-categories like direct pass etc.
- Source-target description: the source of all ball transfers is given by the kicking agent. The target may be a receiving agent, the goal, a position on the field etc.
- Detailed description: this consists of all relevant features of the behavior, that is, of the values of all important attributes. For passing, these are the kicking and the receiving players, the start and end time of the pass, the ball speed and the movement of the receiver.

Of these levels, the source-target description and the detailed description are used for the behavior prediction, while the general and manifestation description maybe important to describe and predict higher level strategies.

4 Prediction of Behaviors

4.1 The Trigger Patterns

The prediction of behaviors is based on the recognition of associated triggers, which are assumed to cause the agents to start the corresponding behavior. Unfortunately, the comparison between triggers that account for all relevant attributes is computationally expensive, let alone the recognition of similarities. To make the comparison of triggers feasible in real-time, only the most important attributes are considered. The primary attributes for a *pass* are *the position of the initiating player* p_1, *the vector from the initiating to the receiving player* $\overrightarrow{p_1p_2}$, *the angle* $\measuredangle(\overrightarrow{p_1p_3};\overrightarrow{p_1p_2})$ *to the first opponent* p_3 *to the right*, *the distance* $|\overrightarrow{p_1p_3}|$ *to* p_3, *the angle* $\measuredangle(\overrightarrow{p_1p_4};\overrightarrow{p_1p_2})$ *to the first opponent* p_4 *to the left* and *the distance* $|\overrightarrow{p_1p_4}|$ *to* p_4.

For dribbling, clearing and goal-kicking, similar attributes have been defined. For additional tests, player number, game state, name of opponent team and time-step are included as *secondary attributes.*

4.2 Behavior Essences

Just like triggers, behavior essences consist of a sequence of primary and secondary attributes. Only the primary attributes are used for the similarity calculations. For passing, the primary attributes are *the ball speed*, *the direction* *of ball movement*, *the movement of the receiver*, *the duration* and *success or failure.*

The reason for failure is examined as secondary attribute.

For other behaviors, similar behavior essences have been specified.

4.3 Generation of Cases

A case assigns a behavior essence to a trigger:

$$case = (T; BE) : T \in \texttt{triggers}; BE \in \texttt{behaviors}_E$$

Let us examine the generating process of a case. As mentioned in section 2.1, the agents each perceive a partial extract of a world-situation W_t as an observation O_t and react with actions A_t that contribute to the world-situation W_{t+1}. This results in a sequence of world situations within the environment. In every time-step, the modeling agent observes the multi-agent system and receives a sequence of observations, which is subsequently evaluated until a complete behavior B_k^{k+l} is recognized. For this behavior, the situation at its starting point is examined to determine the respective T_k. Eventually, the pair of recognized trigger and behavior essence BE_k^{k+l} is added to the case base.

4.4 Similarity Measures

To determine the similarity between triggers of the same trigger pattern and between behavior essences of the same behavior essence pattern two similarity measures are required:

$$\text{similarity}_\text{T} : \texttt{triggers} \times \texttt{triggers} \rightarrow \mathbb{R}_{[0,1]}$$

$$\text{similarity}_\text{B} : \texttt{behaviors}_E \times \texttt{behaviors}_E \rightarrow \mathbb{R}_{[0,1]}$$

These are defined using a weighted sum of the local similarities that correspond to the individual primary attributes of the triggers and behaviors. The weights are specific to each pattern, that is, they are the same for all triggers or behaviors of the same pattern, respectively. Thus, for each pattern a sequence of weights is required.

The local similarities are functions that return values $\in \mathbb{R}_{[0;1]}$; as an example, we give the local similarity function for the positions v_k, v_l of initiating players for the pass trigger pattern:

$$\text{sim}_{\mathtt{T},1}^{\text{pass}}(v_k, v_l) = \begin{cases} 1 & \text{if } |(v_l - v_k)| < 2 \\ 0 & \text{if } |(v_l - v_k)| > 15 \\ 1 - \frac{|(v_l - v_k)| - 2}{15 - 2} & \text{else} \end{cases}$$

Similar functions exist for the other primary attributes of the pass trigger pattern.

While the local similarities and the weights for behavior patterns are defined by the designer, the weights for trigger patterns are determined automatically using the following method. The goal of this method is, given a set of models $\mathbb{M}$ and a finite set of data $\mathbb{D} \subset \mathbb{X} \times \mathbb{Y}$, to find the model $M^* \in \mathbb{M}$ that best describes the functional relationship between the input values $X \in \mathbb{X}$ to the output values $Y \in \mathbb{Y}$. $\mathbb{D}$ is separated into a set of base data and a set of test data. Each model is then evaluated in combination with the base data for every test item. The best model M is determined by calculating the prediction error of all models and choosing the one with the minimal error.

The course of a game is taken as test data $\mathbb{TD} \subset \texttt{situations} \times \texttt{behaviors}_E$ and the best weights model W^* is then determined using

$$W^* = \operatorname*{argmax}_{W_i \in \texttt{weights}_\text{T}} \frac{1}{|\mathbb{TD}|} \sum_{(S_j, BE_j) \in \mathbb{TD}} \text{similarity}_\text{B}\left(BE_j, \text{predict}(W_i, caseBase, S_j)\right)$$

where predict is the behavior prediction function that returns an estimate for the behavior $BE \in \texttt{behaviors}_E$, based on a sequence of weights for the trigger $W_i \in \texttt{weights}_\text{T}$, a case base $caseBase \subset \texttt{cases}$ and a situation $S_j \in \texttt{situations}$.

$$\text{predict} : \texttt{weights}_\text{T} \times 2^{\texttt{cases}} \times \texttt{situations} \rightarrow \texttt{behaviors}_E$$

Thus, this method determines the sequence of weights that maximizes the similarity between predicted and recognized behavior for a set of test data $\mathbb{TD}$. The sequence of weights depends on the basis data and test data and therefore on the modeled team.

The complexity of this calculation depends linearly on the cardinality of the test set $\mathbb{TD}$, on the cardinality of the set of weight sequences $\texttt{weights}_\text{T}$ and the complexity of predict. By assuming that identical attributes have identical weights, we may reduce the space of $\texttt{weights}_\text{T}$ from $\mathbb{R}_+^{20}$ (because there are 20 primary attributes) to $\mathbb{R}_+^5$ (there are only 5 different primary attributes). If the weights for each attribute are limited to natural numbers $\leq n$, only $(n+1)^5$ sequences of weights have to be tested. Still, this is too computationally expensive for the given predict function.

Instead of varying the weights between 0 and n, a distribution of n weight units on k primary attributes is considered, i.e. n weight points are completely distributed to the k attributes, $\texttt{weights}_\text{T} = \{(w_1, \ldots, w_k) : w_i \in \mathbb{N} \wedge \sum_{i=1}^k w_i = n\}$. Thus, the set of weight sequences has $\binom{n+(k-1)}{k-1}$ elements, and for 10 weights and 5 primary attributes, there are 1001 weight sequences.

4.5 Selecting Cases from the Case Base

The function f of the behavior model $\mathfrak{M} = [\texttt{situations}; \texttt{behaviors}_E; f]$ is primarily determined by the function predict:

$$f(S) = \begin{cases} \text{predict}(W, caseBase, S) & \text{if } S \text{ is a decision situation} \\ \perp & \text{else} \end{cases}$$

that is, f gives results only in decision situations - this is the case when an agent of the modeled team controls the ball. Furthermore, decision situations have to be at least 4 time-steps apart (this is the minimal dribble duration).

In a decision situation, first potential triggers are identified. For passing this are up to 10 triggers, for goal-kick up to one trigger, and for dribbling and clearing exactly 6 triggers.

For every identified trigger, the case base is searched for cases with similar triggers. The similarity leads to an assessment of the cases. The best cases are selected.

From these cases, the best case is chosen using preferences, which are partly derived from the secondary attributes. Cases are preferred, if e.g.

– they reflect the same game state as the trigger
– they have the same opponent team as the trigger
– they have the same initiating agent as the trigger

The degree of preference depends on a bonus value for each criterion. The bonus values are determined automatically and in the same manner as the trigger weights.

The case that maximizes the sum of similarity and bonus value is chosen. Finally, the case is adapted according to the situation at hand: by comparing the observed trigger and the trigger stored in the case, the speed and angle of the ball movement, as well as the behavior duration are adjusted.

The function predict returns the adapted behavior essence of the case that maximizes the sum of similarity and bonus value.

The case base is extended by the current case (i.e. the case derived from the trigger and the actually observed behavior) if the similarity between the predicted behavior and the observed behavior is smaller than a fixed value σ_{limit} (i.e. the behavior essences differ significantly). Deletion or modification of cases does not take place. The similarity measure is updated by recalculating the sequence of weights for all trigger patterns.

5 Evaluation

For testing our approach, we made use of the data set derived from the GermanOpen RoboCup competition of 2001, which consisted of 44 games by 12 teams.

5.1 Behavior Recognition

We have found that using our approach on the full set of 44 games, we managed to classify between 96.4 and 99.7 percent of all ball controlling behaviors. Of these, passing amounted on average to 43.4 percent, dribbling to 24.8 percent, clearing to 20.2 percent and goal kick to 2 percent. Ball combat was observed in 8.2 percent of the time, one-two passes in 1.1 percent. Only in 1.4 percent of the time, non-classifiable ball transfers were observed, with 0.1 percent of completely unrecognized behavior.

The results of the algorithm have been verified against an independent, detailed manual classification of the same data.

5.2 Prediction of Behavior

Prediction of behaviors can be done at different abstraction levels (see section 3.4). We use here the “detailed description”-level and concentrate our experiments on pass behaviors. For every pass the predicted pass instance is compared with the actual performed pass instance. For the comparison the similarity measure for behaviors are used.

Using the described approach, a number of experiments for the prediction of ball-handling behaviors were conducted, especially to determine the relationship between prediction accuracy and the number of cases and the number of weight units. In detail we will only present the experiment regarding the dependency of the behavior model on the modeled team. For more experimental results, especially on the prediction of passing partners, see [15].

When using different numbers of weight units, it becomes clear that the distribution of weights on the individual attributes differs with the modeled team, i.e. teams differ in the importance they apply to the individual attributes.

During the experiments we discovered that a small number of weight units is sufficient to find a good team specific weight sequence. With more than 10 weight units, no further substantial improvements in prediction accuracy could be made using our approach.

The accuracy of prediction can be improved by increasing the number of cases. When using between 900 and 1000 cases (8 games), a plateau in prediction accuracy was reached, and the inclusion of more cases did not result in visible improvements any more. The maximum average prediction accuracy for the team “Brainstormers01” amounted to 0.54 (using 928 cases), however, for a game of the “Brainstormers01” against a different team, the accuracy was 0.46, using the same case base.

Dependency of Behavior Model on Modeled Team. To establish that the models obtained using cross validation are indeed team specific, we have used the “Brainstormers01” model to predict the behavior of other teams in the tournament. We have found that the behavior models are indeed specific for the modeled team.

The experiment was done for all teams who reached the final round. The case bases were generated based on the first five games per team. The weight sequences are determined for a randomly chosen game (among the 5). The prediction is evaluated for all final games of the team a) using the case base of the modeled team, and b) using the case base of the team “Brainstormers01”.

The prediction using the team specific model is in all cases more accurate than the one with the ‘alien’ “Brainstormers01” model. The relatively high similarity between the prediction results of “Brainstormers01” to “DrWeb”, “MRB” suggest that these teams use strategies for behavior selection that bear similarity to that of “Brainstormers01”.

“Aras”, “FCPortugal” and “RoboLog2k1”, on the other hand, seem to use strategies that are very different from those of “Brainstormers01”.

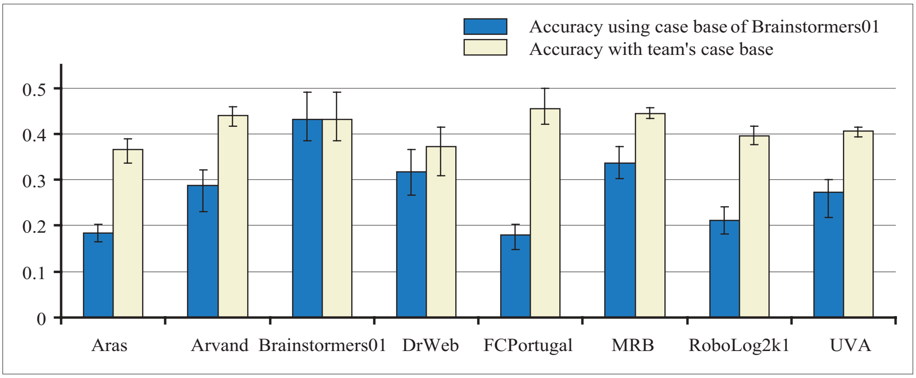

Fig. 1. Prediction accuracy for eight teams using a team specific model and the model of the "Brainstormers01" team

5.3 Conclusion and Outlook

We have found that, using our approach, ball controlling behaviors can be successfully classified, but predicted only with an accuracy between 0.39 and 0.54. However, this compares favorably to random guessing, which returns an accuracy of only 0.17. To obtain these results, a case base with about 1000 entries and a similarity measure using 10 weight units are sufficient.

It can be shown that the prediction model is specific for each team, whereby the use of the model of another team points to the use of similar strategies.

The limited accuracy of the results is due to two aspects: First, there are limitations that stem from the assumptions of the approach, which is ignorant of internal states of the agents, such as an incomplete world model, previous communication between agents or the execution of long-term plans. On the other hand, the description of the cases has been severely restricted to few parameters, which was necessary to limit the complexity of the modeling process.

Our approach only attempts to determine the behavior with the maximum probability. A more general description can be achieved by learning the probability distribution of the behaviors that are triggered by each trigger pattern.

Still, we consider the approach as being successful. We expect that some improvements over our current results can be made with alternative case descriptions. Substantially better results will probably be obtained with the simulation of internal states of the agents, thus performing a transition from a reactive to a context dependent model.

Acknowledgments

This research has been supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316). Most of this work was done at the Humboldt-Universität zu Berlin.

We thank Prof. H.-D. Burkhard and Kay Schröter for their contributions, fruitful discussions and support.

References

1. E. André, G. Herzog, and T. Rist. Generating Multimedia Presentations for RoboCup SoccerGames. In *RoboCup-97:Robot Soccer World Cup I*, 1997.
2. I. Frank, K. Tanaka-Ishii, K. Arai, and H. Matsubara. The Statistics Proxy Server. In T. Balch, P. Stone, and G. Kraetzschmar, editors, *Proceedings of the fourth RoboCup Workshop*, pages 199–204, Melbourne, Australia, 2000.
3. P. Gugenberger, J. Wendler, K. Schröter, and H.-D. Burkhard. AT Humboldt in RoboCup-98. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, volume 1604 of *LNAI*, pages 358–363. Springer-Verlag, 1999.
4. H. Kitano, Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara, and E. Osawa. RoboCup: A Challenge for AI. *Artificial Intelligence Magazine*, 18(1):73–85, 1997.
5. J. Kolodner. *Case-Based Reasoning.* Morgan Kaufmann, 1993.
6. M. Lenz, B. Bartsch-Spörl, H. D. Burkhard, and S. Wess. *Case Based Reasoning Technology. From Foundations to Applications.* LNAI. Springer, 1998.
7. H. Matsubara, I. Frank, K. Tanaka-Ishii, I. Noda, H. Nakashima, and K. Hasida. Automatic Soccer Commentary and RoboCup. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, LNAI. Springer, 1999.
8. A. Miene and U. Visser. Interpretation of spatio-temporal relations in real-time and dynamic environments. In *Proceedings of the 5th International Symposium RoboCup*, Seattle, WA, USA, 2001. The RoboCup Federation.
9. U. T. Müller. Beschreiben und Erkennen von Verhaltensmustern beim simulierten Fußballspiel. Diploma thesis, Humboldt Universität zu Berlin, Germany, 2002.
10. P. Müller-Gugenberger and J. Wendler. AT Humboldt 98 – Design, Implementierung und Evaluierung eines Multiagentensystems für den RoboCup-98 mittels einer BDI-Architektur. Diploma thesis, Humboldt Universität zu Berlin, 1998.
11. I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer Server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12:233–250, 1998.
12. T. Raines, M. Tambe, and S. Marsella. Automated Assistants to Aid Humans in Understanding Team Behaviors. In M. Veloso, E. Pagello, and H. Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, LNAI. Springer, 2000.
13. P. Riley and M. Veloso. Planning for Distributed Execution Through Use of Probabilistic Opponent Models. In *IJCAI-2001 Workshop PRO-2: Planning under Uncertainty and Incomplete Information*, 2001.
14. D. Voelz, E. Andre, G. Herzog, and T. Rist. Rocco: A RoboCup Soccer Commentator System. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, LNAI, pages 50–60. Springer, 1999.
15. J. Wendler. *Automatisches Modellieren von Agenten-Verhalten – Erkennen, Verstehen und Vorhersagen von Verhalten in komplexen Multi-Agenten-Systemen.* PhD thesis, Humboldt Universität zu Berlin, Germany, Feb. 2003.
16. J. Wendler and J. Bach. Case based behavior assessment in simulated robotic soccer. In G. L. H. D. Burkhard, T. Uthmann, editor, *Modellierung und Simulation menschlichen Verhaltens*, volume 163 of *Informatik-Berichte*, pages 134–147. Humboldt-Universität zu Berlin, 2003.
17. J. Wendler, G. A. Kaminka, and M. Veloso. Automatically Improving Team Cooperation by Applying Coordination Models. In B. Bell and E. Santos, editors, *2001 AAAI Fall Symposium Series, Symposium: Intent Inference for Collaborative Tasks*, pages 84–90, Menlo Park CA, 2001. AAAI Press / The MIT Press.
18. M. Wünstel, D. Polani, T. Uthmann, and J. Perl. Behavior Classification with Self-Organizing Maps. In P. Stone, T. Balch, and G. Kraetzschmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, LNAI, Germany, 2001. Springer.