Cases in Robotic Soccer

Hans-Dieter Burkhard and Ralf Berger

Humboldt University Berlin
Institute of Informatics
D-10099 Berlin
{hdb,berger}@informatik.hu-berlin.de

Abstract. Soccer playing robots are a well established test bed for the development of artificial intelligence for use in real environments. The challenges include perception, decision making and acting in a dynamic environment with only unreliable and partial information. Behaviors and skills for such environments must be optimized by experiences. Case Based Reasoning provides an excellent framework for learning as discussed in this paper.

1 Introduction

Early AI was based on symbolic descriptions of problems using logics, theorem provers and search techniques for solutions. There was a common understanding that chess programs could be a milestone to understand and implement intelligent behavior. Now we have machines that can play chess, but these machines are not considered to be really intelligent. We have learned that acting in the real world is much more difficult for machines. Machines are still far away from performing daily tasks. Therefore, the development of soccer playing robots has become a new challenge. The competitions in RoboCup are used to evaluate scientific and technological progress, similarly to the role of chess in the past.

The key problem of AI is the knowledge about daily life, how it is like to ride bicycle or to climb a tree, or simply to walk. Such skills are necessary to understand language, to interpret a scene given by visual sensors, or to decide what to do next. Human beings do acquire this knowledge by learning, by experiencing the environment, by collecting cases about good and bad behavior. Therefore, Case Based Reasoning (CBR) can be used as a basic technology together with other methods from Machine Learning. At the same time, CBR meets again its roots in cognitive science. It is still a challenge to understand how the experience can be stored and organized for later use. The scenario of soccer playing robots provides a lot of different tasks in dynamic real environments. The tasks include perception, skills and deliberation.

Because lack of space, we cannot give a detailed introduction to RoboCup. There are recently five different leagues, introduced to tackle different problems on the base of the available hard- and software. Real robots are investigated in the

– **Middle Size League (MSL)** with robots not exceeding a 50 cm diameter.

R.O. Weber and M.M. Richter (Eds.): ICCBR 2007, LNAI 4626, pp. 1–15, 2007.

– **Small Size League (SSL)** with robots not exceeding 15 cm in diameter.
– **4-Legged League (4LL)** with Sony's AIBO robots.
– **Humanoid League (HL)** with robots of human shape.

The **Simulation League (SL)** was established in order to explore more complex strategic and tactical behaviors which cannot be realized with real robots up to now. Besides individual programs for the 11 players, each team has a coach program for directing the playing style (while analyzing an ongoing match).

More information about RoboCup can be found on the website [1]. Recent developments are discussed in the article [2].

The paper is organized as follows: In Section 2 we start with a very short overview on the programming of soccer robots. It is the basis for the discussion of the Machine Learning tasks in Robotic Soccer in section 3. A discussion of the CBR related general problems is given in section 4, and section 5 gives short overviews about existing work using CBR in RoboCup.

2 Programming Soccer Robots

The robots in RoboCup have to act autonomously, no human interaction is allowed. In the so-called sense-think-act cycle they have to recognize the environment, to decide about their next goals and related actions, and to perform the actions using their skills.

The robots have to gather all needed information using their sensors. They have to process the sensory input to obtain a picture about the situation, the localization of the robot itself, of the other robots, and of the ball. Today, visual sensors are widely used to perceive the environment. Sophisticated algorithms for picture processing and scene interpretation are needed. Statistical methods like Kalman filters or particle filters are used for localization tasks. Not only the place but also the the direction and the speed of the ball are very important. Latency modeling (a good team in SSL has a latency of approx. 110ms) and prediction methods are important as well.

Especially the biped (humanoid) and quadruped robots (AIBO) need various proprioceptive sensors for observing and controlling their movements. Sensors for joint angles, forces, and torques measure the positions, directions and movements of different parts of the body.

Having a belief (not necessarily a true knowledge) about the environment, the robot has to decide for its next goals and actions. This means to check and to evaluate the own chances in comparison to the opportunities of other robots (team mates and opponents) on the playing ground. Therefore the robot needs knowledge about his own skills and about the results it can hopefully achieve.

There are different levels of control. On the lowest level, the robot has to control its body movements. In the case of humanoid robots it has to keep balance while walking or kicking. This needs a continuous interaction between sensor inputs and appropriate actions at the related joints. The compensation of an unexpected force by an adjustment of the heap is an example. It is still an open problem in the worldwide research on humanoid robots how this can be achieved best: how to couple sensors and actors, which sensors to use and where to place them, how to program the control etc. Recent efforts try to implement some kind of a spinal cord inspired by solutions from nature. Because of the lack of complete models, methods from Machine Learning are tested for the development of efficient (distributed) sensor-actor loops.

Having such basic skills like dribbling, intercepting or kicking the ball, the next level of control concerns the choice of an appropriate skill for a given goal. While the skill is performed, the robot has continuously to check the performance of the skill, e.g. maintaining control over the ball while dribbling. Again, a close interaction is necessary between sensors, control, and actuators.

On the highest level(s), tactical and strategic decisions can actually take place. Related reasoning procedures are especially studied in the simulation league because it is the only league which already uses 11 players per team.

3 Machine Learning Tasks in Robotic Soccer

As discussed in the previous section, a soccer program consists of modules according to a “horizontal” structure regarding the sense-think-act cycle, and a “vertical” structure of the levels (layers). The related modules can cooperate in different ways depending on the architecture in use. Visual perception, for example, is performed vertically starting with primitive image operations on the lowest level up to the scene interpretation using background knowledge (physical properties, design of the playground etc.) at the higher levels. Horizontal cooperation is necessary for the sense-think-act cycle.

Many of the processes inside the modules as well as the interconnections of the modules are subject to Machine Learning. Available data are incomplete and unreliable such that programming by hand leads to sub-optimal solutions. Moreover, optimal solutions are often too costly because of real-time requirements. Hand crafted systems in RoboCup were sufficient only during the first years. Now, all the good teams in simulation as well as in the real robot leagues use various Machine Learning techniques to a great extend. RoboCup has become an important scenario for development and evaluation of Machine Learning. The scenario of keep away soccer [3] has become a standard evaluation test bed for Machine Learning.

It is not possible to train all aspects of successful soccer playing in a single learning process. The overall learning task (how to win) has to be decomposed into smaller tasks. Up to now, the most scenarios investigated for Machine Learning in RoboCup are rather granular, but because of the interdependencies of the processes, the scenarios for learning are depending on each other. Actually, the pioneering work for multi layered learning came from the RoboCup community [4]. We will give some examples in section 5.

3.1 Perception

The players need to have beliefs about the movement of the ball, about their own position and the position of other players. In the early days of RoboCup teams used distance data provided by range finders. But driven by the recent cheap camera prices, visual data are most important today. Useful constraints between relative and absolute data (distances, angles, positions, speed) can be exploited. Absolute data are measured with respect to global coordinates, relative data are measured with respect to the player itself (egocentric world model). The data are collected and analyzed over time, usually with an underlying Markov assumption. Statistical methods try to overcome the unreliability of the measurements. Particle filters and Kalman filters are used for positioning tasks today. The tuning of their parameters is a special learning task.

Up to now, the environment of the robots is carefully designed with color coded objects. In the near future, the robots are to play in arbitrary environments, e.g. in a gym. The only spatial background knowledge the robots can rely on is the fact that there should be two goals and maybe some field lines on the ground. Therefore the robots will have to learn orientation also from other landmarks available in a concrete room.

An intensively studied field is opponent modeling, especially in the simulation league. The coach agent can observe the match and try to find out useful information about the other team. There are different player types, and the coach can try to find out which opponent players are on which positions. Moreover he can try to identify special patterns (cases!) about the style of playing. The findings can be used to improve the own strategy. The coach can analyze the behavior recorded from log files as well as online during a match.

3.2 Act

Reliable basic skills are essential for the success in robotic soccer as well as in human soccer. The simulation league provides an ideal test bed for the investigation of different skill learning techniques. The league can provide as many data as wanted with low cost. In the real robot leagues, experiments are expensive regarding the costs for the equipment, and they are time consuming. This leads to more sophisticated experimental designs. Accompanying simulations are used to get a better understanding, and special methods help for off line pre-selection of promising trials [5].

With the arrival of legged robots, especially the humanoid ones, internal sensors for measurement of forces and joint angles are used for the stabilization and for fast movements including omni-directional walking, running, kicking and dribbling. With about 20 degrees of freedom and frame rates of more than 100 fps, learning methods are mandatory to tune appropriate sensor-actor-loops.

Learning basic skills, like approximation of the best interception point for a moving ball was already an early learning task in RoboCup simulation league [6]. It is one of the characteristic properties of RoboCup, that skill learning does not concern only a single action. In most cases, the success depends on the learning of a suitable sequence of actions. This is obvious for the chain of motor commands for legged robots, but it was even necessary in the simulation league from the very beginning. A prominent example was the success of AT Humboldt over the favorite team CMUnited in 1997. A successful shoot consists of a sequence of well tuned kick actions, and CMUnited was not prepared for such kicks. But it is only a nice tale, that AT-Humboldt team could kick "the ball around themselves, continually increasing its velocity so that it ended up moving towards the goal faster than was imagined possible. Since the soccer server did not enforce a maximum ball speed, a property that was changed immediately after the competition, the ball could move arbitrarily fast, making it impossible to stop" [4]. Since faster velocities would make the ball to leave the kickable range immediately, it was not possible to get a higher speed than the later defined maximum speed.

3.3 Decision Making

The control tasks can range from basic reactive behaviors up to high level deliberative behaviors. The overall performance depends on appropriate interconnections. Using weak skills, learning of decision criteria will result in optimization of doing what is possible. Replacing the skill by a better one will need a new learning of the higher level decisions.

There are obvious tasks for Machine Learning approaches like classifiers for the selection of appropriate skills for a situation. Likewise, simple tasks concern the choice of skills for the player in possession of the ball. The player may choose between scoring, passing, dribbling etc. Next he may choose between different ways to perform the chosen action, e.g. by a kick selection procedure and the determination of parameters like direction and speed of the kick.

More complex deliberation concerns the behavior of the players not controlling the ball. Typical tasks are supporting or marking, more complex behavior concerns standard situations. Deliberation of this kind needs more understanding of tactical options. Certain patterns can be identified. The fine tuning or even the detection of useful patterns is very challenging. Besides well developed basic skills, the performance in high level cooperative play is already mandatory for teams in the simulation league, and it becomes more and more important in the real robot leagues. An exquisite example are the Tribots MSL team from University of Osnabrueck (world champion 2006) with successful transformations of methods from their simulation team Brainstormers (vice champion in 2006).

There are different implementations for cooperative team play. Explicit planning and data structures for resulting plans do not seem to be mandatory. Neural networks have been trained using reinforcement learning to determine just the next action useful in the recent situation. Of course, Reinforcement Learning did consider the later progress of playing, but the neural net computes only the immediate action [7]. A related concept with symbolic representations was used by AT Humboldt in 1997 [8]. The idea behind such concepts is the following presumption: If there exists a good potential plan, then the subsequent choices will lead to actions consistent with the potential plan. Problems may arise from oscillations between different potential plans. Therefore the teams with such approaches take some additional care for stability.

Other approaches use explicit symbolic plans. Symbolic approaches permit the description of behavior patterns and standard situations of soccer, like change of wings, wall passes, offside trap, free kicks, corner kicks, etc. The suitable behavior in such situations can be described in a script-like manner, where concrete parameters are filled in as appropriate. Such a behavior is started with only a rough partial plan (the "idea" how to perform the behavior). In the beginning of a wall pass, the both players involved know about the sequence of dribbling (player 1)/positioning(player 2), pass from 1 to 2, intercept (2)/run over opponent (1), pass back from 2 to 1, intercept by player 1. This is only a rough script not a complete plan. The concrete parameters are determined during the progress of the behavior depending on the opponents behavior, the movement of the ball etc. (least commitment). The higher levels of layered architectures are commonly used for the long term commitments. The choice of appropriate plans may be considered again as a classification task. Tuning for optimization is useful to find good parameters.

The coordination of different players can rely on different approaches. Communication is useful to some extend, but limited in bandwidth and subject to losses (especially for wireless communication). Cooperation without communication is also possible since all players act in the same environment. Therefore RoboCup provides a lot of interesting challenges for multi-agent learning.

3.4 Machine Learning Methods in RoboCup

As we have seen, RoboCup needs learning for classification and for optimization. Neural networks and Case Based Reasoning are often used for classification purposes. Evolutionary approaches provide good results for scenarios with large parameter spaces, e.g. complex situations or locomotion of legged robots. Learning of skills with delayed rewards are treated with methods from Reinforcement Learning, where various function approximations for the large parameter spaces are in use.

There is no space to discuss all methods in detail. Instead it can be stated that rather all Machine Learning methods can be applied – and have been applied in different soccer scenarios and for different modules. There are several hundred teams in RoboCup competitions year by year, and there are a lot of people working in Machine Learning.

4 CBR in RoboCup – Some General Remarks

4.1 What Are Cases, and Where Do They Come from?

As in many other CBR systems, the classical distinction between problem part and solution part (rule type cases) is useful for soccer applications, too. For simple classification tasks, the problem part contains examples from the classes, while the solution gives the correct class. The selection of actions or skills can be considered as classification. The solution may also contain a quality measure which evaluates the suitability of the proposed solution. Negative numbers indicate that the solution was not successful or not correct. The contents (vocabulary) of the cases are often given by attribute-value pairs of positions, speeds, teams, score, time, intentions etc. Similarity is then calculated by the local-global principle [9]: Local similarities of the attributes (e.g. inverse distances of positions) are combined by a certain function. Weighted sums are very popular, whereby the weights can be adjusted by learning.

The data of the matches provide a large pool for case bases. For skill learning, the cases can be recorded from special experimental settings (e.g. for intercepts or dribbling). The simulation league can produce data as much as one needs. For the real robots, the collection of data is more limited by time and by the efforts needed for recording.

The case data are then extracted from the recorded files. This means to identify related situations and related sequences of actions from a stream of recorded events. A pass e.g. consists of a kick by the first player, then the ball moves freely for some time, and then the second player intercepts the ball. Thereby, the second player must belong to the same team. It is difficult to judge if a pass was actually performed by intentions of both players. But often it is only important that a pass occurred, regardless for what reasons.

There are useful methods for the analysis originally developed for commentator programs and for coaches, respectively. Such programs can find the co-occurrences of pre-situations, action sequences, and post-situations. They can find the successful passes in the recorded games. But there are also situations in the matches where a pass could have been successfully performed, but the player did not try. Such situations would need a more careful analysis. This addresses an old problem of experience based learning: If there was no trial, then there is no experience (exploration problem). A human expert could consider such situations and design related cases. More sophisticated analysis tools can be used for such tasks at least to some extend.

Moreover, typical cases can be designed completely by humans. There are a lot of standard situations in soccer. They are often explained by related cases in human soccer. This provides a good alternative approach for programming soccer robots. Instead of defining the conditions for the application of a maneuver in terms of spatial relations between the players and the ball, one can provide a set of typical cases and use CBR methods.

The use of cases is often appropriate since the decision are ordered by time: A recent situation (problem) is mapped to subsequent actions (solution).

One drawback of the rule type case format is the need for different cases while dealing with the same standard situation, e.g. during a wall pass. There we have a unique script with ongoing decisions (when to pass to whom, where to run to, where to intercept). Using rule type cases, we would need different cases for each of these decisions.

This problem can be solved using the ideas of case completion as proposed in [10]. A case describes a whole episode (e.g. of a wall pass) with concrete decisions and actions. While performing a new episode, those cases are retrieved step by step from the case base which correspond to the recent belief (initial part of the recent course of events and situations, e.g. after performing the pass from player 1 to player 2 in the wall pass). These cases can then provide more data from their stored experience: How the problem was solved with further decisions (e.g. where the player 2 should intercept the ball, where the player 1 should run to etc.).

Such cases are of constraint type: Unknown values are determined from given data using cases as constraints. Systems using constraint type cases can be implemented with Case Retrieval Nets (CRN, [10]). Such a case is a set of information entities which have occurred together in the stored episode. Their usage consists of retrieving remaining (unspecified) information entities (IEs) to a partially described situation.

Another example of constraint type cases comes from perception. There, a case may contain the positions of objects (e.g. players and ball) in world coordinates, and the distances between the objects. The description is redundant (e.g. distances could be calculated from positions and vice versa if some fixed positioned objects are involved). Therefore a case with all these data can be retrieved by some of its IEs and serve for specifying the unknown IEs (they could also be calculated by trigonometry – but humans do not).

4.2 Maintenance

Maintenance of cases is substantial for the success of CBR in soccer programs. Large case bases are not useful because of the needs for real-time processing. It is necessary to keep a bounded number of cases using related techniques.

Moreover, cases may become invalid over time. This may happen due to changes in other modules, for example after collecting new cases there. It may also occur for cases collected online, e.g. for opponent models. If a team uses such models and changes its style of play, then the opponent usually responds with other patterns of behavior (besides the fact that opponents may also change their behavior using their related modeling methods for our team).

Another problem concerns the consistency of the case bases of different players of a team. If the players use the same CBR system, they have a chance to obtain the same proposals. It then depends on comparable world models. But if the players have different case bases (due to online collected experiences), then their decisions may not lead to a joint intention.

4.3 Generalization (Adaptation)

In the soccer domain, adaptation is usually closely related to the similarity measure. This concerns spatial transformations (positions, symmetries etc.) and seems to be continuous for a first look. Actually, there can be substantial discontinuities according to quantization effects. They are explicitly implemented even in the soccer server of the simulation league.

5 CBR in RoboCup – An Applicational View

In the last 10 years CBR has been applied to a broad variety of aspects of robot control. We know of more than 20 international publications from the RoboCup community in this field. In the following sections we will have a more detailed look on how Case Based Reasoning is incorporated.

5.1 CBR Methods for Self-localization

Sensing a camera image with some striking features similar to an image corresponding to a known position (case) usually implies that the current position is similar to the known one.

The paper [11] utilizes local visual information of landmarks for self localization (position and orientation) of an AIBO robot in the 4-Legged League. The problem part of a case represents an omni-directional view from a certain position. The solution is the according position on the field.

The playground was divided into cells of $20cm \times 20cm$. For each of these partitions a case was generated, that consists of information about all landmarks. In detail these are the following features: the width, height and color of the appearances of all landmarks as well as the angles between pairs of landmarks. Thus every case consists of 68 (out of 859 possibly different) information entities.

In the application, only some landmark features are available (since the robot camera has a view angle of 50 degrees). Hence cases must be retrieved according to partial problem descriptions. This was implemented with the help of Case Retrieval Nets [10]. To find the robot's position a weighted sum is computed over all solutions of cases that are sufficiently similar to the given camera image. The main advantage of this approach is its flexibility and its robustness even against some strongly incorrect visual information.

5.2 CBR Methods for Opponent Modeling

The overall problem of opponent modeling is defined as "building up a model of the opponent's behavior based on observations within a game". The particular practicability of Case Based Reasoning is given since human players seem to solve this problem in a similar way and there are usually only few learning samples to exploit. At least three different research groups applied CBR methods to opponent modeling in RoboCup.

Wendler et al. [12,13] use a combined system for recognizing and predicting of behaviors. The prediction is based on the recognition of associated triggers, which are assumed to cause the agents to start the corresponding behavior. For each particular behavior they define a set of relevant attributes such as positions or relative angles. Cases are generated automatically during the behavior recognition learning phase. Potential triggers have to be identified for the retrieval, and then the case base is searched for cases with similar triggers. Finally, the case is adapted according to the current situation by comparing the observed trigger and the trigger stored in the case.

The work of Steffens [14,15] investigates improvements of prediction accuracy for case-based opponent modeling. The approach also enhances efficiency since it exploits the same observations during learning for different purposes. While the observations remain the same, the similarity measure is adapted to the type or role of the agents. This adaptation is done by integrating problem solving knowledge represented in goal dependency networks (GDN). The GDNs are defined manually and contain very general but domain dependent information. It could be shown that using an adaptive similarity measure regarding the role of the agents leads to a better prediction of a player's actions.

The approach of Ahmadi et al. [16] is similar to the last one. It also tries to optimize the actual CBR process by adapting its meta-parameters. It uses a second case-based subsystem for this task, building a two-layered CBR architecture. The complexity of the problem is split into two subproblems, each of which works with a relatively small number of cases. The features of the ordinary ('lower') cases are defined relative to the ball. These cases provide local solutions which can be applied everywhere in the field. Adapting these cases for different game situations requires information about the current focus of the play. This information is provided by the 'upper' cases and incorporates the position and velocity vector of the ball and a rough estimation of the position of all the players. The second layer provides optimal case parameters (representation, retrieval and adaptation). It monitors the performance of the lower layer. Ahmadi et al. could show that the the system is able to learn a competent opponent model by the iterative application of this approach to only very few games.

5.3 CBR Methods for Situation Analysis and Decision Making

In this paragraph we will address the problems of situation analysis and (individual) decision making. These are probably the tasks where the application of case-based methods is most apparently.

Probably the first application of Case Based Reasoning in RoboCup is shown by the 'AT Humboldt' team [17]. CBR was used for dynamic situation assessment in the Simulation League. The task is to find a 'preference position' where the player should move to. Cases are represented by a feature vector of a game snapshot including the following properties: occupancy of the segmented playground by other players, time until a teammate will control the ball, preference directions, available power resources, distance to the ball and to the other players. The work again uses Case Retrieval Nets.

In [18] the prior work was continued towards decision making of the goalkeeper. He has to decide whether to stay in front of the goal and defend the goal-line or to run towards an attacker to decrease the possible shooting angles to the goal. The logfiles of previous games were analyzed for situations in which a goal attack was running and the goalie had to decide what to do. From each of such situations a case was generated which basically contained the positions of the players and the ball, the ball's velocity, the decision of the goalie regarding the discussed scenario, and the success of this behavior. The problem of finding a suitable similarity measure was tackled by using a combination of an inverse distance and a relevance function that provides a rating of the estimated impact of a players position on the goalie's decision. An interesting aspect of this work is that the whole procedure from processing and analyzing hundreds of logfiles to building up the index and the runtime structure works fully automatically and takes just a few hours.

The work of 'AT Humboldt' was recently extended to a comprehensive CBR-framework [19,20] for decision making for cooperative tasks. Perhaps the most interesting feature is its twofold case-base optimization process.

Firstly only the significant pieces of information from each case are extracted. This is done by defining areas of interest based on the spatial relations between the ball and the relevant players. The deletion of the non-essential information speeds up the retrieval and leads to more general cases. The second optimization task is to delete the redundant cases. To determine whether a case is redundant (it can be deleted without decreasing the competence of the case-base), the individual competence contribution model based on the concepts of coverage and reachability is used. It turned out that the deletion of the redundant cases shrinks the case-base significantly. Furthermore the information density of the case-base decreases and the dispersion of the information becomes more homogeneous which again speeds up the retrieval. First applications of the system used the game play 'wall pass' to successfully show it's performance.

A very comprehensive work comes from the group of Raquel Ros [21,22]. They propose an almost complete methodology for case-based decision making applied to the 4-legged League. Their work covers:

Case-acquisition: The idea of this work is to start with an initial case base of prototypical cases that was manually designed with the help of expert knowledge. A supervised training is installed afterwards where an expert reviews the retrieved solution of the system. The robot can then adopt the scope of the case accordingly.

Case format: As usual a case represents a snapshot of the environment at a given time. The case definition is composed of three parts: $case = (P, K, A)$. P is the problem description containing a set of spatial attributes as well as some game-based attributes (timing of the match and current goal difference). K indicates the scope of the case defined as the regions of the field within which the ball and the opponents should be located in order to retrieve that case. A is the solution description – a sequence of actions the robots should perform. This is often denoted as 'game play'.

Retrieval: The retrieval is implemented as a twofold process: It considers the similarity between the problem and the case, and the cost of adapting the problem to the case. The similarity function indicates how similar non-controllable features (cannot directly be influenced) are between the problem and the case using local similarities and a global aggregation function. The cost function defines the cost of modifying the controllable features (own and teammates' positions) of the problem to match the case.

Reuse: The reuse phase refers to the adaptation of case features before executing the associated actions. Its basic idea is to transform the controllable features of the current problem in a way that the relation between these features w.r.t. the ball is the same as in the retrieved case.

5.4 CBR Methods for Planning

There are a lot of different possibilities for the integration of Case Based Reasoning into the robot's planning process (from multi-agent decision making to complete architectural models). Since the spectrum is too broad we will only pick some exemplary work and outline its ideas briefly.

A complete single-player hybrid architecture (CBRFuze) is introduced in [23]. It combines a deliberative problem solver using Case Based Reasoning and a reactive part using fuzzy behavioral control. The problem description part of the cases uses a set of fuzzy linguistic variables which is also helpful for case indexing and provides an easy similarity measure.

Marling et al. [24] show how Case Based Reasoning can be integrated into various tasks of their Small Size team. They present three CBR prototypes, for the tasks of positioning the goalie, selecting team formations, and recognizing game states. So far the prototypes are only realized in simulation yet.

In [25] a system for strategic behavior selection in the Small-Size League is proposed. It utilizes Case Based Reasoning for dynamic situation assessment in offensive and defense game situations. In addition Bayesian classifiers are used to choose between optimal behaviors. The approach was tested using the formerly mentioned 'keepaway' task.

Karol et al. [26] propose a theoretical model for high level planning in the Four-Legged League. Their model supports game play selection in common and key game situations. It is argued that developing a case base for robot soccer game plays can capture creative genius and enduring principles of how to play the game. The proposed approach uses the conceptual spaces framework for categorization of cases by well-defined similarity measures.

5.5 CBR Methods for Coaching

Until now coaching is available exclusively in the Simulation League. A coach may give advises to adapt the team's game strategy. Furthermore, the coach can initially choose between varying player types that differ from each other in their physical attributes. He can assign them their roles in the game, and he can substitute players up to three times during a match.

The problem of finding a good line-up is investigated using a case-base approach in [27]. The problem part of a case consists of the individual properties of all available heterogenous players. The solution part presents some alternative solutions. Each solution features descriptive elements like formation type, main strategy or opponent team as well as the assignment of the player types. It also provides some measure of the quality of the solution. This was done by analyzing the performance of games played with the related formation. The second important issue was the definition of appropriate similarity measures between heterogeneous player types, i.e. the question which of the properties have a significant impact on the similarity between two player types.

5.6 CBR Methods for Acting

There is some published work of using Case Based Reasoning for acting in general robotics (e.g. for navigation and parameter optimization) but we know only one paper in RoboCup. In [28] an interesting combination of Reinforcement Learning and CBR is presented. Case-based methods are used for approximating a high-dimensional, continuous state value function of the Reinforcement Learning task. A case is regarded as pair of state representation and state value estimation learned from exploration examples. To determine a specific state value, k-nearest neighbor regression is used based on Euclidean distances. Special maintenance procedures are implemented. They handle the growth of the case base and serve for deleting older cases. Since early cases may be due to early insufficient approximations, such cases should be removed from the case base when approximation becomes better. The approach was evaluated using the ball interception task and could produce good behavior policies within a very short time and with comparatively little case data.

6 Conclusion

Intelligent behavior in restricted domains – as today already implemented in numerous assistance systems or in chess – can be achieved using special methods and techniques (e.g. search, statistics, artificial neural networks). But complex intelligent behavior needs the solution of lots of different combined problems using a large variety of methods and technical staff. Many skills which humans seem to perform easily are of that kind. Perception and action, language understanding and communication are examples.

Soccer playing robots provide a very challenging test bed with a lot of different requirements similar to the requirements of intelligent behavior in real world scenarios. It is impossible to program such robots in all its details. Instead, methods from Machine Learning are needed for the development and the tuning of suitable features, skills and behaviors. Since acting in the real world is based on experiences, Case Based Reasoning is best suited for the tasks on hand.

We have shown, that CBR can be used for all aspects of the sense-think-act-cycle, and we have discussed the existing work in this field. There are in fact a lot of interesting results and useful applications. Nevertheless, there are more open than solved problems to date. Especially the integration of different solutions is a challenging task for CBR-methods.

The development of autonomous intelligent robots is a challenge which can only be achieved by the integration of different fields. The soccer playing robots are an attempt to study these problems and to use the framework of friendly competitions for scientific research. Thus it does not really matter if robots can win against human players in 2050. Nevertheless it is important to have this vision in mind as a long term goal to consider new questions and to foster new results.

Acknowledgments

The authors wish to thank all the thousands of researchers within the RoboCup Federation. All of the exciting work in RoboCup is possible only with the help of this world wide community.

References

1. The RoboCup Federation: Official Website URL: `http://www.robocup.org`
2. Visser, U., Burkhard, H.D.: RoboCup – 10 years of achievements and future challenges. AI Magazine (to appear)
3. Stone, P., Kuhlmann, G., Taylor, M.E., Liu, Y.: Keepaway Soccer: From Machine Learning Testbed to Benchmark. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 93–105. Springer, Heidelberg (2006)
4. Stone, P.: Layered Learning in Multiagent Systems. PhD thesis, Carnegie Mellon University, Computer Science Department (1998)
5. Quinlan, M.: Machine Learning on AIBO Robots. PhD thesis, The University of Newcastle, Faculty of Engineering and the Built Environment (2006)
6. Stone, P., Veloso, M.: A Layered Approach to Learning Client Behaviors in the RoboCup Soccer Server. Applied Artificial Intelligence 12, 165–188 (1998)
7. Riedmiller, M., Gabel, T.: On Experiences in a Complex and Competitive Gaming Domain: Reinforcement Learning Meets RoboCup. In: Proceedings of the 3rd IEEE Symposium on Computational Intelligence and Games (CIG 2007)
8. Burkhard, H.-D., Hannebauer, M., Wendler, J.: AT Humboldt – Development, Practice and Theory. In: Kitano, H. (ed.) RoboCup-97: Robot Soccer World Cup I. LNCS, vol. 1395, pp. 357–372. Springer, Heidelberg (1998)
9. Burkhard, H.-D., Richter, M.M.: On the Notion of Similarity in Case Based Reasoning and Fuzzy Theory. In: Pal, S.K., Dillon, T.S., Yeung, D.S. (eds.) Soft Computing in Case Based Reasoning, pp. 29–46. Springer, Heidelberg (2000) CISM Courses and Lectures No. 408, pp. 141–152. Springer, Wien, New York (2000)
10. Burkhard, H.D.: Extending some Concepts of CBR – Foundations of Case Retrieval Nets. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.) Case-Based Reasoning Technology. LNCS (LNAI), vol. 1400, pp. 17–50. Springer, Heidelberg (1998)
11. Wendler, J., Brüggert, S., Burkhard, H.-D., Myritz, H.: Fault-tolerant self localization by Case-Based Reasoning. In: Stone, P., Balch, T., Kraetzschmar, G.K. (eds.) RoboCup 2000. LNCS (LNAI), vol. 2019, pp. 259–268. Springer, Heidelberg (2001)
12. Wendler, J.: Automatisches Modellieren von Agenten-Verhalten - Erkennen, Verstehen und Vorhersagen von Verhalten in komplexen Multi-Agenten-Systemen. Dissertation, Humboldt Universität zu Berlin (2003)
13. Wendler, J., Bach, J.: Recognizing and Predicting Agent Behavior with Case Based Reasoning. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 729–738. Springer, Heidelberg (2004)
14. Steffens, T.: Enhancing Similarity Measures with Imperfect Rule-Based Background Knowledge. Dissertation, University of Osnabrueck (2006)
15. Steffens, T.: Adapting Similarity Measures to Agent Types in Opponent Modelling. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) Adaptive Agents and Multi-Agent Systems II. LNCS (LNAI), vol. 3394, Springer, Heidelberg (2005)

16. Ahmadi, M., Lamjiri, A., Nevisi, M., Habibi, J., Badie, K.: Using two-layered case-based reasoning for prediction in soccer coach. In: Proc. of the Intern. Conf. of Machine Learning; Models, Technologies and Applications (MLMTA'04) (2004)
17. Wendler, J., Gugenberger, P., Lenz, M.: CBR for Dynamic Situation Assessment in an Agent-Oriented Setting. In: Proc. AAAI-98 Workshop on CaseBased Reasoning Integrations (1998)
18. Berger, R., Gollin, M., Burkhard, H.-D.: AT Humboldt 2003 – Team Description. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, Springer, Heidelberg (2004)
19. Lämmel, G.: Fallbasierte Verhaltenssteuerung im RoboCup – Ein Verfahren zum Fallbasierten Erkennen von Doppelpass-Situationen und der Steuerung eines entsprechenden Verhaltens in der RoboCup Simulationsliga. Diploma Thesis, Institut für Informatik, Humboldt Universität zu Berlin (2006)
20. Berger, R., Lämmel, G.: Exploiting Past Experience – Using Case-Based Techniques for Decision Support in Soccer Agents. Technical report (2007)
21. Ros, R., Veloso, M., de Antaras, R.L., Sierra, C., Arcos, J.L.: Retrieving and Reusing Game Plays for Robot Soccer. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 47–61. Springer, Heidelberg (2006)
22. Ros, R., Arcos, J.L.: Acquiring a Robust Case Base for the Robot Soccer Domain. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 1029–1034. AAAI Press, California (2007)
23. Lin, Y., Liu, A., Chen, K.: A Hybrid Architecture of Case-Based Reasoning and Fuzzy Behavioral Control Applied to Robot Soccer. In: International Computer Symposium ICS'02, Workshop on Artificial Intelligence (2002)
24. Marling, C., Tomko, M., Gillen, M., Alexander, D., Chelberg, D.: Case-Based Reasoning for Planning and World Modeling in the RoboCup Small Sized League. In: IJCAI Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments (2003)
25. Srinivasan, T., Aarthi, K., Meenakshi, S.A., Kausalya, M.: CBRRoboSoc: An Efficient Planning Strategy for Robotic Soccer Using Case Based Reasoning. In: International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'06) (2006)
26. Karol, A., Nebel, B., Stanton, C., Williams, M.-A.: Case Based Game Play in the RoboCup Four-Legged League – Part I The Theoretical Model. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, Springer, Heidelberg (2004)
27. Gabel, T.: A CBR Approach to the Heterogeneous Players Problem in Robotic Soccer Simulation. Student theses, University of Kaiserslautern (2002)
28. Gabel, T., Riedmiller, M.: CBR for State Value Function Approximation in Reinforcement Learning. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 206–221. Springer, Heidelberg (2005)