An Overview on Opponent Modeling in RoboCup Soccer Simulation 2D

Shokoofeh Pourmehr, Chitra Dadkhah,

Computer and Electrical Engineering Department of K.N. Toosi University of Technology
shpourmehr@ee.kntu.ac.ir, dadkhah@eetd.kntu.ac.ir

Abstract. This paper reviews the proposed opponent modeling algorithms within the soccer simulation domain. RoboCup soccer simulation 2D is a rich multi agent environment where opponent modeling plays a crucial role. In multi agent systems with adversarial and cooperative agents, team agents should be adapted to the current environment and opponent in order to propose appropriate and effective counteractions. Predicting the opponent's future behaviors during competition allows for more informed decisions. We divide opponent modeling into two categories of individual agent behaviors and team behaviors. Individual behaviors concern modeling the low-level behaviors of individual opponent agents, however in team behaviors, the high-level strategy of the entire team like formation, offensive and defensive system, is recognized. Several methods have been proposed to create different models of opponents to improve the performance of teams in an essential aspect. In this paper, we review the approaches to the problem of opponent modeling published from 2000 to 2010.

1 Introduction

The idea of knowledge extraction from other agents' behavior was originally used in the field of the game theory [1]. In order to win a competitive game against an unknown adversary, it is vital to adapt to the dynamics of the environment, mainly caused by opponent's game play. An opponent is an agent that has private strategies and has goals that are conflicting with your own [2]. Opponent modeling predicts and identifies the future behaviors of opponent and proposes an appropriate counteraction [3].

One of the most interesting environments where agent modeling has been used is the robotic soccer domain [4]. The Robot World Cup Initiative (RoboCup) is an ambitious initiative whose ultimate goal is to create, by 2050, a robotic soccer team capable of beating the human soccer world champion. As such RoboCup represents a multidisciplinary area where one of the main domains encompasses the variety of

different areas of computational intelligence. Three of the major aspects necessary for generating a competitive team are advanced learning, teamwork and opponent modeling concepts [5].

The task of adapting to the environment and opponent is enormously difficult due to the dynamic nature of soccer matches along with the multiple interactions between players. This is where opponent modeling can be used to recognize or predict the behavior of the agents or team such as formation, defense and offense system and pass graphs which allows for more informed decisions. There is a growing body of research on the use of opponent modeling as a common challenge for agents in both RoboCup and general multi-agent systems.

RoboCup soccer simulation provides a good platform for modeling a soccer team in a dynamic and multi-agent domain. The soccer simulation 2D league is based on the publicly available soccer server [45] system enabling 24 client programs (22 players and 2 coaches) to connect through UDP/IP socket [6]. The server simulates the players and the field in a 2D soccer match (shown in Figure 1). The server accepts low-level commands from the players, executes them in an imperfect way and sends (imperfect) perception information to the players [6]. That means, it creates a virtual soccer field and provides all players with local, incomplete and noisy perception information. In RoboCup Soccer Simulation 2D, it is often difficult for players to make a correct decision because of the uncertainty in the field information [7].
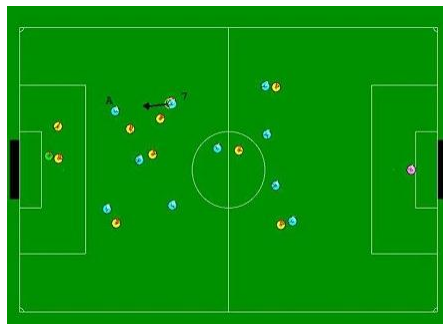
Fig. 1. Soccer simulation 2D's environment as appears in the Soccer Monitor

Each team is allowed to employ a further agent, the coach, which gets a noise-free, global view over the field. It intended to observe the game and to provide additional advice and information to the players. A frequently used opponent modeling approach in RoboCup soccer simulation is to rely on coach agent because it is provided with the complete and noiseless information from the field and can aid in creating agents with adjustable autonomy [8]. In order to focus entirely on opponent modeling, the RoboCup Simulation Coach Competition was held from 2001 to 2006. This competition was situated in the same soccer server, but instead of creating a full soccer team, a single coach agent had to be implemented. RoboCup Coach Competition changed in 2005 in order to emphasize opponent modeling approaches. This challenge calls for research on modeling a team of opponents in a dynamic, multi-agent domain [5]. In this Competition, the teams were directly evaluated based

on how its coach agents identify the weaknesses and strengths (patterns) of the opponent from other opponent behaviors without these patterns [9].

Taking a general overview, it can be seen that the task of solving the problem of opponent modeling has been translated in different ways, namely learning of opponent formation, extracting opponent patterns, recognizing opponent plays and forecast the opponent behavior. Thus, various methods have been proposed to create these models. We divide the task of opponent modeling into two classes of individual behaviors and team behaviors. Individual behavior is about modeling the skills of an individual soccer agent including goalie's positioning, intercepting and path planning. Team behavior deals with the high-level strategy of the entire team like formation, offensive and defensive systems.

The goal of this paper is to take a general overview of the opponent modeling concept in soccer simulation 2D. The paper structure is as follows: Section 2 describes the most prominent approaches which are proposed to model team behavior. Section 3 presents the researches on individual behavior of agents. Section 4 concludes the paper.

2   Team Behavior

The classification and prediction of strategies and team formations play an important role in opponent modeling and can represent relevant information to implement counter strategy or tactic to reduce the performance of the opposite team. By analyzing play history, it is possible to gather critical insights into future plays [10]. Prediction of an opponent's behavior requires that their current behaviors can be recognized and classified and recalled later [2].

The techniques of Data Mining for classification problems can be used to solve the problem of behavior detection within the simulated robotic soccer environment. All the matches in simulated robotic soccer are saved in log files. Thus, it is possible to create a repository of data containing the historical records of certain teams. Using specialized software, such as Soccer Monitor [11], the log files can be used to provide a visual perception of the behavior of virtual players in the field. When necessary it can lead to analyze the behavior of the team, forecasting the actions or players positions, using information from previous matches [12]. Team behaviors in soccer domain can involve formation of agents or other strategies and tactics which are called game plays. In this paper, we classify the approaches to team behavior opponent modeling into two parts; formation and game play.

2.1   Game Play

Extensive works have been done on adapting teams' behavior to its opponents. One of the most useful approaches is to do adaptation which relies on classification of the current opponent into predefined models. Opponent modeling process in this approach is comprised of feature extraction, model construction and classification. First of all, in feature extraction part, useful features from the raw sensor data will be

abstracted. After that, from the information in the features, opponent models will be constructed which embody the strategic information of how this type of opponent's models behaves. Then, in a matching task, observed opponent will be classified to the predefined models [13].

Riley and Veloso [14] used a windowing approach to extract useful features and a decision tree for classification. They have tried to model high-level adversarial behavior by classifying opponent actions as belonging to one of a set of predefined behavioral classes. Their system could classify fixed duration windows of behavior using a set of sequence-invariant action features. An observation occurred over a fixed length of time (i.e. a window) which affects the accuracy of the classifier and its performance. The classification accuracy for 30-37 classes was around 40% which in order of magnitude is better than guessing (<3%).

Based on this classification algorithm, Riley et al. contributed adaptive setplays which changed and improved throughout a game in response to the opponent team's behavior [15,16]. Their approach named ATAC, standing for Adaptive Team-Adversarial Coaching, used coach agent which was equipped with a repository of predefined hand-written opponent models of behavior. Using a naïve Bayes style algorithm, the coach agent was able to select between different models online. When the game is stopped, e.g., due to an out-of-bound call, the coach rapidly takes advantage of the short available time to create a team setplay plan that is a function of the matched modeled opponent's behavior. For adaptation to modeled opponent the coach agent uses Simple Temporal Network based plan representation and execution algorithm which expresses temporal coordination and monitoring in a distributed fashion.

The downside of this approach is that during execution, the agents do not take advantage of opportunities which may occur. For example if an agent ends up with a good shot on the goal, but the plan is to pass, then it will pass the ball. As they suggested, storing alternative plans and intelligently adding monitors for these plans as in [17] could make the plan execution opportunistic [15, 16]

Iglesias et al. [4] used similar process for recognizing and classifying an observed team behavior. After abstracting useful features from the previous games' log files, they analyzed these features in order to recognize different events. Based on the fact that the actions performed by a soccer team are sequential, they proposed to store events in a trie data structure and use that to obtain useful information. A trie (which is abbreviated from "retrieval") is a kind of search tree similar to the data structure commonly used for page tables in virtual memory systems. Example of this trie data structure is as follow:

"{Pass1to2(R)→Dribble2(R)→Pass2to10(R)→Pass10to11(R)} and {Dribble2(R)→Pass2to10(R)→Goal10(R)}"

The advantage of this kind of data structure is that every event is stored in the trie just once. Each event has a number that indicates how many times it has appeared. In addition, they used a statistical dependency test [18] for discovering the significance of sequences and subsequences. To evaluate the relation between an event and its previous events sequence Chi-square test is used [19].

They claimed that this approach works successfully when the pattern followed by a team is related to the players' actions. However, in this research, the different field regions in which the action occurs, has not been represented. As a result, if the pattern

followed by the team is related to this aspect, this proposed method would not be viable. Furthermore, if the pattern is related to actions that occur when the player is not in the possession of the ball, this method would not be viable. [4], [20]

They effectively improved their work as represented in [21]. In this work, a soccer agent team behavior is represented as a distribution of its relevant atomic behaviors. In addition, modification of the Chi-Square Test is used in the classification method. In previous method only the expected values are compared and if an observed value is not represented in the expected distribution, it is not considered. In order to solve these problems, the way to compare the two distributions is modified to the sum of the terms which is called (Chi-Square-Obs Test). An important advantage of the proposed test is its speed since only the observed subsequences are evaluated.

Unlike Riley who stored every observation in the opponent models [13], Steffens [22] claimed that a limited number of opponent models can describe a wide range of opponents so his featured-based models contain only a small number (between two to fifteen) of distinct and stable features. Steffens suggested FBDOM method which stands for Feature Based Declarative Opponent Modeling. This method can be used either as an online or offline method for opponent modeling. He demonstrated a feature in RoboCup domain with the following example:

"The opponent often does long pass along the left wing to the forwards."

Then he proposed that rules which map actions to situations are the proper means to express such features. However, a domain-specific language which can formalize situation and action descriptions is necessary. During the match, features in the opponent models which have a probabilistic nature, will be detected by the observations that come in as raw sensor data. In classification part, with a Bayesian classifier, the opponent model with the best value will be chosen. Then a knowledge base will decide which counter-strategy is applicable. Steffens designed several experiments to test if feature-based models are able to represent opponent behaviors, if they generalize to previously unseen teams, and what effect the observation length (i. e. the amount of classification data) has. These experiments showed that the identification accuracy was high for the modeled teams, so the claim that features are a well-suited method to describe opponent behaviors can be supported.

In another work [23], Steffens presented a similarity-based approach to model high-level opponents actions (e.g. shoot towards goal). In its approach, Steffens proposes the use of Case-based reasoning (CBR) in order to predict the opponent's actions from the coach point of view. He increases the classification accuracy by including some derived attributes from imperfect domain theories. Its results showed that similarity-based opponent modeling can benefit from domain knowledge even if it is not known whether the opponent uses the same domain knowledge.

Case-Based Reasoning (CBR) is a powerful and a frequently applied way to solve problems for humans. However, using CBR in highly dynamic environments like soccer simulation 2D results in a large number of cases to be retained, leading to high computational costs for subsequent case management [24].

Ahmadi et al. [24] solved this problem by using an additional layer of cases to an ordinary layer which provides representation, adaptation and similarity measurement parameters for using a case-based architecture where new cases are recognized and stored during the games, but modeling is still done by a coach. In this work, Ahmadi et al. proposed a direct learning strategy, where the agents decision making is directly

adapted based on the predicted behavior of opponent agents. This learning approach is appropriate for soccer simulation domain, because the cost of acquiring examples is high, and a high occurrence of samples for a specific situation is not available (required by most learning approaches like Neural Networks and statistical learning approaches). Due to this, every event should be well exploited. They suggested that fuzzification of the cases may also improve the performance of both of the implemented CBR learning systems. Additionally, it provides understandable high-level rules to comprise the game strategies.

In [25], the focus is on the unsupervised autonomous learning of the sequential behaviors of agents from observations of their behavior. Using a hybrid approach, the observations of a complex and continuous multi-variant world state would be translated into a time-series of recognized atomic behaviors. Then this time-series are analyzed in order to find repeating sub-sequences that characterize each team behavior. This system is able to identify different events corresponding to the basic behaviors of the agents using a set of models specialized in recognizing simple and basic behaviors of the agents (e.g., intercept, pass). It should be noted that this approach is based mainly on the tactic of the team without considering the formational behavior of the team.

In [26], a symbolic approach (similar to [27]) for behavior prediction based on association rule mining has been presented. They focused on qualitative representations of information. In their approach a sequential pattern mining algorithm is applied in order to learn frequent patterns in the data. These patterns are then transformed into prediction rules which can be applied to estimate what is likely to happen in the future. One characteristic of the learning approach is high representational power with the potential of learning complex patterns with predicates and variables from relational and temporal data. The drawback of the approach is the high complexity of the learning algorithm. The experiments support the assumption that without limiting the search space during pattern generation the algorithm cannot be used to learn complex patterns on-line due to time and space complexity. They suggested that it is necessary to develop heuristics which allow an efficient learning of patterns without cutting off a large number of potentially good patterns.

Ramos and Ayanegui [28] proposed a model able to manage the constant changes occurring in the game. The solution proposed by these authors is a model that allows the building of topological structures based on triangular planar graphs, able to manage the constant changes in the match. This approach enables multiple relations between the agents. Based on this model tactical behavior patterns had even been discovered in the dynamic conditions.

In another line of work, the champion of the RoboCup Soccer Coach Simulation Competition at 2004 and 2006 used a rule based expert system for modeling the opponent team [3]. Fathzadeh et al. defined the model of the opponent as a collection of multiple identified patterns. In this approach the identification of the opponent's patterns was done by an autonomous coach agent who analyzed the log files of the opponent's past games in offline mode and advises own players. To classify opponent's behaviors, a 3-tier learning architecture was developed. Firstly, sequential events of the game were identified using environmental data. Then the patterns of the opponent were predicted using statistical calculations. Eventually, by comparing the opponent patterns with the rest of team's behavior, a model of the opponent was

constructed. In online mode, observing the live game, coach exposes an online model of the opponent and compares it with the stored models in repository. One of the major factors in success of this team was its capability in the handling of the noises and conflicts. In addition, they advised their players to motivate the opponent players to demonstrate the patterns. This trick had assisted them in identifying the opponent behaviors in a simply and fast manner [29-31].

In [32] a relational model to characterize adversary teams based on its behavior has been proposed. In particular, this paper focuses on the tasks of sequence classification using a logic representation for the sequences and the extracted features. The difference of this method compared to previous works is the logical representation language used to model the sequences and the proposal of a distance measure between agent behaviors described as logical sequences.

2.2 Formation

Some of the most important team behaviors are related to tactical plays. In general, tactical plays are planned most of the time and they should occur under the context of formations [28]. Team formation has been defined in different ways. In some works [33], [7], [34] formation is defined as set of player positionings according to the ball position. So the most basic formation would be made out of eleven positionings, ball position information, and the correspondent position of all team players. However, the most popular definition which most of the researchers are using is based on positions of soccer-agents and relevant relations between them. For example, a code 5:2:3 represents a formation composed by five defenders, two midfielders, and three forwards. Goalkeepers are not counted because they are always in a single position [28-31], [12], [35], [36].

A series of computational experiments showed that for this complex task the learning ability of the artificial neural network (ANN) is good [7], [28], [34], [36-37].

For recognizing formations Visser et al. [37] proposed a model based on an artificial neural network. In this model a set of default formations supply information about the opponent team to the online coach. The coach observes the game continually and analyzes the formation of the opponent team at given points in time with an artificial neural network and broadcasts an adequate counter formation to the players during the next interruption. The positions of the players serve as inputs for the ANN, which is trained with the formations most commonly played in test games and the log-files. Whenever the play mode switches to another state than PLAY ON, the coach generates a message for his team. A main consideration about this approach is the lack of any evaluation on counter formations' quality.

Another similar work is proposed in [35] that recognize the formation of the opponent team using a neural networks model. This work feeds the observed player positions into a neural network and tries to classify them into a predefined set of formations. If a classification can be done, the appropriate counter-formation is looked up and transmitted to the players. Like Ramos and Ayanegui, Santiago takes into account multiple relations among defender, midfielder and forward players.

Nakashima et al. in [7] used neural networks to learn opponent's team formation. They employed an off-line learning algorithm where neural networks were used to

learn opponent formations from log files that were generated after the matches of target opponent teams.

The idea of using home areas of players to recognize formations comes from the research done by Riley and Veloso in [38]. A home area specifies the region of the field in which the agent should generally be. Thus, they propose that in identifying home areas, the agents can infer a role in the team (defender, midfielder or forward players). A drawback of this approach is that due to dynamic conditions of the world, the player movements can generate such a wide range extending considerably the home areas, which makes the task of determining the role of a player more difficult.

UT Austin Villa, the champion of the RoboCup 2005 coach competition [39], constructed a model of the opponent by characterizing their behavior with a set of features calculated from statistics gathered while observing a game. These features can be characterized as the team formation indicating the general positioning of the agents (e.g. how many players are defenders or attackers) or play-by-play statistics indicating the frequency of game events such as passes, shots, dribbles, etc. Despite its good results, there are many additional potential features that could boost performance within the same framework. In addition, the announcement strategy could be extended to explicitly model the likelihood of correctness relative to the cost of waiting longer to announce the score.

Almeida and colleagues present a Data Mining methodology for the forecast of team formation in [12]. To perform the detection of the formations they used the software Waikato Environment for Knowledge Analysis (WEKA) [40] since it includes a diversity of learning algorithms able to assist in the forecast of the teams' formations and has an easy to use graphical interface. Six learning algorithms were chosen on the basis of popularity and known experimental results: J48, Naive Bayes, K-Nearest Neighbor (IBK), PART, Multi-layer Perceptron and Sequential Minimal Optimization (SMO). Their experiments results show that the learning algorithm that generates the most appropriate model for predicting the formation of a team in simulated robotic soccer matches is the SMO. This study provides a flexible tool for formation forecast which enables the coach of robotic soccer team to be assisted with a decision support system when for instance a given team changes its formation. This tool can be improved by decreasing the percentage of cases incorrectly classified and analyzing the time spent by a team in the transition from one formation to another in order to reduce the incorrectly classified cases.

In a similar work, Faria and colleagues in [36] compared four machine learning techniques in identifying the opponent team and classification of the opponent formations. The conclusions obtained revealed that if a model was trained with certain games, the 3-NN showed better results in predicting formations. However, since the training games are not the same that we wish to predict in a competition, another test set with a different data set of games was applied and the results produced by Support Vector Machines (SVM) were, in terms of accuracy, the best.

3 Individual Behavior

In modeling individual behavior of agents, positioning and interactions between a small numbers of agents are predicted. Stone et al. [41] proposed a technique named

"Ideal-Model-Based Behavior Outcome Prediction" (IMBBOP) that uses opponent optimal actions based on an ideal world model to model the opponent's future actions. This work was applied to improve the agent low level skills. For example, it was used to decide when to shoot and when to pass when an agent has the ball close to the opponent's goal. It is also used by agents to determine when the opponents are likely to be able to steal the ball. Stone's work does not directly construct a model.

In [8] a probabilistic adaptive opponent modeling named D-AdHoc (for Dynamic-AdHoc) is proposed. In this approach each agent observes and classifies during the game the encountered opponents into adversary classes which are automatically learned online. Each opponent class predicts the opponent's movements as a positional range where the opponent may be found certain time in the future together with a confidence value. This opponent modeling approach eliminates the assumption of the existence of a coach and the requirement of predefined opponent models.

Ledezma et al. [42] translated the task of acquiring the opponent model into a classification task. It used the logs of opponent team's player to predict its actions using a hierarchical learning schema. In this approach, after learning the agent action numerical parameter of its action will be learnt. The advantage of this trend is that the prediction of the opponent's action is increased since if we do not know the strength of the kick, the agent knows that it is going to kick rather than dashing. In this work Ledezma et al. considered that they had direct access to the opponent's inputs and outputs. In Another work [27] Ledezma and colleagues extend previous approach in the simulated robosoccer domain by removing this assumption. To do so, they have used machine learning to create a module that is able to infer the opponent's actions by means of observation. Next, this module can be used to label opponent's actions and learn a model of the opponent based on their observed input and output behavior, described by low-level actions. In this work, decision trees are learnt to predict the player's action type.

Continuing the previous works, in [43] Ledezma and his colleagues proposed an approach to model low-level behavior of individual opponent agents. In this work, the goalie actions were anticipated by a striker agent using OMBO (Opponent Modeling Based on Observation) so that the striker got as close to the goal as possible and shoot when the goalie was predicted to move. OMBO used machine learning techniques for opponent modeling at three levels. Mapping sensory data into discrete actions build the opponent model, and generate the decision-making algorithm.

The Modeling task itself consisted of two modules; Action Labeling Module (ALM) and Model Builder Module (MBM). In ALM, the last action (and its parameters) performed by any robosoccer opponent will be labeled based on the observations performed by the agent that is going to build the model. The MBM will then label other agent's actions. Once a tagged log of sensory data from agents and their performed actions was ready, the model of the opponent will be created based on data inside of the ALM. In contrast to most activity recognition tasks, where actions are manually tagged by looking at sensory data, Ledezma et al. proposed an automatic way of tagging actions for the robosoccer simulator. Additionally, in this case, there are not "a priori" models of other agent's behaviors.

Similar to this work, Illobre in [44], proposed a method to learn the behavior of a goalkeeper based on the actions of a shooting player. The results show that the method achieves high levels of accuracy with restricted evidence and time. This

performance is achieved because of the chosen default assumption of the goalkeeper speed. Another problem is that the movement of the opponent goalkeeper might depend on the position of other opponent players.

4 Conclusion

In this paper, we have explored several techniques of opponent modeling in the competitive domain of RoboCup soccer simulation 2D. From our review it can be inferred that the problem of opponent modeling can be defined in various ways in an application dependent manner. In some applications it is important to model the opponent's formation or game plays, while in some others, the agents' individual behaviors are of interest. Once the opponent's model is defined, these methods usually classify data to fit the defined models using different machine learning methods. The data can vary from low level numerical logs to high level symbolic representations. The so-called coach agent considers this model to make proper counteraction during the decision making process. Although, in theory opponent modeling can be very useful, in practice it is both difficult to accurately do and to effectively use to improve game play. In summary, it was demonstrated that opponent modeling can be definitely applied and enjoy success in improving team or player performance, but it is still an open challenge in adversary games.

References

1. Turocy, T. L., Stengel, B.V.: Game Theory. CDAM Research Report LSE-CDAM. (2001)
2. Ball, D., Wyeth, G.: Classifying an Opponent's Behaviour in Robot Soccer. Proceedings of the 2003 Australasian Conference on Robotics and Automation (ACRA), (2003)
3. Fathzadeh, R., Mokhtari, V,. Kangavari, M. R., Mousakhani, M.: Opponent provocation and behavior classification: A machine learning approach. In Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.), RoboCup 2007. LNCS (LNAI), vol. 5001, pp. 540-547. Springer, Heidelberg (2008)
4. Iglesias, J.A., Ledezma, A., Sanchis, A.: A Comparing Method of Two Team Behaviors in the Simulation Coach Competition. In Torra, V., Narukawa, Y., Valls, A., Domingo-Ferrer, J. (eds.), MDAI, vol. 3885 of LNCS, pp. 117–128, Springer, (2006)
5. Kitano, H., Tambe, M., Stone, P., Veloso, M., Coradeschi, S., Osawa, E., Matsubara, H., Noda, I., Asada, M. : The RoboCup Synthetic Agent Challenge 97. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, pp, 24–29, (1997)
6. Noda, I., Matsubara, H., Hiraki, K., and Frank, I. : Soccer server: A tool for research on multi-agent systems Applied Artificial Intelligence, 12(2–3):233–250, (1998)
7. Nakashima, T., Uenishi, T., Narimoto, Y. : Off-line learning of soccer formations from game logs. In : World Automation Congress (WAC) , pp. 1-6, (2010)
8. Marín, C.A., Castillo, L.P., Garrido, L.: Dynamic Adaptive Opponent Modeling: Predicting Opponent Motion while Playing Soccer. In: Eduardo Alonso, E., Guessoum, z. (eds.), Fifth European Workshop on Adaptive Agents and Multi-agent Systems Proceedings. LIP6. Paris, France. March (2005)
9. Fyfe, C., Tiño, P., Charles, D., García-Osorio, C., Yin, H.: Intelligent Data Engineering and Automated Learning - IDEAL 2010, 11th International Conference, Paisley, UK, September 1-3, 2010. Proceedings Springer (2010)

10. Laviers, K., Sukthankar, G., Klenk, M., Aha, D.W., Molineaux, M.: Opponent modeling and spatial similarity to retrieve and reuse superior plays. Proceedings of the Workshop on Case-Based Reasoning for Computer Games, AAAI Press, California, USA, (2009)

11. The RoboCup Soccer Simulator, http://sourceforge.net/projects/sserver/files/rcssmonitor/

12. Almeida, R., Reis, L.,: Analysis and Forecast of Team Formation in the Simulated Robotic Soccer Domain. In L. Seabra Lopes et al. (Eds.): EPIA 2009, LNAI 5816, pp. 239–250, 2009. Springer-Verlag Berlin Heidelberg (2009)

13. Riley, P., Veloso, M. : On Behavior Classification in Adversarial Environments. In Lynne E. Parker, George Bekey, Jacob Barhen (eds.): Distributed Autonomous Robotic Systems 4, Springer, Heidelberg, Germany, pp. 371-380 (2000)

14. Riley, P., Veloso, M. : Recognizing Probabilistic Opponent Movement Models. In Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup-2001: Robot Soccer World V LNCS, Vol. 2377, pp. 205-245, Springer-Verlag (2002)

15. Riley, P., Veloso, M. : Coaching a Simulated Soccer Team by Opponent Model Recognition. In Proceedings of the fifth international conference on Autonomous agents (Agents2001), pp. 155-156, (2001)

16. Riley, P., Veloso, M.: Planning for distributed execution through use of probabilistic opponent models. In Proceedings of the IJCAI-2001Workshop PRO- 2: Planning under Uncertainty and Incomplete Information, pp. 72–81 (2001)

17. Veloso, M.M., Pollack, M.E., Cox, M.T.: Rationale-Based Monitoring for Planning in Dynamic Environments. In Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems. (1998)

18. Huang, Z., Yang , Y., Chen, X.: An approach to plan recognition and retrieval for multi-agent systems. Proc of AORC, Sydney, Australia. Jan (2003)

19. Iglesias, J.A., Ledezma, A., Sanchis, A.: Caos coach 2006 simulation team: An opponent modeling approach. Computing and Informatics 28(1), 57–80 (2009)

20. Iglesias, J.A., Ledezma, A., Sanchis, A.: Comparing behavior in agent modeling task. *Structure*, pp. 289-296, (2006)

21. Iglesias, J.A., Ledezma, A., Sanchis, A., Kaminka, G.A.: Classifying efficiently the behavior of a soccer team. In: Burgard,W., et al. (eds.) IAS-10, pp. 316–323 (2008)

22. Steffens, T.: Feature-based declarative opponent-modeling. In: Polani, D., Brown- ing, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 125–136. Springer, Heidelberg (2004)

23. Steffens, T.: Similarity-based opponent modeling using imperfect domain theories. In: CIG. (2005)

24. Ahmadi, M., Lamjiri, A.K., Nevisi, M.M., Habibi, J., Badie, K.: Using a two-layered case-based reasoning for prediction in soccer coach. In Proceedings of the International Conference on Machine Learning; Models, Technologies and Applicatio*ns*, p. 181–185, (2004)

25. Kaminka, G.A., Fidanboylu,M., Chang, A., Veloso, M.M.: Learning the sequential coordinated behavior of teams from observations. In: Kaminka, G.A., Lima, P.U., Rojas, R. (eds.) RoboCup 2002. LNCS (LNAI), vol. 2752, pp. 111–125. Springer, Heidelberg (2003)

26. Lattner, A.D., Miene, A., Visser, U., Herzog, O.: Sequential pattern mining for situation and behavior prediction in simulated robotic soccer. In: RoboCup 2005: Robot Soccer World Cup IX. pp. 118–129, (2006)

27. Ledezma, A., Aler, R., Sanchis, A., Borrajo, D.: Predicting opponent actions by observation, in RoboCup- 2004: The Seventh RoboCup Competitions and Conferences. Berlin: Springer Verlag, (2005)

28. Ramos, F., Ayanegui, H.: Discovering Tactical Behavior Patterns Supported by Topological Structures in Soccer Agent Domains. In: International Conference on

Autonomous Agents, Proceedings of the 7th International joint conference on Autonomous Agents and Multiagent Systems, Estoril, vol. 3, pp. 1421–1424 (2008)

29. Fathzadeh, R., Mokhtari, V., Mousakhani, M., Mahmoudi, F.: Mining Opponent Behavior: A Champion of RoboCup Coach Competition. In IEEE 3rd Latin American Robotics Symposium, pp. 80-83, (2006)

30. Fathzadeh, R., Mokhtari, V., Shahri, A.M.: Coaching with Expert System towards RoboCup Soccer Coach Simulation. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020. Springer, Heidelberg (2006)

31. Fathzadeh, R., Mokhtari, V., Haghighat, A.T., Mousakhani, M.: Using expert system in robocup soccer coach simulation: An opponent modeling approach. In Proceedings second IEEE Latin-American Robotics Symposium, Sao luis-Maranhao, Brazil, (2005)

32. Bombini, G., Mauro, N.D., Ferilli, S., Esposito, F.: Classifying Agent Behavior through Relational Sequential Patterns. In: Jedrzejowicz, P., Nguyen, N.T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2010, vol. 6070, pp. 273-282 (2010)

33. Reis, L.P., Lopes, R., Mota, L., Lau, N.: Playmaker: Graphical Definition of Formations and Setplays. In Information Systems and Technologies (CISTI), pp. 1-6, (2010)

34. Uenishi, T., Nakashima, T.: Team Description of opuCI 2D for RoboCup (2009)

35. Ayanegui-Santiago, H.: Recognizing Team Formations in Multi-agent Systems: Applications in Robotic Soccer. Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, p. 163–173, (2009)

36. Faria, B.M., Reis, L.P., Lau, N., Castillo, G.: Machine Learning Algorithms applied to the Classification of Robotic Soccer Formations and Opponent Team. Proceedings of the 2010 IEEE Conference on Cybernetics and Intelligent Systems (CIS) and Robotics, Automation and Mechatronics (RAM), Singapore, pp. 344-349 (2010)

37. Visser, U., Drucker, C., Hübner, S., Schmidt, E., Weland, H.G.: Recognizing for-mations in opponent teams. In: Stone, P., Balch, T., Kraetzschmar, G.K. (eds.) RoboCup 2000. LNCS (LNAI), vol. 2019, pp. 391–396. Springer, Heidelberg (2001)

38. Riley, P., Veloso, M., Kaminka G.: An empirical study of coaching. In Asama, H., Arai, T., Fukuda, T., Hasegawa, T. (eds.), Distributed Autonomous Robotic Systems 5, pages 215–224. Springer-Verlag, 2002.

39. Kuhlmann, G., Stone, P., Lallinger, J.: The champion UT Austin Villa 2003 sim- ulator online coach team. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020. Springer, Heidelberg (2004)

40. Weka. Weka Machine Learning Project, http://www.cs.waikato.ac.nz/~ml/index.html (acessed: October 04, 2008)

41. Stone, P., Riley, P., Veloso, M.: Defining and Using Ideal Teammate and Opponent Agent Models. In Proceedings of the Twelfth Annual Conference on Innovative Applications of Artificial Intelligence, (2000)

42. Ledezma, A., Aler, R., Sanchis, A., Borrajo, D.: Predicting opponent actions in the RoboSoccer. In IEEE International Conference on Systems, Man and Cybernetics, p. 5, (2002)

43. Ledezma, A., Aler, R., Sanchis, A., Borrajo, D.: OMBO: An opponent modeling approach. AI Communications, vol. 22, p. 21–35, (2009)

44. Illobre, A., Gonzalez, J., Otero, R., Santos, J.: Learning action descriptions of opponent behavior in the Robocup 2D simulation environment. ILP (2010).

45. Chen, M., Foroughi, E., Heintz, S., Kapetanakis, S., Kostiadis, K., Kummeneje, J., Noda, I., Obst, O., Riley, P., Steffens, T., Wang, Y., Yin, X.: RoboCup Soccer Server manual for Soccer Server version 7.07 or Latest, http://sourceforge.net/projects/sserver (acessed on: October 01, (2003)