The RoboCup Agent Behavior Modeling Challenge

José Antonio Iglesias
Carlos III University
Madrid
jiglesia@inf.uc3m.es

Agapito Ledezma
Carlos III University
Madrid
ledezma@inf.uc3m.es

Araceli Sanchis
Carlos III University
Madrid
masm@inf.uc3m.es

Abstract

RoboCup is an international joint project that aims to foster Artificial Intelligence (AI) and intelligent robotics research by providing a standard problem. RoboCup offers different challenges for intelligent agent researchers in a dynamic, real-time and multi-agent domain. One of these challenges, especially in the *Simulation League*, is the opponent modeling, which is crucial for the ultimate goal of the RoboCup project: *develop a team of fully autonomous.*

In order to emphasize opponent-modeling approaches, the *RoboCup Coach Competition* was created and it was held every year (with some changes) from 2001 to 2006. Although there were several interesting research works about the agent modeling challenge during that time, several considerations were not well defined and the competition was suspended after *RoboCup Coach Competition 2006*. In this paper, we propose a new approach for the competition to face the opponent modeling challenge in the RoboCup competition.

1 Introduction

In a Multi-Agent System (MAS), the adaptation of an agent to the environment is essential to be successful in complex domains, especially to the current behavior of other agents. This adaptation should occur at all levels of strategy, from individual reactive behaviors to team strategy. In complex domains with no clear optimal policy, the more quickly and effectively the agents can adapt to the new situation, the better they will perform [1].

To recognize the behavior of the opponent in a MAS, it is necessary some kind of model of the opponent. Opponent modeling is a skill in a MAS which attempts to create a model of the behavior of the opponent. This model can be used to predict the future actions of the opponent and generate appropriate strategies to play against it.

Robot World Cup (RoboCup) is an international joint project that aims to foster AI and intelligent robotics research by providing a standard problem. RoboCup was proposed in 1994 and three years later Kitano et al. [2] proposed four different technical RoboCup Challenges: (1) The RoboCup Learning Challenge, (2) The RoboCup Teamwork Challenge, (3) The RoboCup Opponent Modeling Challenge, and (4) the Managing Challenges.

This paper focus on the RoboCup Opponent Modeling Challenge which calls for research on modeling a team of opponents in a dynamic, multi-agent domain. The modeling issues in RoboCup can be broken into three parts: On-line tracking (involves individual players´real-time, dynamic tracking of opponents´goals and intentions based on observations of actions), On-line strategy recognition ("Coach" agents for teams may observe a game from the sidelines, and understand the high-level strategies employed by the opposing team) and Off-line review ("Expert" agents may observe the teams playing in an after-action review, to recognize the strengths and weaknesses of the teams, and provide an expert commentary) [2].

Taking into account these challenges and because the dynamical and adversarial nature of a soccer play, opponent modeling has been very relevant in the RoboCup environment, especially in the simulation league. In 2001, a new competition was created: *RoboCup Coach Competition*, in which an on-line *coach* was able to act as an advice-giving agent [3]. In order to improve the behavior of the coached team, the *coach* could receive a global view of the world environment and communicate with the team. *RoboCup Coach Competition* changed in 2005 in order to emphasize opponent-modeling approaches. The main goal of that new competition was to model the behavior of a soccer team. A play pattern (way of playing soccer) was activated in a test team and the *coach* should detect this pattern and then, recognize the patterns followed by a team by observation. That competition was held in 2005 and 2006.

Nowadays, there is not a special competition for this task but some simulation soccer teams are developed taking opponent-modeling into account as a winning advantage [4]. There exist several research works that present different methods to create an opponent model in the RoboCup environment. How these models can impact the performance of teams is an essential aspect. However, how this impact can be measure is a complicated task. In this paper, we propose a new approach to face the opponent modeling challenge in the RoboCup competition: *The RoboCup Agent Behavior Modeling Challenge.*

2 Background and Related Work

In general, an opponent model is an abstracted description of the behavior of one or several players in a game. The beginning of opponent modeling is a work done in 1996 by Carmel and Markovitch [5], in which it is introduced the M* algorithm, a generalization of the minimax algorithm that uses an arbitrary opponent model to benefit from its flaws. In addition, in [6], it is assumed that agent´s strategies can be modeled as finite automate and a *model-based* approach is presented as a method for learning effective interactive strategies.

2.1 Opponent Modeling in the RoboCup Simulation

Although there are lot of dynamic multiagent domains with adversary agents, in which opponent modeling can be implemented efficiently; in this paper we focus on the opponent modeling in the *RoboCup Soccer Simulation* environment. This environment provides a good platform for modeling a team of opponents in a dynamic, multi-agent domain and a large number of publications related to the *RoboCup Soccer Simulation League* have been publicated[1] from 1996. In this environment many approaches, in which each agent observes and recognizes the behavior of the adversaries [7] have been proposed.

However, a frequently used opponent modeling approach in the RoboCup environment is to rely on an omniscient agent (*Coach*) to recognize the opponent behavior and to communicate to the team agents the model of the opponent or an strategy for that model. In addition, in order to focus entirely on opponent modeling the *RoboCup Simulation Coach Competition* was held in 2001, 2002, 2003, 2005 and 2006. This competition is situated within the same soccer server, but instead of creating a full soccer team, a single *coach* agent (which has a full view of the field but only can advice to its team via the standardized language called Clang [8]) must be implemented. The main advantages of the *coach* is that it can *see* the field noise-free and it has access to logfiles of past games played by the team to model (opponent).

In the first three coach competitions (2001, 2002 and 2003), the opponent logfiles (recordings of their past games) could be analyzed during 24 hours. As a result, most of the coaches were created entirely by hand and in those cases, the purpose of encouraging automated opponent modeling was not achieved. However, in this environment, interesting researches were carried out: Riley et al. [9] presented several implemented coaching techniques for a simulated robotic soccer domain and justified that coaching can help teams im-

[1] http://www.cs.utexas.edu/~pstone/tmp/sim-league-research.pdf

prove in this domain. Kuhlmann et al. [10] presented a multi-facted learning approach to give advice in RoboCup simulated soccer and identify the learned formation rules as the most effective type of advice. Dulalia et al. [11] developed a system (*SimSoccer Coach*) that shows a single agent learning by analyzing the fixed opponent´s behavior and then providing offensive and defensive advice to improve the performance of the team.

RoboCup Coach Competition changed in 2005 to emphasize opponent-modeling approaches. In this competition the coach agents were directly evaluated based on how they model a team which performs a pattern (predictable and exploitable). After modeling the team, the coach is rated on how well it recognizes the pattern. This environment is the base for a large number of works: Kuhlmann et al. [12] modeled a soccer team by characterizing their behavior with a set of features calculated from statistics gathered while observing a game. The winners of the RoboCup Coach 2006 Competition (Ramin Fathzadeh et al.) presented in [13] a novel learning architecture for modeling the opponent and a rule based expert system architecture to provide a strategy for opponent players. Recently, Iglesias et al. [14] presented a novel method used by the CAOS team to model and recognize successfully the behavior of a soccer team.

Very related with this competition and due to the importance of the evaluation of agent teamwork is crucial, the 2D simulation environment is a *good* environment for researches about this aspect:Kaminka et al. [15] presented a hybrid approach to learning the coordinated sequential behavior of teams, from a time-series of continuous multi-variate observations, of multiple interacting agents. Bezek et al. [16] presented a domain-independent framework (*MASDA*) for discovering strategic behavior of multi-agent systems. *MASDA* was evaluated only on the RoboCup domain, but domain-specific knowledge can be introduced in the form of role, action and domain feature taxonomies.

2.2 RoboCup Simulation Tools

RoboCup Simulation Competition uses a simulator (*rcssserver*) which is a network-based graphical simulation environment for multiple autonomous mobile robots in a 2D space. Robocup simulation games are recorded as *log files*, in which the positions of the ball and all players for both teams at each simulation cycle are stored. In addition, the RoboCup Soccer Simulator Monitor, called *rcssmonitor* (Figure 1), is used to view a simulation as it takes place by connecting to the *rcssserver* or to view the playback of a simulation by connecting to the *rcsslogplayer*.

Figure 1: The RoboCup Coach Soccer Server Monitor (*rcssmonitor*).

During the last years, there have been created many sophisticated tools for analyzing robocup simulated games:

Virtual RoboCup [17] is a real-time 3D visualization tool for 2D simulated soccer games as played in the RoboCup simulation league. Players are modeled as anthropomorphic animated figures. **LogMonitor** [18] is a tool for analyzing games from logfiles and displaying statistical data such as counts of soccer plays. **Team Assistant** [19] is a logplayer/debugger/analyzer for RoboCup soccer simulation. The analyzer proposed in this tool recognizes different events and graphically display them on the field. **Logalyzer** [20] is a powerful tool for visualization and analysis of RoboCup log files which provides information about detected actions and several visualizations for the collected data about a soccer game. In addition, this tool creates abstract action models using action graphs.

In order to help to face the RoboCup Agent Modeling Challenge, we have developed a visual tool called *Viena*[2]. The main purpose of *Viena* is to analyse automatically an observed team behavior. However, this tool is also proposed as: 1) graphical visualization of a RoboCup soccer game, 2) dynamic data (position, velocity, stamina...) representation and visualization, 3) dynamic configuration of different visual parameters, 4) detection of soccer actions (pass, intercept, shoot..), 5) analysis of a soccer team behavior by using different algorithms.

3 RoboCup Coach Competition 2006

The *RoboCup Coach Competition* structure has been changing gradually from 2001. In this section, we describe in detail the structure of the competition held in Bremen (Germany) in 2006. In addition, we present the main faults and shortcomings of this structure.

3.1 RoboCup Coach Competition 2006 Structure

Before describing the structure of this competition, the definition of two phrases are presented:

- *Play Pattern:* This term is used to describe a simple behavior that a team performs which is predictable and exploitable for the coach. In this paper we use the term pattern as a contraction of play pattern.
- *Base Strategy:* The general strategy of the test team regardless of the pattern in it.

According to the *RoboCup 2006 Coach Competition* official rules, previously to the competition, a set of strategies to be used as the base strategies of the patterns are created by the organizing committee and some games are played (*no-pattern log files*). Then, the patterns are added to these base strategies, and some sample games are played again (*pattern log files*). Many pairs of log-files (*pattern log file* and its corresponding *no-pattern log file*) are created.

At the beginning of the competition, each coach team participant is provided with some[3] pattern game log-files (only one pattern is activated in a log-file) and its corresponding no-pattern game log-files (games with the same base strategy but the pattern not activated). The main goal of the current competition is to look for the qualitative differences among the pattern log file and its corresponding no-pattern log file. The coach should detect the patterns followed by the test team in the pattern log files and report them. Also, once every pattern has been detected and stored, the coach should recognized them by observing a live game. Therefore, this competition consists of two phases:

- *Offline Analysis:* The inputs of this phase are several *pattern log files* and its corresponding *no-pattern log files.* The coach has to look for the qualitative differences between the two log files in order to detect the pattern. The output of this analysis is a set of files (*Pattern Library*) where the specifications of all the patterns are recorded.
- *Online Recognition:* The coach observes a live game where some patterns have been activated in the test team. The coach should recognize on-line the patterns activated in the test team in a 6000 cycle game and report them. The sooner the coach sends the report, the more score it gets.

The Figure 2 shows the overview structure of the RoboCup Coach Competition. Hence, the performance of a given coach is based only on its ability to detect and report patterns. The research focus is on team/opponent modeling and on-line recognition. The coaches

[2] Available at: http://www.caos.inf.uc3m.es/~viena/

[3] The number of pairs of log files received by the coach for one round is around 20.

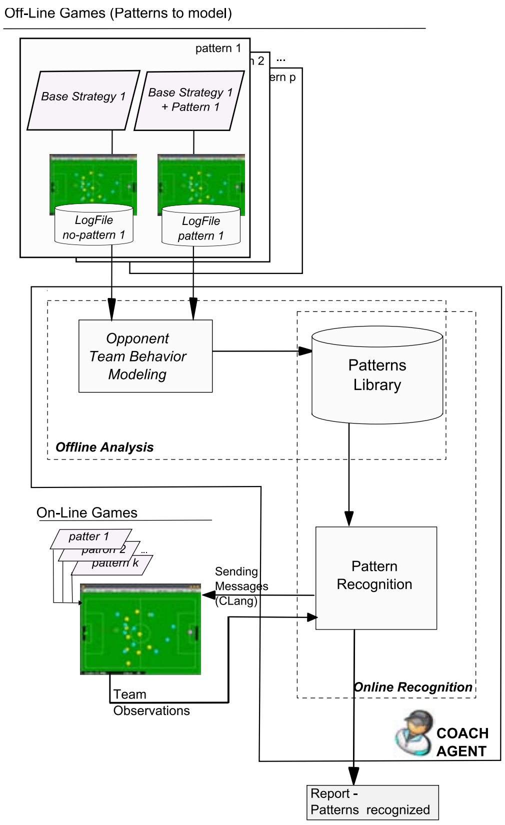

Figure 2: The RoboCup Coach Competition. Overview structure.

work both by analyzing logs of previous games and by observing and recognizing while a game is running.

3.2 Faults and shortcomings of the RoboCup Coach Competition 2006

After participating in this competition, we realized that there were some faults and shortcomings in the proposed structure, which we detail in this section.

Although the competition rules describe the characteristics of a pattern (predictable and exploitable for the coach), they are very general and it is difficult to detect when a pattern fulfills these requirements. The construction of these patterns is a hard part of the competition mainly because they can only be defined using CLang, which restricts the description of a team behavior. The CLang advice language was created by the RoboCup community as a standard language so that coaches could effectively talk to and work with teams from other research groups. CLang is a simple declarative programming language with a set of domain-specific terms composed using a primarily prefix notation like LISP. Tactics and behaviors are distributed as *if-then rules*, which consist of a *condition* description and a list of *directives* that are applicable when the condition is true. Directives are lists of *actions* that individual sets of players should or should not take. There are expressions for soccer-specific entities such as *regions* and *points*.

The *Pattern Library* stores the different patterns which have been created. However, how well these patterns have been created is not evaluated in the final score because the pattern does not need to be described.

One of the biggest problems during the competition in the on-line game, was to detect a pattern when it was executed a limited number of times. Although any player in the on-line phase only followed a specific pattern, sometimes it can not be executed for different reasons (e.g., the player never gets the ball and its corresponding pattern can not be executed).

The patterns are reported using its corresponding number, without considering the similarity between the "real pattern" and the created pattern by the coach.

Finally, as the performance of a given coach is calculated only considering the number of activated patterns reported correctly, the detection of patterns is not used to adapt the behavior of the soccer team during a game. However, this adaptation should be the most important challenge in this competition.

4 The proposed RoboCup Agent Modeling Challenge

In order to use learning and modeling, we propose the structure shown in Figure 3, which consists of two different phases:

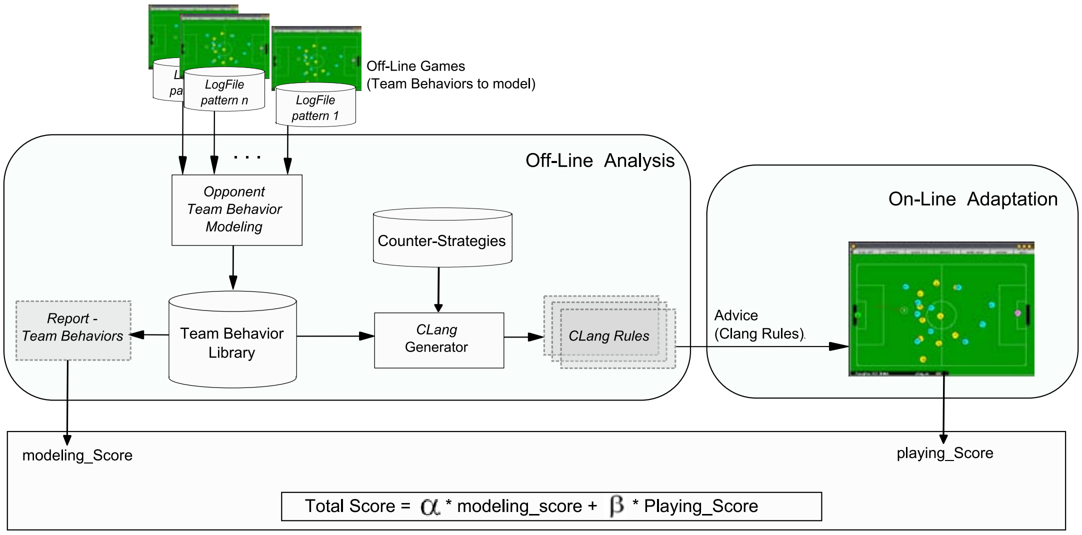

Figure 3: The proposed RoboCup Agent Modeling Challenge. Overview structure.

4.1 Off-Line Analysis

To make the behavior of the modeling team more realistic and to be sure that a team is following a pattern predictable and exploitable, we propose that the coach models a "real" team. This way, every participant creates a team that follows a specific behavior. The behavior of the team is specified by a "special" description that consider: offensive/defensive team, players positions (player and its action field area), actions executed by the players and so on. This consideration solves an important problem in the structure of the last competition: the process of pattern generation.

According to this, every participant creates a team and its corresponding "team behavior" description. Also, we consider that these teams should be good enough to win a simple but effective *coachable team*. Then, every participant plays a game against all the teams in order to find its corresponding "team behavior". The created behavior models are stored in a library using a description for each team. The first score of the competition (modeling_Score) is the similarity between the team behavior description obtained by the participant and the real team behavior description.

Also, the information extracted from the opponent team behavior should be used for the coach in the *on-line Adaptation phase* to improve the performance of its team. This improvement can be considered by creating different counter-strategies. A counter-strategy consists of several counter-actions as response to a recognized behavior. CLANG language can be used to inform and advice to the *coachable players* (players of the "coach" team) what they should do. Thus, the counter-strategies can be sent to the corresponding players by the *coach* using CLANG. This language is suited to represent strategies because its messages are basically production rules mapping conditions to actions: CLANG conditions are constructed from logical connectives (and, or, not) of descriptions of the world state like player and ball positions, play modes, scores, and time. CLANG actions are designed to have relatively clear semantics and are recommended macro-actions for the players such as *position-in-regions*, *marking*, *passing-to-regions*, *passing-to-players*, *dribbling*, *intercepting* and *tackling*. As an example, the following lines described a rule in CLANG:
1.(definer REGION1(rec(pt 30 20)(pt 40 35)))

2. (definerule RuleNumber1 direc
3. (and (bowner opp 2) (playm play_ on) (bpos REGION1))
4. (do our 2 (markl7)))

The first line defines a region (a rectangle area) in the field. In the next 3 lines, the rule named *RuleNumber1* is defined: The second line is the beginning of the rule and it is due to the *coach* protocol. In the third line the situation description is detailed and denotes that: the opponent player 2 has the ball, the play mode is *on*, and the ball is in a region defined in the first line. In the last line, the action is specified: the player 2 marks the opponent player 7 (marking is a standard soccer term meaning to play defense against a player).

Finally, as we can see in the off-line phase of figure 3, to generate the CLang rules, several "counter strategies" are used. This file of strategies can be generated previously to the competition by a learning process. In this phase, the developed visual tool called Viena can be very useful.

4.2 On-Line Adaptation

One of the main aim of this challenge is to create a team that is able to exploit the opponent behavior model on-line. This aim was not considered in the RoboCup Coach Competition 2006 because the result of the on-line game was not part of the the final score of the participants.

The goal of this phase in the proposed structure is to evaluate the CLang rules created in the previous phase. Its goal is to adapt the behavior of the *coachable team* to the opponent team taking into account the previous team behavior detection process. For this reason, a *coachable team* (that previously received the CLang rules) plays versus one of the previous modeled teams.The score of this part is the result of the soccer game.

4.3 Evaluation

The final score of a participant consists on (1) how well the coach models the behavior of the opponent team and (2) how well the *coachable team* exploits the behavior models and adapts its behavior to the opponent team behavior. To calculate the final score we propose the following equation: $SCORE = \alpha * modeling\_Score + \beta * playing\_Score$.

5 Conclusions

Adaptation and learning abilities are essential for an intelligent agent that interacts with other selfish agents. The RoboCup (Simulation) Coach Competition is a challenging dynamic, real-time and multi-agent domain for autonomous agents that is specially geared towards opponent modeling and adaptation. In fact, we consider that this competition is the most capable competition to implement different artificial intelligent methods.

The research works in this competition are very interesting and relevant for the official goal of the RoboCup (*to create a team of fully autonomous humanoid robot soccer players that win the soccer game against the winner of the most recent World Cup*) because, as humans do, the robot players should recognize the opponent behavior and adapt its behavior in order to act optimally.

In this paper we have proposed a new structure for future RoboCup Coach Competitions which solves some of the faults and shortcomings of this competition in 2006. The proposed structure takes into account the opponent team behavior modeling and the use of these models to adapt to the opponent behavior. This structure raises many interesting questions which will continue to pursue.

References

[1] P. Riley, M. M. Veloso, On behavior classification in adversarial environments, in: Proceedings of Distributed Autonomous Robotic Systems 4, 2000, pp. 371–380.

[2] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, H. Matsubara, Robocup: A challenge problem for ai and robotics., in: RoboCup, 1997, pp. 1–19.

[3] I. Noda, H. Matsubara, K. Hiraki, I. Frank, Soccer server: a tool for research on multiagent systems, in: Applied AI, Vol. 12, 1998, pp. 233–258.

[4] A. M. Fard, M. H. Ansari, A. Ildermi, S. Molavipour, M. Aledaghi, F. Seifi, M. Naghibzadeh, Nexus 2d 2008 team description, in: IEEE Latin American Robotics Competitions - RoboCup Brazil Open 2008, 2008.

[5] D. Carmel, S. Markovitch, Incorporating opponent models into adversary search, in: Proceedings of the National Conference on Artificial Intelligence, Portland, Oregon, 1996, pp. 120–125.

[6] D. Carmel, S. Markovitch, Opponent modeling in MAS, in: Adaptation and Learning in Multi-Agent Systems, 1996, pp. 40–52.

[7] M. Wünstel, D. Polani, T. Uthmann, J. Perl, Behavior classification with self-organizing maps, in: RoboCup 2000: Robot Soccer World Cup IV, Springer-Verlag, London, UK, 2001, pp. 108–118.

[8] M. Chen, E. Foroughi, F. Heintz, Z. Huang, S. Kapetanakis, K. Kostiadis, J. Kummeneje, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang, X. Yin, Soccerserver manual ver. 7 (2002).

[9] P. Riley, M. Veloso, G. Kaminka, An empirical study of coaching, in: Distributed Autonomous Robotic Systems 5, Springer-Verlag, 2002, pp. 215–224.

[10] G. Kuhlmann, P. Stone, J. Lallinger, The champion UT Austin Villa 2003 simulator online coach team, in: RoboCup-2003: Robot Soccer World Cup VII, Springer Verlag, Berlin, 2004.

[11] C. L. Dulalia, P. S. L. Go, P. V. C. Tan, M. Z. I. O. Uy, R. de Dios Bulos, Learning in coaching, in: Proceedings of the fourth IEEE International Workshop, WSTST-05, 2005, pp. 1338–1347.

[12] G. Kuhlmann, W. B. Knox, P. Stone, Know thine enemy: A champion robocup coach agent, in: Proceedings of the National Conference on AI, 2006, pp. 1463–68.

[13] R. Fathzadeh, V. Mokhtari, M. R. Kangavari, Opponent provocation and behavior classification: A machine learning approach, in: RoboCup 2007: Robot Soccer World Cup XI, 2007, pp. 540–547.

[14] J. A. Iglesias, A. Ledezma, A. Sanchis, CAOS Coach 2006 Simulation Team: An opponent modelling approach, Computing and Informatics 28 (1) (2009) 57–80.

[15] G. A. Kaminka, M. Fidanboylu, A. Chang, M. M. Veloso, Learning the sequential coordinated behavior of teams from observations, in: RoboCup 2002: Robot Soccer World Cup VI, Springer, 2002, pp. 111–125.

[16] A. Bezek, M. Gams, I. Bratko, Multiagent strategic modeling in a robotic soccer domain, in: AAMAS '06: Proceedings of the fifth conference on Autonomous agents and multiagent systems, ACM, New York, NY, USA, 2006, pp. 457–464.

[17] B. Jung, M. Oesker, H. Hecht, Virtual RoboCup: Real-Time 3D Visualization of 2D Soccer Games, in: International Workshop on RoboCup, IJCAI'99, Stockholm, 1999, pp. 121–126.

[18] T. Takahashi, Logmonitor: From player's action analysis to collaboration analysis and advice on formation, in: RoboCup-99: Robot Soccer World Cup III, London, UK, 2000, pp. 103–113.

[19] R. S. F. A. S. Eslam Nazemi, Amir-Reza Zareian, Team assistant - team description paper, in: RoboCup 2002: Robot Soccer World Cup VI, 2002.

[20] A. Bezek, Modeling multiagent games using action graphs, in: Proceedings of Modeling Other Agents from Observations, 2004.