Scenario-based Teamworking, How to Learn, Create, and Teach Complex Plans?

Ali Ajdari Rad[1], Navid Qaragozlou[1], Maryam Zaheri[2], Emad Bahrami[3], Vahid Ghafarpour[3]

[1] Chista Software Group (www.itsi.ws)
{ali, navid}@itsi.ws
[2] Computer Science Department, University of North Carolina at Charlotte, USA
mzaheri@uncc.edu[2]
[3] Computer Engineering and IT Department, Amirkabir University of Technology
{vahidg, emadb}@aut.ac.ir

Abstract. This paper presents a wise commander that uses a novel method in the multi-agent teamwork field called Scenario-based Teamworking (SBT) to improve performance of its teammate's cooperation. In SBT method a team of cooperative intelligent agents could be able to execute complex plans in nondeterministic, adversary, and dynamic environments which communication cost is high. The base idea of this method is to define Scenario for different situations. With a graph of scenarios, a team of agents can execute, learn, adapt, and create team plans automatically. This method has implemented in a soccer team of intelligent agents (players and coach) and evaluated with standard RoboCup simulator environment and results show a significant improvement. This paper submitted as team description of ItSi team for coach league in RoboCup world cup 2004 competitions.

1 Introduction

One of the most important and complicated problems in designing multi-agent systems is the agents teamwork. A team of intelligent agents without cooperation will not going to act as well as a team of agents with less individual intelligence but teamwork understanding. The more complex environment and the higher cost for communication among agents make it harder to design a method for teamwork and managing agents.

The creation of the robotic soccer, the robot world cup initiative (RoboCup), is an attempt to foster AI and intelligent robotics research by providing a standard problem where wide range of technologies can be integrated and examined (Riley and Veloso 2000). Some of the fields covered include multi-agent collaboration, strategy decision making, intelligent robot control and machine learning (Noda et al. 1998). Similar to real soccer games, the simulated teams can have a coach. The duty of the coach is to employ appropriate tactics based on abilities of teammates and also the strategy of the opponents. Furthermore, the coach is responsible to finding the weakness of its own team and improves their teamwork by applying appropriate strategies (Pourazin, Ajdari, and Atashbar 1999).

In this paper we present a powerful method to define plans for a team of soccer player agents called Scenario-based Teamworking. The main idea in this method was introduced by the Canned Plans concept of Essex Wizard team (Kalyviotis and Hu 2001). Using scenario–based approach, player agents are able to learn, adopt and execute complex team plans against their opponent, and the coach agent is able to modify and teach plans to its players. Another advantage of SBT is opponent modeling and the capability of automatically creating new plans.

The paper is organized as follows. Section 2 describes SBT idea in general. Finally, in section 3, we describe the conclusion and our plans for future work till world cup games.

2 Teamwork based on Scenarios

The SBT method is based on the concept of scenario. In this method we describe a scenario for each plan of team. Each scenario including Triggers, Goal, Abort conditions, Evaluation parameters, and Side effect. Triggers are conditions that explain the current situation of the environment and internal state of the agent. These conditions are divided as follows:

- Data Triggers: facts, which made by direct information about the environment. For example, agent is in area #1, or ball is on opponent side.
- Time Triggers: concepts, which are dependent on time, such as playing modes.
- Communication Triggers: situations that are affected by agents' communications, such as "Agent #1 said something" or "Agent #2 sent a pass request."
- Action Triggers: situations, which are related to an agent's action, such as owning the ball, or shooting the ball.
- Situation Triggers: Conditions that are brought about by what happens in the game. Usually these conditions are related to the high level concepts in soccer such as attack mode, or crowding around the ball.

Goal describes the final goal of the plan. In RoboCup field, we categorized the final goal as Scoring, Clearing, and Possession (when the team is ball owner), Blockade, Close goal, and Close pass (when the opponent is ball owner). Scoring scenarios occur near opponent goal and describe a plan that its final action is shooting to opponent goal and scoring. Clearing scenarios occur near home goal and describe a plan that aims to kick the ball away from home goal. Possession scenarios occur in middle of field and aim to keep the ball and create a chance to achieve Scoring situation. Blockade scenarios are selected when some agents want to obtain the ball from opponents (make pressure on ball). In Close goal scenarios, agents will close home goal, so the opponent ball owner couldn't score. The final goal of Close pass is to force the opponent ball owner to keep the ball or make a bad pass with closing its useful pass lines.

Abort Conditions describe the conditions that abort the plan. They are defined just like Triggers (but in negative meaning). With separating these two conditions, in addition to making the concept simple, we can add some possibilities, such as approximate matching and risk management involving Triggers or Abort conditions.

Each scenario has its own evaluation parameters. They can be classified in two groups: Cost and Score. Cost is a real number that is determined by the designer in the beginning. The designer can describe the amount of cost as a function of time, power of the agent, number of agents participating in the scenario, the effects of the incomplete scenario (if this scenario fails) and other concepts. Like cost the score is a real number and shows the rewards that are obtained if the scenario finishes successfully. These parameters could be learned and changed during a game.

Executing a scenario will change the environment and make new situations. This fact can be presented in side effect concept. For example "playing in width" scenario makes the play wide, or a scenario based on fast and long passing will increase the speed of the game.

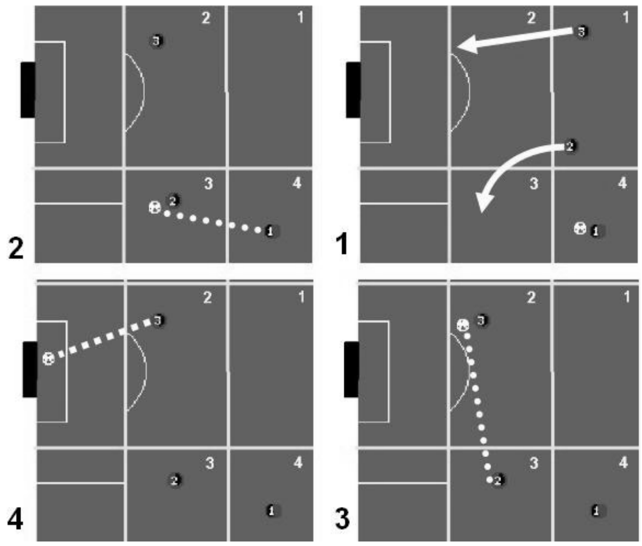

Fig 1. Sample scenario.

With respect to parts that we describe above, we define the main plan in the scenario model. Each scenario includes a set of sub-plans that are performed step-by-step. Each step includes actions; abort conditions, and triggers of agents who play in that step. For example a scenario, which its main goal is scoring, is shown here (figure 1).

In the real game three players in the above condition can choose this scenario even with the lowest and least accuracy information. Choosing a scenario identifies the agent's roles and then there will be a matching procedure to match players with agents in the scenario. In this case each player knows its role, and the player can do it without extra communication (in fact with a few communications).

A major problem in the other methods (like Canned Planes) was incompatibility of plans with the new situation or new opponent's plan. In SBT model, while a scenario is performed in a game and coach detects it is successful, its score increases with a coefficient by coach. If the scenario fails, then its score decreases with a coefficient. In this way coach can have a kind of adaptation during the game and transfers new scenario scores to players. Also a connected graph of scenarios is used to adapt sequence of scenarios (starting with a Clearing scenario and ending with Scoring scenario) with current game (Ajdari, Qaragozlou, Zaheri 2001). In this order, we can use the effect of more performing scenarios. At last scenarios are chosen with respect to these parameters:

- Matching of triggers with current conditions.
- Matching of abort conditions with current conditions.
- Matching of goal of scenario with local goal of players.
- Matching of side effect of scenario with general strategy of team.
- We take into consideration costs and scores.

Automatically creating new scenarios by coach is another benefit of SBT. This could be done in two ways:

- Watching the opponent's play method. The coach can model the opponents' scenarios by determining current trigger, opponent's selected actions, side effects, and other features of a scenario (maybe some of features are not determined exactly). The opponent modeling procedure could be done in similar way (Riley and Veloso 2000).
- Creating new scenario with evolutional methods by coach. New scenarios are built with random triggers and actions and then an evaluation phase determines the usability of it. Because of complexity of environment, some limited rules should be considered to reach better performance.

In a real game, according to opponent (prior knowledge about its pervious plays), one of the scenario graphs will be selected by human, before the beginning of the game (a team may have different scenario graphs for different strategy). Players select and execute scenarios and coach observes the result. Results will modify the graph so the behavior of team will be changed based on its online experiments. Then coach broadcast new strategy with standard coach language (Ajdari 2001).

3 Conclusion and Future Work

The Scenario-based approach was used in this paper for analyzing and arranging of a team of soccer player agents. Our experience shows that the SBT approach presents great performance against traditional methods (Ajdari et al. 2003, Ajdari et al. 2004). As our previous version coach had not been a standard coach, we of SBT with standard coach language; it should have migrated to standard language. Until world cup competitions, reaching all functionality and performance of previous coach in standard mode, test the standard coach with coachable players and making a larger bank of scenarios are planned for future works.

Acknowledgements

We hereby appreciate Prof. Reza Safabakhsh our advisor and Dr. Ahmed Abdollahzadeh the professors of computer engineering department of Amirkabir University for their technical advices.

References

1. Riley, P. and Veloso, M. 2001. Coaching a Simulated Soccer Team by Opponent Model Recognition. In Proceedings of the Fifth International Conference on Autonomous Agents (Agents-2001).
2. Noda, I., Matsubara, H., Hiraki, K., and Frank, “Soccer server: A tool for research on multi agent systems”, Applied Artificial Intelligence, 12:233-250, 1998.
3. Pourazin, S. Ajdari Rad, A. Atashbar, H. 1999. Pardis, Our team in RoboCup simulation league. Team description of Simulation league of robocup99, 95-97. Linkoping University, Electronic press.
4. Kalyviotis N. and Hu H. 2001. A Co-operative Framework for Strategic Planning. In proceedings Towards Intelligent Mobile Robots (TIMR) '01, Manchester.
5. Ajdari Rad, A. Qaragozlou, N. Zaheri, M. 2003. Scenario-based Teamworking, How to Learn, Create, Teach Complex Plans. To appear in proceeding of RoboCup symposium-2003, Paduva, Italy.
6. Riley, P. and Veloso, M. 2000. Towards Behavior Classification: A Case Study in Robotic Soccer. In Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), AAAI Press, Menlo Park, CA.
7. Ajdari Rad, A. 2001. Design and implementation of a soccer coach; Dynamic leading of a team of intelligent agents, BS thesis, Dept. of Computer engineering, Amirkabir University of Technology.
8. Qaragozlou, N. 2001. Design and implementation of soccer player architecture; an architecture for multi agent, dynamic, and real time systems, BS thesis, Dept. of Computer engineering, Amirkabir University of Technology.
9. Ajdari Rad, A., Safabakhsh, R., Qaragozlou, N., Zaheri, M., 2003. Teamwork Improvement using a Coach and Scenario-based Approach in Simulated Soccer. In proceeding of 5th conference on Intelligent Systems, Iran.
10. Ajdari Rad, A., Safabakhsh, R., Qaragozlou, N., Zaheri, M., 2004. Using a Coach and Scenario-based Approach in Simulated Soccer Teamwork Improvement. Submitted to The Nineteenth National Conference on Artificial Intelligence, San Jose, CA, USA.