IAS technical report series, nr. IAS-UVA-02-04

Mutual Modeling of Teammate Behavior

Jelle R. Kok and Nikos Vlassis
Computer Science Institute
Faculty of Science
University of Amsterdam
The Netherlands

In multiagent systems the action that an agent chooses depends on the state of other agents. It is therefore important to have an accurate representation of the current world state. In case the world is only partially observable, an agent does not continually perceive the state of the other agents. It is then essential to model the actions of the other agents to predict their future state.
In this paper, we introduce "Mutual Modeling of Teammate Behavior" (MMTB). This model simulates the executed action of a teammate. In combination with the known world dynamics this action can be used to determine the teammate's future state. In this model, we assume that all teammates are homogeneous and follow a policy that is common knowledge among them.
Furthermore, we will describe how MMTB is implemented in our RoboCup soccer simulation team *UvA Trilearn* and give empirical results about its effectiveness.

Contents

1 Introduction **1**

2 Mutual Modeling of Teammate Behavior **1**

3 RoboCup **2**

4 Application: UvA Trilearn **2**

5 Results **4**
5.1 Internal Testing 4
5.2 External Testing 4

6 Related Work **5**

7 Conclusion **5**

Intelligent Autonomous Systems
Computer Science Institute
Faculty of Science
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands

tel: +31 20 525 7461
fax: +31 20 525 7490

`http://www.science.uva.nl/research/ias/`

Corresponding author:

Jelle Kok
`jellekok@science.uva.nl`

1 Introduction

In multiagent systems it is desirable to have an accurate representation of the current world state. In case the other agents are only observed periodically, a model can be used to simulate their performed actions when sensory information is absent. These actions can be used to update their state.

In this paper, we introduce "Mutual Modeling of Teammate Behavior" (MMTB), a model to predict the joint action of all teammates. For each teammate its executed action is determined based on its previously perceived state and its policy, the latter being common knowledge among all teammates. The joint action, which is the combination of all individual actions, and the known world dynamics are finally used to update the world state. The same model can also be used to simulate what actions a teammate will execute in the future, which can help an agent to determine its action in the current situation.

This model is especially useful in applications in which the world is dynamically changing and is only partially observable. An example application of such a domain is robotic soccer. In this paper we will also describe our implementation of MMTB in our RoboCup soccer simulation team *UvA Trilearn* and show results of its effectiveness.

This paper is organized as follows. We begin with a general description of MMTB (Section 2). Thereafter, we will give a short introduction of the RoboCup domain (Section 3), describe how MMTB was implemented in our team *UvA Trilearn* (Section 4), and show empirical results about its effectiveness (Section 5). At the end, we will present some related work (Section 6) and conclusions (Section 7).

2 Mutual Modeling of Teammate Behavior

In this section we will describe MMTB, a model to predict the joint action of teammates based on their previously perceived state and their known policy. We consider an agent to be a teammate if it belongs to a group of agents that fulfill a common goal.

We make the following assumptions:

- The world state $s = [\theta, e]$ holds among others the state information θ of the N homogeneous teammates. Each teammate i is represented by its current state θ_i (e.g. position and velocity). The combined state of all teammates is denoted by $\theta = \{\theta_i\}_{i \in N}$. Apart from θ, the world state s also contains state information e about the rest of the environment. Note that e can contain information about other agents that are not a member of the team[1].

- The world is partially observable. However, we assume that an agent i periodically observes the index j and the state θ_j of other agents $j \neq i$, as well as some components of e. In general, an observation of an agent i induces a probability distribution over world states. Agent i takes the most likely value of this distribution as the true state s_i.

- Each agent i has a policy π_i that depends on its index. This policy maps a world state s_i to an individual action a_i. We thus assume that the policy is deterministic and depends on the individual perception of the world by agent i.

- The individual policies are assumed to be *common knowledge* among all agents [5]. In practice this implies that each π_i is predefined and encoded in all agents. This is similar in spirit to the locker-room agreement in [9].

- Explicit communication between the different agents is not allowed.

[1] In robotic soccer for instance, θ consists of the state information of all teammates, while e consists of the state information of the opponents, the ball, and the static objects on the field.

The main idea behind MMTB is that an agent can use the common knowledge assumption about the individual policies in order to *simulate* the behavior of the other agents. For example an agent i can determine the action of an agent $j \neq i$ by simulating the policy

$$a_j^t = \pi_j(s_j^t). \tag{1}$$

Since agent i and agent j have different perceptions of the world state, agent i simulates π_j by using its own perception s_i in place of s_j, yielding

$$\hat{a}_j^t = \pi_j(s_i^t). \tag{2}$$

In a sense, agent i estimates the action $\hat{a}_j^t$ by getting 'into the mind' of agent j.

Clearly, the accuracy of the method depends on the degree of similarity of the state components of s_i and s_j that are relevant to the current policy. In practical situations the policy depends only on a few components of the complete state. This means that if these components are observed with high certainty, one gets a predicted $\hat{a}_j^t$ close to the correct a_j^t.

Using MMTB an agent i is able to simulate the actions of all other agents $\hat{a}_{j\neq i}^t$ and combine them with its own performed action a_i^t into a joint action $a^t = [a_i^t, \hat{a}_{j\neq i}^t]$. This joint action can be used to update the world state using the known world dynamics

$$p(s_i^{t+1} | s_i^t, a^t). \tag{3}$$

Again, the agent i selects the most likely state s_i^{t+1} as the true world state.

In summary, MMTB can be used for predicting the future state of the agents when the amount of observation is limited. The idea is that the absence of new sensor evidence results in poor estimation of the current state, unless an agent can simulate the actions of the other agents and apply them to the transition model.

3 RoboCup

The Robot World Cup (RoboCup) Initiative is an attempt to foster AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined [7]. The goal of RoboCup is to have a team of fully autonomous humanoid robot players, which by 2050 can win a soccer game against the winner of the most recent world cup for human players. In order to achieve this goal, the RoboCup organization has introduced several leagues which each focus on different abstraction levels of the problem. One of these leagues is the *RoboCup Simulation League*. This league is based on a soccer simulation system called the *RoboCup Soccer Server* [3]. The soccer server provides a realistic multi-agent environment in which everything happens in real time. Various forms of uncertainty have been added into the simulation such as sensor and actuator noise, limited perception and noise in object movement. One of the advantages of the soccer server is the abstraction made, which relieves researchers from having to handle robot problems such as object recognition and movement. This abstraction makes it possible to focus on higher level concepts such as learning, agent modeling and strategic reasoning.

4 Application: UvA Trilearn

In this section we will describe how MMTB is used to predict the future states of the teammates in our RoboCup simulation team *UvA Trilearn* [1].

UvA Trilearn agents select a new action based on the distinction whether they are in an active or passive situation. An agent is in an active situation when it currently has an active role in the game and in a passive situation otherwise. In *UvA Trilearn* an agent has an active role when it can reach the ball faster than all other teammates. In most cases an agent will thus be in a passive situation. In this case an agent will move to a *strategic position* on the field in anticipation of the possibility of becoming active again. This position is based on the current team formation, the role of the agent inside this formation and the position of the ball on the field which serves as an attraction point. The current team formation and the role of each agent in this formation are common knowledge among all teammates. This procedure is heavily based on Situation Based Strategic Positioning (SBSP) [8]. For more details the reader is encouraged to see [1].

In an active situation an agent will first try to intercept the ball and after interception chooses between different shooting alternatives. Shooting actions that can be performed are scoring, passing the ball to a teammate, dribbling, or clearing the ball up front. The action that is selected depends on the current state of the other agents on the field. However, not all agents are perceived all the time. It is therefore important to have an accurate representation of the current state of all other agents.

The policy which maps the current world state to an action in a specific situation is thus very simple. In case an agent is the fastest player to the ball it will intercept the ball, otherwise it will move to its strategic position which is based on the position of the ball[2]. This simple decision procedure makes it possible for all agents to easily derive the intentions of their teammates in their current situation.

MMTB is applied in the following way. Each agent i keeps track of the world state $s_i = [\theta, e]$ which holds the state of all teammates θ and the state of the other objects e; e consists of the state of all opponents and the state of the ball. In the case that agent i receives visual information about teammate j, it can derive its current state information θ_j^t consisting of its index number j, global position, global velocity, and body direction. This information is then stored in the world model s_i of the observing agent i. In case a teammate j is not observed, we can use the previously recorded state information in combination with its known policy (either moving to its strategic position or intercepting the ball) to predict its executed action a_j^t. This action is calculated in the same way as the observing agent i would have done should it be in that situation:

$$\hat{a}_j^t = \pi_j(s_i^t) \tag{4}$$

Agent i thus simulates which action $\hat{a}_j^t$ teammate j will perform in the current situation using its own perception of the world s_i.

Consider the most common case where agent i models the behavior of teammate j which moves to its strategic position. The strategic position p_j of the observed teammate j is calculated using its known home position in the formation and the current ball position as observed by agent i. Given the current stored state information θ_j^t of teammate j, agent i can simulate which action teammate j has to perform to move to this position p_j (this will either be a turn command to make sure j will be directed towards p_j or a movement command which will get j closer to p_j).

For all other teammates their executed action is determined in the same way. The vector of all individual actions gives the joint action a^t, which can be used to update the world state using the known world dynamics of the simulator

$$p(s_i^{t+1} | s_i^t, a^t) \tag{5}$$

MMTB is not only used to model the behavior of the teammates when they are not observed. It is also applied to determine an action that involves the future state of a teammate. Consider

[2]Note that the ball position is a very important part of the policy of an agent. It is thus very important that an agent has a good estimate of the ball position. Each agent therefore looks at the part of the field in which the ball is located to make sure that each observation contains information about the ball.

the case where a player wants to pass the ball to a teammate. A suboptimal solution would be to pass the ball directly to the current location of this teammate. However, it will take some time before this teammate observes the change in ball movement due to the fact that the arrival of visual information and the execution of actions occur asynchronously in the soccer simulator. The agent will therefore still assume that it is in a passive situation and continue to move to its strategic position (and thus away from the passing point). After observing the approaching ball the teammate will become active, but now has to turn back to the position it just moved away from in order to intercept the ball. To circumvent this problem, we do not pass the ball to the current location of the teammate, but to its predicted position after a short number of time steps. This position is calculated using the proposed MMTB method. This makes it easier for the teammate to intercept the approaching ball, since the ball will move to the location the teammate is currently heading towards.

5 Results

In this section we present empirical results demonstrating the effectiveness of MMTB. We compare our model with the previous used approach in which the state of a teammate was not updated until this teammate was observed again. These approaches were compared using both internal testing (average error in estimate) and external testing (resulting team behavior).

5.1 Internal Testing

In order to test the benefit of our approach, we simulated 10 full-length matches, each lasting 10 minutes. During these matches one specific midfielder recorded two states for each teammate. One state was calculated using MMTB, while the other one was calculated without using MMTB. In each time step the position values from each of these two states was recorded. Afterwards the recorded positions were compared with the perfect position information gathered by the coach. The average distance error per time step with and without MMTB are displayed in Table 1 together with their corresponding standard deviation. This shows that MMTB improves the state information of the teammates stored in the world model and provides us with a better representation with respect to the real world state.

	Avg. error	**St. dev.**
With MMTB	0.53	0.49
Without MMTB	0.77	0.95

Table 1: Average distance error and standard deviation in teammate's position.

5.2 External Testing

To further test the benefit of our approach *UvA Trilearn* with MMTB played matches against *UvA Trilearn* without. The behavior of the players on both teams was otherwise identical. Table 2 shows the results over the course of 10 full-length games.

The results show that MMTB has a positive effect on the performance on the team as a whole. Most games were won and only one game has been lost. Although this shows that MMTB improves the overall quality of the team, it does not show which part of the game actually caused this. In order to get better insight in the actual origin of this improvement, we analyzed the match statistics using the *Statistics Proxy Server*. This is an analysis tool,

	Wins	**Draws**	**Losses**	**Avg. score**	**St. dev.**
With MMTB	7	2	1	2.1	1.05
Without MMTB	1	2	7	0.9	0.99

Table 2: Results of 10 games between *UvA Trilearn* with and without MMTB.

which provides real-time, in-depth statistics about soccer games in the RoboCup simulation league [4]. The only significant difference between the two teams was the result in the passing statistics. The successful passing percentage over these 10 matches was 80.12% for the team with MMTB and 72.56% for the team without. These percentages indicate that due to the better estimate of the state of the teammates, fewer mistakes are made when the ball is passed from one teammate to the other. The benefit of using MMTB was also visible during the *German Open 2002* tournament in which we became champion. The players could pass the ball rapidly between each other, which made it possible to move the ball quickly to another part of the field and outplay the opponent's defense.

6 Related Work

Two other approaches that model the behavior of other agents' future actions are RMM (recursive modeling method) and IMBBOP ("ideal-model-based behavior outcome prediction").

RMM [6] provides a theoretical framework for representing and using the knowledge that an agent has about its expected payoffs and those of others. An agent models the internal state and action selection strategy of the other agents in order to predict its action. The method is recursive since the other agents on their turn will be modeling the first agent, etc. Taken to an extreme, the amount of knowledge an agent need to possess in order to coordinate its interactions with others might outstrip the agent's limited reasoning capacity. Techniques are therefore introduced for keeping agents "ignorant enough" so that they can practically coordinate [2].

The main difference between RMM and our approach is that we assume that the policy of the agents is common knowledge among the teammates and we therefore do not have to recursively model the predicted actions of the other agents actions and their effect on our action selection.

Another approach which models the behavior of other agents' future actions is IMBBOP [10], which models the results of other agents' future actions in relation to their optimal actions based on an ideal world model. A benefit of IMBBOP is that it does not rely on the assumption that the state of the other agents and their actions capabilities are known, since it operates in relation to the ideal capabilities given the world dynamics.

In our case we do have the knowledge about the policy and utilize this knowledge to model the behavior of the other agents. As a result it gives us an accurate representation of the actual state of the agent.

7 Conclusion

In this paper we described MMTB and how it is applied in our RoboCup simulation team *UvA Trilearn* to model the behavior of teammates to improve upon their state information. This model predicts the future actions of a teammate by simulating its policy. This model can be used to update the state of an unobserved teammate or to predict its future state. Results show that MMTB has a positive effect on the average error of the state estimate and proves to be successful for the overall team behavior. Furthermore, MMTB is one of the aspects of our team *UvA Trilearn*, which reached fourth place at the *RoboCup-2001* world cup and became champion at the *German Open 2002*.

An essential assumption in MMTB is that the individual policies of the teammates are common knowledge among all teammates. It is therefore not possible to model opponent behavior since their policy is unknown. In order to model the behavior of agents of which the policy is initially not known two approaches can be considered. The first method is to try to create a model of the opponent policy based on observations. This is a difficult task, especially in situations where there is not enough interaction to learn such a model. Another approach is to use IMBBOP [10] which models the results of other agents' future actions in relation to their optimal actions based on an ideal world model.

Acknowledgments

This research is supported by PROGRESS, the embedded systems research program of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the Technology Foundation STW.

References

[1] R. de Boer and J. Kok. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team. Master's thesis, University of Amsterdam, The Netherlands, feb 2002.

[2] E. H. Durfee. Practically coordinating. *AI Magazine*, 20(1):99–116, 1999.

[3] E. Foroughi, F. Heintz, S. Kapetanakis, K. Kostiadis, J. Kummeneje, I. Noda, O. Obst, P. Riley, and T. Steffens. RoboCup Soccer Server User Manual: for Soccer Server version 7.06 and later, 2001. At http://sourceforge.net/projects/sserver.

[4] I. Frank, K. Anaka-Ishii, K. Arai, and H. Matsubara. The statistics proxy server. In P. Stone, T. Balch, and G. Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 303–308, Berlin, 2001. Springer Verlag.

[5] J. Geanakoplos. Common knowledge. *Economic Perspectives*, 6(4), 1992.

[6] P. Gmytrasiewicz, E. Durfee, and D. Wehe. A decision-theoretic approach to coordinating multiagent interactions. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 62–68, 1991.

[7] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. RoboCup: The Robot World Cup Initiative. In *Proceedings of the IJCAI-95 Workshop on Entertainment and AI/Alife*, 1995.

[8] L. P. Reis and J. N. Lau. FC Portugal Team Description: RoboCup-2000 Simulation League Champion. In P. Stone, T. Balch, and G. Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 29–40. Springer Verlag, Berlin, 2001.

[9] P. Stone. *Layered Learning in Multi-Agent Systems*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, Dec. 1998.

[10] P. Stone, P. Riley, and M. Veloso. Defining and Using Ideal Teammate and Opponent Agent Models. In *Twelfth Innovative Applications of AI Conference (IAAI-2000)*, 2000.