Defining and Using Ideal Teammate and Opponent Agent Models: A Case Study in Robotic Soccer*

Peter Stone
AT&T Labs — Research
180 Park Ave., room A273
Florham Park, NJ 07932
pstone@research.att.com

Patrick Riley and Manuela Veloso
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{pfr,veloso}@cs.cmu.edu

1 Introduction

A common challenge for agents in multiagent systems is trying to predict what other agents are going to do in the future. Such knowledge can help an agent determine which of its current action options are most likely to help it achieve its goals.

Ideally, an agent could learn a model of other agents' behavior patterns via direct observation of their past actions. However, that is only possible when agents have many repeated interactions with one another.

We explore the use of agent models in an application where extensive interactions with a particular agent are not possible, namely robotic soccer. In robotic soccer tournaments, such as RoboCup [1], a team of agents plays against another team for a single, short (typically 10-minute) period. The opponents' behaviors are usually not observable prior to this game and there are not enough interactions during the game to build a useful model.

In this paper, we introduce "ideal-model-based behavior outcome prediction" (IMBBOP). This technique predicts an agent's future actions in relation to the optimal behavior in its given situation. This optimal behavior is agent-independent and can therefore be computed based solely on a model of the world dynamics. IMBBOP does not assume that the other agent *will* act according to the theoretical optimum, but rather characterizes its expected behavior in terms of deviation from this optimum.

2 The Application: Goal-Scoring in Soccer

Our IMBBOP implementation is carried out in the simulated robotic soccer domain using the RoboCup soccer server [2]. Over the past several years, we have created teams of soccer-playing agents for use in the RoboCup simulator. The teams are all called "CMUnited-XX," where "XX" indicates the year in which they first participated in the RoboCup international simulator tournament.

Although CMUNITED-98 [5], the champion of RoboCup-98, out-scored its opponents by a combined score of 66–0, it failed to score on many opportunities in which it had the ball close to the opponent's goal, especially against the better opponents. Similarly, when playing against itself, there are many shots on goal, but few goals (roughly one by each team every 3 games). Since CMUNITED-98 became publicly available after the 1998 competition, we expected there to be several teams at RoboCup-99 that could beat CMUNITED-98, and indeed there were. In order to improve its performance, we introduced IMBBOP into the CMUNITED-99 team [3], specifically to improve its goal-scoring ability.

IMBBOP is used in several ways in CMUNITED-99. Most significantly in terms of performance, it is used to decide when to shoot and when to pass when an agent has the ball very near to the opponent's goal and when it is on a "breakaway." It is also used by agents to determine when the opponents are likely to be able to steal the ball from them.

3 IMBBOP

IMBBOP is designed for situations in which an agent X has a goal G to be achieved by time T. X must determine whether agent Y can prevent (if an "opponent") or achieve (if a "teammate")[1] G after X takes action A. In particular, X must determine which of its possible actions $A_1, \ldots, A_n$ is most likely to achieve G by time T.

IMBBOP makes the following assumptions:

- X must select an action from among $A_1, \ldots, A_n$ to be executed immediately. It then ceases to affect the achievement of G.

*An extended version of this paper is available [4].

[1]Here we consider an agent to be a teammate if it also has the goal G and to be an opponent if it has the goal of preventing G. We assume that X knows which agents are teammates and which are opponents.

- Whether or not Y can achieve or prevent X's goal depends on T. That is, $\exists t$ s.t. Y could achieve G by, or prevent G from being achieved by, time t.
- X has a model of the world dynamics.
- X has incomplete information regarding Y's current state.
- X has an incomplete model of Y's capabilities (how it can affect the world). That is, X knows (through the world model) what actions Y can take, but has no model of how Y chooses its action. However, based on the world model, X can deduce an upper bound on Y's capabilities in terms of the minimum time necessary to execute tasks. For example, the world model could specify a maximum possible agent speed.

Given these assumptions, IMBBOP works as follows.

1. Using the model of world dynamics and the resultant upper bounds on agent capabilities, determine analytically the minimum t such that Y could prevent or achieve G by time t after X takes action A.

2. Use a threshold on $T - t$ to predict whether or not action A will succeed: the greater $T-t$, the more likely Y is to be able to prevent or achieve G by time T. Thus, $T-t$ is an indication of the likelihood that action A will result in goal G being achieved by time T.

In step 1, such an analysis is made possible under the simplifying assumption that the world dynamics and a time-based bound on the action capabilities of Y are known. In addition, X fills in missing information about Y with best-case values from Y's perspective (i.e., if Y could be in one of n states, X assumes that Y is in the state from which it could most quickly achieve or prevent G). Note that there is no guarantee that Y could *actually* achieve G by time t.

For example, if Y is currently located at location (x_1, y_1) and must get to location (x_2, y_2) in order to prevent G, then, using a theoretical maximum speed of s, X could compute analytically that Y cannot get to location (x_2, y_2) in time less than $\frac{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}{s}$. In actual fact, it may be unlikely that Y could actually arrive at (x_2, y_2) so quickly given the time necessary for it to figure out that it needs to get there and possibly accelerate to the maximum speed.

In practice, X will execute action A based on whether or not $T - t$ exceeds some threshold.

4 Results

IMBBOP has proven to be very useful to us in creating the CMUNITED-99 team of soccer-playing agents [3]. While CMUNITED-98 could rarely score when playing against itself (roughly 1 goal every 3 games), CMUNITED-99 scores about 9 goals per game when playing against CMUNITED-98. There is clear evidence that incorporating IMBBOP into the agents' breakaway strategy is itself enough to lead to a significant improvement in the team's performance.

We played five versions of CMUNITED-99 against the CMUNITED-98 team. The only difference among these 5 versions was that their agents used 5 different breakaway strategies. The three strategies (1–3) using some form of IMBBOP all performed significantly better than the two (4–5) which do not. Note that the the CMUNITED-98 team used breakaway strategy 4.

Each version played 9 10-minute games against CMUNITED-98. Table 1 displays the goals per game scored by each of these versions. CMUNITED-98 never scored a goal.

	Breakaway Strategy				
Goals/Game	1	2	3	4	5
Mean	8.9	10.6	8.6	3.6	3.6
Std. Dev.	± 1.5	± 1.3	± 2.6	± 1.4	± 1.0

Table 1: Goals scored by CMUNITED-99 against CMUNITED-98 when using the different breakaway strategies.

Since the RoboCup tournaments themselves do not provide controlled testing environments, we cannot make any definite conclusions based on the competitions. However, when watching the games during RoboCup-99, we noticed many goals scored as a result of well-timed shots and passes near the opponent's goal. In the end, CMUNITED-99 went on to win the RoboCup-99 championship, outscoring its opponents, many of which were ale to beat CMUNITED-98, by a combined score of 110–0.

References

[1] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada. The RoboCup synthetic agent challenge 97. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 24–29, San Francisco, CA, 1997. Morgan Kaufmann.

[2] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12:233–250, 1998.

[3] P. Stone, P. Riley, and M. Veloso. The CMUnited-99 champion simulator team. In M. Veloso, E. Pagello, and H. Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, Berlin, 2000. Springer Verlag.

[4] P. Stone, P. Riley, and M. Veloso. Defining and using ideal teammate and opponent agent models. In *Proceedings of the Twelfth Annual Conference on Innovative Applications of Artificial Intelligence*, 2000.

[5] P. Stone, M. Veloso, and P. Riley. The CMUnited-98 champion simulator team. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag, Berlin, 1999.