Adaptive Agent Tracking in Real-world Multi-Agent Domains: A Preliminary Report

Milind Tambe, Lewis Johnson and Wei-Min Shen
Information Sciences Institute and Computer Science Department
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
{tambe,johnson,shen}@isi.edu

Abstract

In multi-agent environments, the task of *agent tracking* (i.e., tracking other agents' mental states) increases in difficulty when a tracker (tracking agent) only has an imperfect model of the trackee (tracked agent). Such model imperfections arise in many real-world situations, where a tracker faces resource constraints and imperfect information, and the trackees themselves modify their behaviors dynamically. While such model imperfections are unavoidable, a tracker must nonetheless attempt to be adaptive in its agent tracking. In this paper, we analyze some key issues in adaptive agent tracking, and describe an initial approach based on discrimination-based learning. The main idea is to identify the deficiency of a model based on prediction failures, and revise the model by using features that are critical in discriminating successful and failed episodes. Our preliminary experiments in simulated air-to-air combat environments have shown some promising results but many problems remain open for future research.

Introduction

In multi-agent environments, intelligent agents must interact — collaborate or compete — to achieve their goals. Many of these multi-agent domains are dynamic and real-time, requiring the interaction to be flexible and reactive. For instance, in the arena of training, there is a recent thrust on dynamic, real-time interactive simulations — e.g., realistic traffic environments (Cremer *et al.* 1994), or realistic combat environments(Tambe *et al.* 1995) — where intelligent agents may interact with tens or hundreds of collaborative and non-collaborative participants (agents and humans). In the arena of education, intelligent tutors, whether in the form of "standard" intelligent tutoring systems(Anderson *et al.* 1990) or as participants in virtual environments (e.g., a virtual guide in a historical simulation(Pimentel & Teixeira 1994)), must interact with students in real-time. Similarly, in the arena of entertainment, recent work has focused on real-time, dynamic interactivity among multiple agents within virtual reality environments (Bates, Loyall, & Reilly 1992; Hayes-Roth, Brownston, & Gen 1995). Such real-time interaction is also seen in robotic environments (Kuniyoshi *et al.* 1994).

In all these environments, *agent tracking* is a key capability required for intelligent interaction (Tambe & Rosenbloom 1995; Ward 1991; Rao 1994; Anderson *et al.* 1990). It involves monitoring other agents' observable actions and inferring their unobserved actions or high-level goals, plans and behaviors. This capability is closely related to plan recognition (Kautz & Allen 1986; Song & Cohen 1991), which involves recognizing agents' plans based on observations of their actions. One key difference is that plan-recognition efforts generally assume that agents are executing plans that rigidly prescribe the actions to be performed. Agent tracking, in contrast, involves recognizing a broader mix of goal-driven and reactive behaviors. It is appropriate in domains where agents exhibit dynamic behaviors in response to the changing environment and the actions of other agents.

This paper focuses on *adaptive agent tracking*, an important requirement to scale up tracking to real-world domains. In particular, agent tracking is typically based on *model tracing* (Anderson *et al.* 1990), where a tracker (tracking agent) executes a runnable model of the trackee (tracked agent), matching the model's predictions with actual observations. However, in real-world domains, a tracker's model of the trackee's behaviors is often imperfect, i.e., incomplete or incorrect. The trackee further contributes to this imperfections, since its behaviors are not static, but alter over time. Of course, eradicating all such model imperfections is difficult (if not impossible) — and thus tracking must proceed despite such imperfections. Nonetheless, the tracker must engage in some adaptive tracking, i.e., it must adapt its model of the trackee to remdey at least some of these imperfections.

In this preliminary report, we analyze some of the key challenges in adaptive agent tracking, and present an approach (based on discrimination-based learning) to address them. Our analysis is based on a real-world multi-agent domain, of combat simulations. An initial implementation of our approach in this domain has shown some promising results; although many issues remain open for future research.

Agent Tracking in Real-world Domains

The domain of our work on agent tracking is one of virtual battlefields based on Distributed Interactive Simulation Environments (DIS). These are synthetic, yet real-world environments, and they have already been used in large-scale operational military exercises. These environments promise to provide cost-effective and realistic environments for training and rehearsal, as well as for testing new doctrine, tactics and weapon system concepts. The realization of this promise is critically dependent on intelligent automated agents that can act as effective human surrogates — interacting intelligently with humans as well as other agents. Agent tracking is of course one key aspect of such an intelligent interaction (Tambe & Rosenbloom 1995). Certainly, an adversary will not communicate information regarding its goals and plans to an agent voluntarily — such information must be inferred via tracking. Furthermore, even in collaborative situations, tracking often assumes importance due to communication difficulties.

Given that agent tracking occurs here on a synthetic battlefield, there are some key challenges that it poses:

- *Imperfect models:* A tracker's model of the trackee is often imperfect. Such imperfection could be divided into two categories:
  1. Dynamic model imperfections: The tracker possesses a perfect static model of the trackee — so that the trackee's set of possible goals, plans, and beliefs is known. However, the tracker has to infer the trackee's currently active goals, plans and beliefs. This task is, of course, at the heart of agent tracking.
  2. Static model imperfections: The tracker's static model of the trackee is itself incomplete, i.e., the trackee's overall set of possible goals or plans or beliefs is itself not known. This situation may arise due to *adaptiveness*: an intelligent adversary will very likely adapt its tactics to exploit possible weaknesses in a tracker's behaviors.
- *Real-time and dynamism:* The tracker and trackee interact in real-time; for example, in simulated air-to-air combat, *speed is life* (Shaw 1988) in the real-world, and in the simulated combat environment.
- *Complexity of environment:* This is a realistic environment, in which entities and objects have a rich set of properties.
- *Cost of trial:* Trials are not straightforward to run. Indeed, trials with same initial conditions may have very different outcomes due the "chaotic" nature of the environment.

While our analysis focuses on the combat-simulation environment, given its real-world character, we expect that the lessons will generalize to some of the other multi-agent environments mentioned above.

Issues in Adaptive Agent Tracking

As mentioned earlier, to track a trackee, tracker executes a runnable model of the trackee, matching the model's predictions with actual observations. One key reason that tracking in this fashion remains a challenging problem is dynamic model imperfections. Dynamic imperfections introduce ambiguity in tracking. For instance, in air-combat simulations, a tracker cannot directly observe a missile fired by its opponent (the trackee). It needs to infer a missile firing from the trackee's observable maneuvers, even though those are often ambiguous. Nonetheless, given a reasonably accurate model of the trackee, the tracker agent can hope to address such ambiguity in real-time (Tambe & Rosenbloom 1995).

Given adaptiveness on part of the trackee (static model imperfections), however, the situation becomes much more complex. The tracker cannot necessarily assume its model of the trackee is accurate, or that it will stay accurate over time. Such a situation does arise in the synthetic battlefield environment, given the adaptive character of intelligent adversaries. In particular, human adversaries will very likely adapt and evolve their tactics to exploit weaknesses in an intelligent agent's behaviors. For instance, in the simulated theater of war exercise (STOW-E) held in November of 1994, human pilots deliberately changed their missile firing tactic — instead of pointing their aircraft nose straight at the target before firing a missile (0 - 5 degrees "nose-off" as shown in Figure 1), they began firing missiles while maintaining a 25-degree nose-off from the target (as shown in Figure 2). This was intended to confuse the participating intelligent pilot agents, and indeed it did (Tambe *et al.* 1995). Unable to track this changed missile firing tactic, intelligent pilot agents got shot down. Of course, human pilots are bound to come up with novel variations on known maneuvers, and intelligent agents cannot be expected to anticipate them. Yet, at the same time, intelligent agents cannot remain in a state of permanent vulnerability — for instance, getting shot down each time the 25-degree nose-off variation gets used — otherwise they would be unable to continue to provide a challenging and appropriate training environment for human pilots.

To deal with such imperfections in the trackee's model, the tracker must:

- Recognize the deficiency of its model;
- Adapt the model by either revising its assumptions regarding known agent actions or postulating the existence of heretofore unknown actions.

Unfortunately, characteristics of the combat simulation environment outlined in the previous section conspire to make this difficult. Thus, given imperfect in-

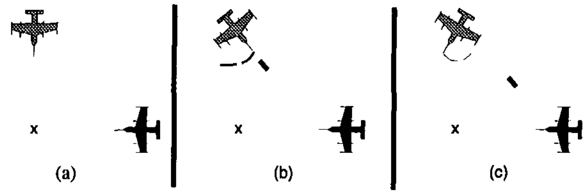

Figure 1: A simulated air-combat scenario illustrating the "normal" missile firing maneuvers: (a) aircraft approach each other; (b) the trackee points straight at the tracker to fire a missile; (c) the trackee executes an Fpole turn to continue supporting the missile without flying right behind the missile. An arc on an aircraft's nose shows its turn direction.

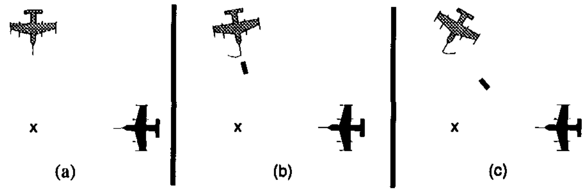

Figure 2: A simulated air-combat scenario illustrating a change in the missile firing maneuver.

formation, it is difficult for the tracker to pinpoint the deficiencies in its model of the trackee. For instance, in the above STOW-E example, the tracker failed to recognize that 25-degree "nose off" is also a legitimate condition for missile firing (for its model is that missile firing is possible only if the nose-off angle is within 0-5 degrees). Yet, since it cannot observe the missile, and thus cannot know when it was fired, the tracker cannot easily pinpoint this precise deficiency. All it does detect is that it was unexpectedly shot down. (One simplifying assumption here is that the tracker assumes that it is the trackee's missile that shot it down).

The complexity of the environment and its real-time nature further complicate matters. Basically, it is difficult for the tracker to obtain a perfectly accurate model of the trackee, and access all of the relevant information for accurate tracking. For instance, in some situations, to predict whether an opponent pilot (trackee) will engage in an offensive or defensive maneuver, it is useful to know whether or not the trackee is willing to enter into risky situations in pursuit of its goals. However, it is difficult obtain such information in advance or during a real-time air-combat simulation — in fact, the tracker may end up jeopardizing its own survival in such a test.

Therefore, we believe that attempting to learn an exact model of the trackee will not be a very fruitful enterprise for the tracker. Instead, it should focus its learning effort on situations involving catastrophic failures, such as the unexpected missile firing in the STOW-E example. With respect to other less harmful imperfections, using a flexible tracking strategy — one that can work with an imperfect model of the trackee — would appear to be a more fruitful approach. Such a strategy would need the capability to switch inferences dynamically in real-time (Tambe & Rosenbloom 1995). For instance, if the tracker is not sure if the trackee is performing an offensive maneuver or a defensive maneuver, it may first assume that the maneuver is offensive (worst-case scenario), and then flexibly modify this assumption as soon as warranted by further observations. Thus, the tracker need not depend on acquiring information regarding risk.

Our Current Approach to Adaptive Agent Tracking

We are currently investigating an approach to adaptive agent tracking that involves learning prediction rules based on discrimination (Shen 1993; 1994). The discrimination-based approach is used to locate deficiencies in tracker's model of the trackee. This approach is augmented using discovery learning techniques for explaining successful and unsuccessful agent tracking experiences (Johnson 1994), to further specialize the analysis of such deficiencies.

The main idea of discrimination-based learning is the framework of predict-surprise-revise. The agent always makes predictions that can be verified or falsified by the actual observations, and if a prediction fails in the current situation, then the current episode will be compared with an earlier episode where the prediction was successful. The agent compares the features of the two situations looking for differences in the features, and incorporates those features into a revised prediction rule. In the event that no known features are effective in discriminating the situations, it may be necessary to hypothesize that new unknown features exist.

In order to apply discrimination-based learning to the agent tracking problem, three issues must be resolved:

- The tracker must decide whether a given episode is worth analyzing — this may be because the trackee performed a particularly interesting action, or because the trackee's action had particularly dire consequences.
- The tracker must decide which episodes to compare, and which points of time to compare between episodes. Each episode consists not of a single situation, but of a sequence of situations, each of which may be different from the previous one. It is impractical to discriminate all of the situations that arise in each episode; rather, the tracker must select specific situations in each episode which are likely to yield meaningful differences when compared. Then, once a situation is selected within one episode, it is necessary to determine which situation corresponds to

it in the other episode, so that an effective discrimination can be performed.

- The tracker must determine which features to discriminate, and decide whether the existing features are effective in discriminating the situations.

Our tracker agents record important events during each episode, such as an air-combat engagement, in their memory. They also keep track of changes in the environmental context, or their own mental state, as these events occur. This enables them to perform an after-action review of the engagement, both to determine whether the engagement contains significant events that should be analyzed, and to analyze those events and the situations in which they occur in order to learn form them.

In the example under discussion (Figure 2), after-action review reveals two reasons why the engagement is worth analyzing. First, it had a catastrophic outcome—the tracker was unexpectedly shot down. Second, the tracker had serious difficulties tracking the opponent's actions at the end of the engagement. This suggests that the opponent may have performed some unexpected action which enabled it to win the fight. If the tracker is able to improve its tracking ability, it might be able to anticipate such actions in the future and plan against them.

We are evaluating two ways of deciding which events to focus on in the episode for analysis. One is to apply domain knowledge to reason backwards from the undesired event (being hit by a missile) to its causal antecedent (the opponent firing a missile). The other is to examine the event trace, and scan backwards from the point of catastrophe to the first point where a prediction failed (the failure of the opponent to turn its nose toward the agent). The tracker last predicts the opponent to move to a nose-off of 0-5 degrees, to fire a missile. This never occurs, and remains the last point of prediction/tracking failure. In this example the approaches to selecting events for analysis can lead to similar learning outcomes: either relaxing the conditions under which the tracker determines that the opponent is on target for a missile shot, or relaxing the conditions under which the opponent actually takes the shot.

The decision of what instances to compare between engagements is carried out as follows. The tracker compares the instances under study, where an opponent action was expected to occur and did not, with previous engagements where the expected action did occur. In the current implementation, the comparison is indirect; the agent first analyzes events where the expected action did occur, learns rules (chunks) that identify key features in the situations in which those events occurred, and tries to apply those rules to the the situation in which the prediction failed. This analysis is performed by the Debrief explanation system. As described in (Johnson 1994), when a user requests Debrief to explain a decision, it analyzes the situational factors leading to the decision, and builds chunks that recognize those factors in future decisions. Figure 3 shows a snapshot of the Debrief user interface listing some of the conclusions drawn by the tracker, primarily about the opponent's actions, which may be selected and explained.

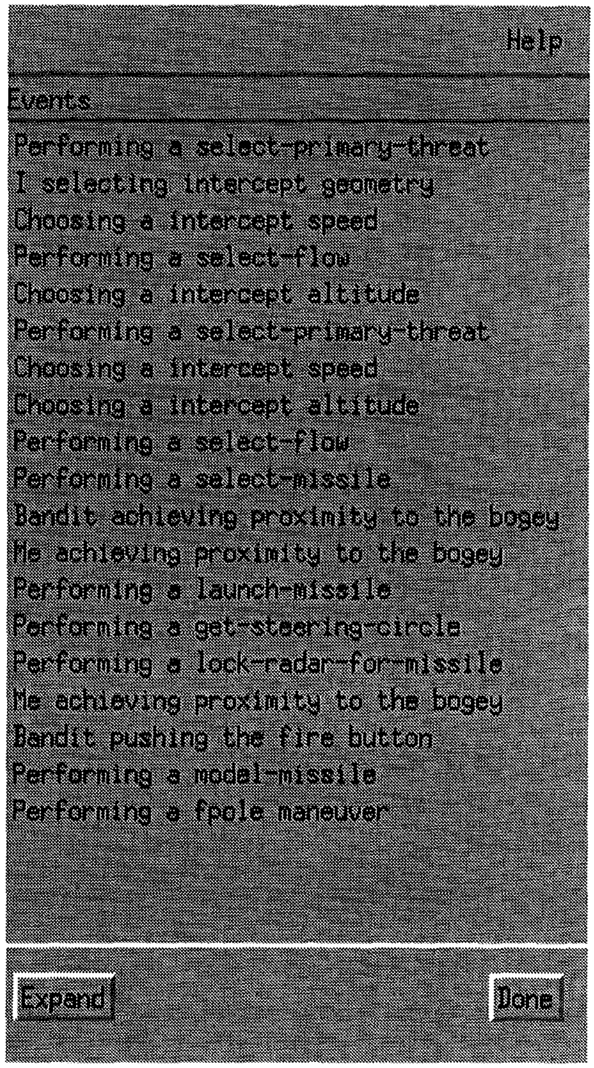

Figure 3: Debrief's user interface listing some of the conclusions drawn by the tracker, primarily about the opponent's actions.

Unfortunately, in complex domains such as battlefield simulations, discrimination-based learning can be difficult to apply without knowledge and/or heuristics for deciding which features to attend to. A feature description of the environment can be quite complex: our agent models typically have hundreds of features, many of which change constantly. Thus, comparing the feature set where the missile firing was successfully tracked with one where the tracking failed may yield differences in a large number of irrelevant features, e.g., the altitude or speed of the trackee.

Under these circumstances, techniques akin to *incremental enlargement* (Shen 1993) or learning via experimentation (Gil 1993) can be employed to determine which features to discriminate.

Incremental enlargement is a heuristic applicable to finding relevant features that are crucial in discriminating two environmental states that contain a large number of differences. The idea is to first focus on the features that are mentioned in the action or operator involved in the prediction failure. If difference can be found in this core set of features, then the search for difference stops. Otherwise, this core set will be enlarged to include other features that are related to some core features and the search for difference continues. This process of enlargement stops as soon as differences are found among the two states. To illustrate the idea, consider the STOW-E example again. The feature "nose-off" is mentioned in the operator "steering-circle-achieved" which is involved in the current prediction failure. Since this feature has different values in the two states, it is identified as the cricial feature that can discriminate the two states, although in these two particular environmental states there are many other features that have different values as well. The initial core set of relevant features is provided by the prior analysis by Debrief of successful tracking episodes.

Experimental Results

We are taking a two-pronged approach in implementing the above approach. First, we have begun implementing the approach in intelligent pilot agents for the air-combat simulation environment (Tambe *et al.* 1995). To this end, two versions of the pilot agents — one developed for tracking (Tambe & Rosenbloom 1995; Tambe 1995) and one that learns models of decisions for explanation (Johnson 1994) — have been integrated. (The user interface in Figure 3 is from this integrated version.) We are experimenting with this integrated agent to address the 25-degree nose-off example discussed above. (These agents are based on the Soar integrated architecture (Newell 1990), and use chunking, a form of EBL, as the basis of all of their learning (Laird, Rosenbloom, & Newell 1986)).

We are also in the process of implementing the above approach in a simple test-bed, where agents mimic the behavior of fighter aircraft, and are provided limited information about other agents. The goal here is to engage in controlled experiments, and gain some understanding of the tradeoffs involved. We have begun with two sets of experiments: one is to give the tracker a good model of the opponent and see how it can deal with imperfect information and make flexible and safe predictions, and the other to have the tracker learn the model from scratch.

Conclusions and Future Research

This paper analyzed the problem of adaptive agent tracking, where the tracker possesses an imperfect model of the trackee. This problem is especially important in real-world situations where a tracker faces resource constraints and imperfect information, and a trackee may itself modify its behavior dynamically. We have proposed an approach — based on discrimination-based learning — to address this problem; with some initial promising results. Furthermore, this work has revealed several important future research directions for adaptive agent tracking. In particular, we plan to investigate: (i) the criteria that an agent may use in selecting "interesting" episodes for analysis; (ii) mechanisms for maintaining episodic memory, and comparison across episodes; and (iii) feature discrimination, particularly via incremental enlargement.

Acknowledgments

This research was supported under subcontract to the University of Southern California Information Sciences Institute from the University of Michigan, as part of contract N00014-92-K-2015 from the Advanced Systems Technology Office (ASTO) of the Advanced Research Projects Agency (ARPA) and the Naval Research Laboratory (NRL); and under contract N66001-95-C-6013 from the Advanced Systems Technology Office (ASTO) of the Advanced Research Projects Agency (ARPA) and the Naval Command and Ocean Surveillance Center, RDT&E division (NRAD). Critical expertise and support has been provided by BMH Inc. The third author (Wei-Min Shen) was supported in part under Contract Number F30602-94-C-0210 (Rome Laboratory of the Air Force Systems Command and the Advanced Research Projects Agency) and in part under Contract Number MDA972-94-2-0010 (Technology Reinvestment Program award funded by the Advanced Research Projects Agency) and by the State of California under Contract Number C94-0031.

References

Anderson, J. R.; Boyle, C. F.; Corbett, A. T.; and Lewis, M. W. 1990. Cognitive modeling and intelligent tutoring. *Artificial Intelligence* 42:7–49.

Bates, J.; Loyall, A. B.; and Reilly, W. S. 1992. Integrating reactivity, goals and emotions in a broad agent. Technical Report CMU-CS-92-142, School of Computer Science, Carnegie Mellon University.

Cremer, J.; Kearney, J.; Papelis, Y.; and Romano, R. 1994. The software architecture for scenario control in the Iowa driving simulator. In *Proceedings of the Conference on Computer Generated Forces and Behavioral Representation.*

Gil, Y. 1993. Efficient domain-independent experimentation. Technical Report ISI/RR-93-337, USC / Information Sciences Institute. Appears in the Proceedings of the Tenth International Conference on Machine Learning.

Hayes-Roth, B.; Brownston, L.; and Gen, R. V. 1995. Multiagent collaobration in directed improvisation. In *Proceedings of the International Conference on Multi-Agent Systems (ICMAS-95).*

Johnson, W. 1994. Agents that learn to explain themselves. In *Proceedings of the National Conference on Artificial Intelligence*, 1257–1263. Seattle, WA: AAAI.

Kautz, A., and Allen, J. F. 1986. Generalized plan recognition. In *Proceedings of the National Conference on Artificial Intelligence*, 32–37. Menlo Park, Calif.: AAAI press.

Kuniyoshi, Y.; Rougeaux, S.; Ishii, M.; Kita, N.; Sakane, S.; and Kakikura, M. 1994. Cooperation by observation: the framework and the basic task pattern. In *Proceedings of the IEEE International Conference on Robotics and Automation.*

Laird, J. E.; Rosenbloom, P. S.; and Newell, A. 1986. Chunking in soar: The anatomy of a general learning mechanism. *Machine Learning* 1(1):11–46.

Newell, A. 1990. *Unified Theories of Cognition.* Cambridge, Mass.: Harvard Univ. Press.

Pimentel, K., and Teixeira, K. 1994. *Virtual reality: Through the new looking glass.* Blue Ridge Summit, PA: Windcrest/McGraw-Hill.

Rao, A. S. 1994. Means-end plan recognition: Towards a theory of reactive recognition. In *Proceedings of the International Conference on Knowledge Representation and Reasoning (KR-94).*

Shaw, R. L. 1988. *Fighter combat: tactics and maneuvers.* Annapolis, Maryland: Naval Institute Press.

Shen, W. 1993. Discovery as autonomous learning from the environment. *Machine Learning* 12:143–165.

Shen, W. 1994. *Autonomous Learning from the Environment.* W. H. Freeman, Computer Science Press.

Song, F., and Cohen, R. 1991. Temporal reasoning during plan recognition. In *Proceedings of the National Conference on Artificial Intelligence.* Menlo Park, Calif.: AAAI press.

Tambe, M., and Rosenbloom, P. S. 1995. RESC: An approach for real-time, dynamic agent tracking. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

Tambe, M.; Johnson, W. L.; Jones, R.; Koss, F.; Laird, J. E.; Rosenbloom, P. S.; and Schwamb, K. 1995. Intelligent agents for interactive simulation environments. *AI Magazine* 16(1).

Tambe, M. 1995. Recursive agent and agent-group tracking in a real-time dynamic environment. In *Proceedings of the International Conference on Multi-agent systems (ICMAS).*

Ward, B. 1991. *ET-Soar: Toward an ITS for Theory-Based Representations.* Ph.D. Dissertation, School of Computer Science, Carnegie Mellon Univ.