

# Case-Based Behavior Recognition in Beyond Visual Range Air Combat

Hayley Borck<sup>1</sup>, Justin Karneeb<sup>1</sup>, Ron Alford<sup>2</sup> & David W. Aha<sup>3</sup>

<sup>1</sup>Knexus Research Corporation; Springfield, VA; USA

<sup>2</sup>ASEE Postdoctoral Fellow; Naval Research Laboratory (Code 5514); Washington, DC; USA

<sup>3</sup>Navy Center for Applied Research in Artificial Intelligence; Naval Research Laboratory (Code 5514); Washington, DC; USA

{first.last}@knexusresearch.com | {ronald.alford.ctr, david.aha}@nrl.navy.mil

## Abstract

An unmanned air vehicle (UAV) can operate as a capable team member in mixed human-robot teams if it is controlled by an agent that can intelligently plan. However, planning effectively in a beyond-visual-range air combat scenario requires understanding the behaviors of hostile agents, which is challenging in partially observable environments such as the one we study. In particular, unobserved hostile behaviors in our domain may alter the world state. To effectively counter hostile behaviors, they need to be recognized and predicted. We present a Case-Based Behavior Recognition (CBBR) algorithm that annotates an agent's behaviors using a discrete feature set derived from a continuous spatio-temporal world state. These behaviors are then given as input to an air combat simulation, along with the UAV's plan, to predict hostile actions and estimate the effectiveness of the given plan. We describe an implementation and evaluation of our CBBR algorithm in the context of a goal reasoning agent designed to control a UAV and report an empirical study that shows CBBR outperforms a baseline algorithm. Our study also indicates that using features which model an agent's prior behaviors can increase behavior recognition accuracy.

## 1. Introduction

We are studying the use of intelligent agents for controlling an unmanned air vehicle (UAV) in a team of piloted and unmanned aircraft in simulated beyond-visual-range (BVR) air combat scenarios. In our work, a *wingman* is a UAV that is given a mission to complete and may optionally also receive orders from a human pilot. In the situations where the UAV's agent does not receive explicit orders, it must create a plan for itself. Although UAVs can perform well in these scenarios (Nielsen et al. 2006), planning may be ineffective if the behaviors of the other agents operating in the scenario are unknown. To

effectively account for hostile and allied agents we use a Case-Based Behavior Recognition (CBBR) algorithm to recognize their behaviors so that, in combination with a predictive planner, UAV plans can be evaluated in real time.

We define a *behavior* as tendency or policy of the agent over a given amount of time. A behavior is comprised of a set of unordered actions (e.g., 'fly to target', 'fire missile') taken in relation to other agents in the scenario. This differs from a plan in that the agent is not following a set of ordered actions; it is rather taking actions that are indicative of certain behaviors.

BVR air combat involves executing precise tactics at large distances where little data relative is available of the hostiles. What is available is only partially observable. Yet if the UAV can identify a hostile agent's behavior or plan it can use that information when reasoning about its own actions.

We hypothesize that Case-Based Reasoning (CBR) techniques can effectively recognize behavior in domains such as ours, where information on hostile agents is scarce. Additionally, we hypothesize that representing and leveraging a memory of a hostile agent's behaviors during CBR will improve behavior recognition. To assess this, we encode discrete state information over time in cases, and compare the performance of CBBR using this information versus an ablation that lacks this memory.

We summarize related work in Section 2 and describe our CBBR algorithm in Section 3. In Section 4, we describe its application in 2 vs 2 scenarios (i.e., two 'friendly' aircraft versus two 'hostile' aircraft), where we found that (1) CBBR outperformed baseline algorithms and (2) using features that model the past behavior of other agents increases recognition accuracy. Finally, we discuss the implications of our results and conclude with future work directions in Section 5.

## 2. Related Research

Our behavior recognition component, which lies within a larger goal reasoning (GR) agent called the Tactical Battle Manager (TBM), is designed to help determine if a UAV wingman's plan is effective (Borck et al. 2014). In this paper we extend the CBBR algorithm's recognition abilities by (1) employing a confidence factor  $C_q$  for a query  $q$  and (2) using features to model other agents' past behaviors. We also present the first empirical evaluation of our CBBR system; we test it against a baseline algorithm and assess our hypothesis that these new features improve CBBR performance.

In recent years, CBR has been used in several GR agents. For example, Weber et al. (2010) use a case base to formulate new goals for an agent, and Jaidee et al. (2013) use CBR techniques for goal selection and reinforcement learning (RL) for goal-specific policy selection. In contrast, our system uses CBR to recognize the behavior of other agents, so that we can predict their responses to our agent's actions.

CBR researchers have investigated methods for combating adversaries in other types of real-time simulations. For example, Aha et al. (2005) employ a case base to select a sub-plan for an agent at each state, where cases record the performance of tested sub-plans. Auslander et al. (2008) instead use case-based RL to overcome slow learning, where cases are action policies indexed by current game state features. Unlike our work, neither approach performs opponent behavior recognition.

Opponent agents can be recognized as a team or as a single agent. Team composition can be dynamic (Sukthankar and Sycara 2006), resulting in a more complex version of the plan recognition problem (Lavieris et al. 2009; Sukthankar and Sycara 2011). Kabanza et al. (2014) use a plan library to recognize opponent behavior in real-time strategy games. By recognizing the plan the team is enacting they are able to recognize the opponent's, or team leader's, intent. Similarly our algorithm uses a case base of previously observed behaviors to recognize the current opponents' behaviors. Our approach however attempts to recognize each agent in the opposing team as a separate entity. Another approach to coordinating team behaviors involves setting multiagent planning parameters (Auslander et al. 2014), which can then be given to a plan generator. Recognizing high-level behaviors, which is our focus, should also help to recognize team behaviors. For example, two hostile agents categorized as '*All Out Aggressive*' by our system could execute a pincer maneuver (in which two agents attack both flanks of an opponent).

Some researchers describe approaches for selecting behaviors in air combat simulations. For example, Rao and Murray (1994) describe a formalism for plan recognition

that stores the mental states of adversarial agents in air combat scenarios (i.e., representing their beliefs, desires, and intentions), but did not evaluate it. Smith et al. (2000) use a genetic algorithm (GA) to learn effective tactics for their agents in a two-sided experiment, but assumed perfect observability and focused on visual-range air combat. In contrast, we assume partial observability and focus on BVR air combat, and are not aware of prior work by other groups on this task that use CBR techniques.

## 3. Case-Based Behavior Recognition

The following subsections describe our CBBR algorithm. In particular, we describe its operating context, our case representation, its retrieval function, and details on how cases are pruned during and after case library acquisition.

### 3.1 CBBR in a BVR Air Combat Context

Our CBBR implementation serves as a component in the TBM, a system we are collaboratively developing for pilot-UAV interaction and autonomous UAV control. The CBBR component takes as input an incomplete world state and outputs behaviors that are used to predict the effectiveness of a UAV's plan. The TBM maintains a world model that contains each known agent's capabilities, past observed states, currently recognized behaviors, and predicted future states. A complete state contains, for each time step in the simulation, the position and actions for each known agent. The set of actions that our agent can infer from the information available are *Pursuit* (an agent flies directly at another agent), *Drag* (an agent tries to kinematically avoid a missile by flying away from it), and *Crank* (an agent flies at the maximum offset but tries to keep its target in radar). For the UAV and its allies the past states are complete. However, any hostile agent's position for a given time is known only if the hostile agent appears on the UAV's radar. Also, a hostile agent's actions are never known and must be inferred from the potentially incomplete prior states. We infer these actions by discretizing the position and heading of each agent into the features of our cases. We currently assume that the capabilities of each hostile aircraft are known, though in future work they will be inferred through observations.

The CBBR component revises the world model with recognized behaviors. Afterward, we use an instance of the Analytic Framework for Simulation, Integration, & Modeling (AFSIM), a mature air combat simulation that is used by the USAF (and defense organizations in several other countries), to simulate the execution of the plans for the UAV and the other agents in a scenario. AFSIM projects all the agents' recognized behaviors to determine the effectiveness of the UAV's plan. Thus, the accuracy of these predictions depends on the ability of the CBBR

component to accurately recognize and update the behaviors of the other agents in the world model.

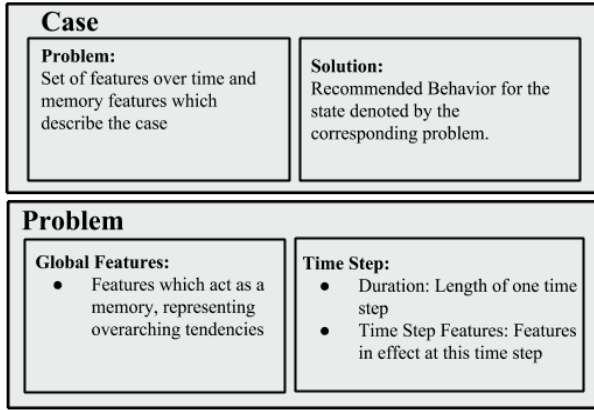


Figure 1: Case Representation

### 3.2 Case Representation

A case in our system describes an agent’s behavior and its response to a given situation including a memory and tendencies. Cases are represented as  $\langle$ problem, solution $\rangle$  pairs, where the problem is represented by a set of discrete features and the solution is the agent’s response behavior. The feature set contains two feature types: global features and time step features (Figure 1). Global features act as a memory and represent overarching tendencies about how the agent has acted in the past. Time step features represent features that affect the agent for the duration of the time step. To keep the cases lean, we merge time steps that have the same features and sum their durations. The features we model are listed below.

#### Time Step Features:

- CLOSINGONOPPOSINGTEAM – This agent is closing on an opposing agent.
- FACINGOPPOSINGTEAM – This agent is facing an opposing agent.
- INOPPOSINGTEAMSRADAR – This agent is in an opposing agent’s radar cone.
- INOPPOSINGTEAMSWEAPONRANGE – This agent is in an opposing agent’s weapon range.
- INDANGER – This agent is in an opposing agent’s radar cone and in missile range.
- 

#### Global Features:

- HASSEENOPPOSINGTEAM – This agent has observed an opposing agent.
- HASAGGRESSIVETENDENCIES – This agent has behaved aggressively.
- HASSELFPRESERVATIONTENDENCIES – This agent has avoided an opposing agent.
- HASDISENGAGED – This agent has reacted to an opposing agent by disengaging.

- HASINTERESTINOPPOSINGTEAM – This agent has reacted to an opposing agent.

Features have Boolean or percentage values. For example, the HASSEENOPPOSINGTEAM global feature is ‘true’ if the UAV observes an opposing team member in its radar. Conversely FACINGOPPOSINGTEAM is a value that represents how much an opposing agent is facing this agent, calculated via relative bearing. This value is averaged over all opposing agents in the agents’ radar which are also facing it. In the current system the Global features are all Boolean while the Time Step features are have percentage values. The set of behaviors include *All Out Aggressive* (actions that imply a desire to destroy opposing team members regardless of its own safety), *Safety Aggressive* (the same as *All Out Aggressive* except that it also considers its own safety), *Passive* (actions that do not engage with opposing team members but keep them in radar range), and *Oblivious* (actions that imply the agent does not know the hostile team exists).

Table 1: Feature Specific Weights

Global Feature	Weight	Time Step Feature	Weight
Seen Opposing	0.1	Closing on Hostiles	0.1
Aggressive Tendencies	0.3	Is Facing Hostiles	0.3
Preservation Tendencies	0.2	In Radar Range	0.1
Has Disengaged	0.2	In Weapon Range	0.2
Interest in Opposing Team	0.2	In Danger	0.3

### 3.3 Case Retrieval

To calculate the similarity between a query  $q$  and a case  $c$ ’s problem descriptions, we compute a weighted average from the sum of the distances between their matching global and time step features. Equation 1 displays the function for computing similarity, where  $\sigma(w_f, q_f, c_f)$  is the weighted distance between two values for feature  $f$ ,  $N$  is the set of time step features, and  $M$  is the set of global features.

$$\text{sim}(q, c) = -\alpha \frac{\sum_{f \in N} \sigma(w_f, q_f, c_f)}{|N|} - \beta \frac{\sum_{f \in M} \sigma(w_f, q_f, c_f)}{|M|} \quad (1)$$

We use a weight of  $\alpha$  for time step features and  $\beta$  for global features, where  $\alpha \geq 0, \beta \geq 0$ , and  $\alpha + \beta = 1$ . We also weighted individual features based on intuition and on feedback from initial experiments as shown in Table 1. These feature weights remain static throughout the experiments in this paper.

In addition we calculate a confidence factor  $C_q$  for each query  $q$ . For a query  $q$  all cases greater than a similarity threshold  $\tau_s$  are retrieved from the case base  $\mathcal{L}$ .  $C_q$  is then computed as the percentage of the retrieved cases whose

solution is the same as that of the most similar case  $c_1$  ( $c_1.s$ ).

$$C_q = \frac{|\{c \in \mathcal{L} \mid \text{sim}(q,c) > \tau_s \wedge c.s = c_1.s\}|}{|\{c \in \mathcal{L} \mid \text{sim}(q,c) > \tau_s\}|} \quad (2)$$

If no cases are retrieved or the  $C_q$  of the most similar case is below a confidence threshold  $\tau_c$ , then the solution is labeled as unknown. We set  $\tau_c$  low (see Section 4.1) so that a solution is returned even when the CBBR is not confident. The confidence factor is used only in the retrieval process in the current CBBR system. In future work, we will extend it to reason over the confidence factor.

### 3.4 Case Acquisition

Cases are acquired by running a BVR simulation with CBBR in acquisition mode. The simulation is run using the same pool of experimental scenarios as the empirical study but with different random trials (see Section 4.1). During a run, the acquisition system receives perfect state information as well as each agent's actual behavior. Unfortunately these are not necessarily accurate for the duration of the simulation. For example most behaviors are similar to *All Out Aggressive* until the agent performs an identifying action that differentiates it (i.e., a drag). This is a limitation of the current acquisition system as it can create cases with incorrectly labeled behaviors. We address this issue during case pruning in Section 3.5.

### 3.5 Case Pruning

To constrain the size of the case base, and perform case base maintenance, cases are pruned from  $\mathcal{L}$  after all cases are constructed (Smyth and Keane 1995). We prune  $\mathcal{L}$  by removing (1) pairs of cases that have similar problems but distinct solutions and (2) redundant cases. Similarity is computed using Equation 1. If the problem  $c.p$  of a case  $c \in \mathcal{L}$  is used as query  $q$ , and any cases  $c' \in \mathcal{C}' \subseteq \mathcal{L}$  are retrieved such that  $c.s = c'.s$  and  $\text{sim}(c.p, c'.p) > \tau_b$  (for a similarity threshold  $\tau_b$ ), then a representative case is randomly selected and retained from  $\mathcal{C}'$  and the rest are removed from  $\mathcal{L}$ . In the future we plan to retain the most common case instead of a random case from  $\mathcal{C}'$ . If instead any case  $c' \in \mathcal{L}$  has a different solution than  $c.s$  (i.e.,  $c.s \neq c'.s$ ), then both cases  $c'$  and  $c$  are removed, if  $\text{sim}(c.p, c'.p) > \tau_d$ .

As discussed in Section 3.4 the pruning algorithm must take into account the limitations of the acquisition system, which generally results in *All Out Aggressive* cases being mislabeled. When pruning cases with different solutions we check a final threshold  $\tau_a$ . If the similarity of a retrieved case  $c'$  is such that  $\text{sim}(c.p, c'.p) > \tau_a$  and the

solution of one of the cases is *All Out Aggressive*, then that case is retained in  $\mathcal{L}$  while the other is pruned.

## 4. Empirical Study

### 4.1 Experimental Design

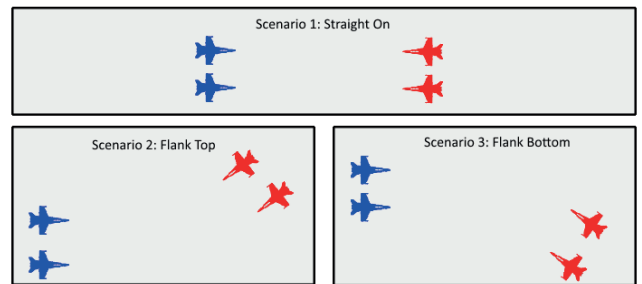
Our design focuses on two hypotheses:

- H1:** CBBR's recognition accuracy will exceed those of the baseline.
- H2:** CBBR's recognition accuracy is higher than an ablation that does not use global features.

To test our hypotheses we compared algorithms using *recognition accuracy*. Recognition accuracy is the fraction of time the algorithm recognized the correct behavior during the *fair duration*, which is the period of time in which it is possible to differentiate two behaviors. We use fair duration rather than total duration because it is not always possible to recognize an agent's behavior before it completes an action. One example of this is that *Safety Aggressive* acts exactly like *All Out Aggressive* until the agent performs a drag action. Thus, in this example, we should not assess performance until the observed agent has performed a drag action. To calculate the *fair duration* defining actions were identified for each behavior. Defining actions were then logged during each trial when an agent performed that action.

We tested the CBBR component with eleven values for  $\alpha$  and  $\beta$  that sum to 1, and pruned the case base for each CBBR variant using their corresponding weights. We compare against the baseline behavior recognizer Random, which randomly selects a behavior every 60 seconds of simulation time.

We ran our experiments with 10 randomized test trials drawn from each of the 3 base scenarios shown in Figure 2. In these scenarios one of the blue agents is the UAV while the other is running one of the behaviors. These scenarios reflect different tactics described by subject matter experts, and are representative of real-world BVR scenarios. In Scenario 1 the hostiles and friendlies fly directly at each other. In Scenarios 2 and 3 the hostiles perform an offset



**Figure 2:** Prototypes for the Empirical Study's Scenarios

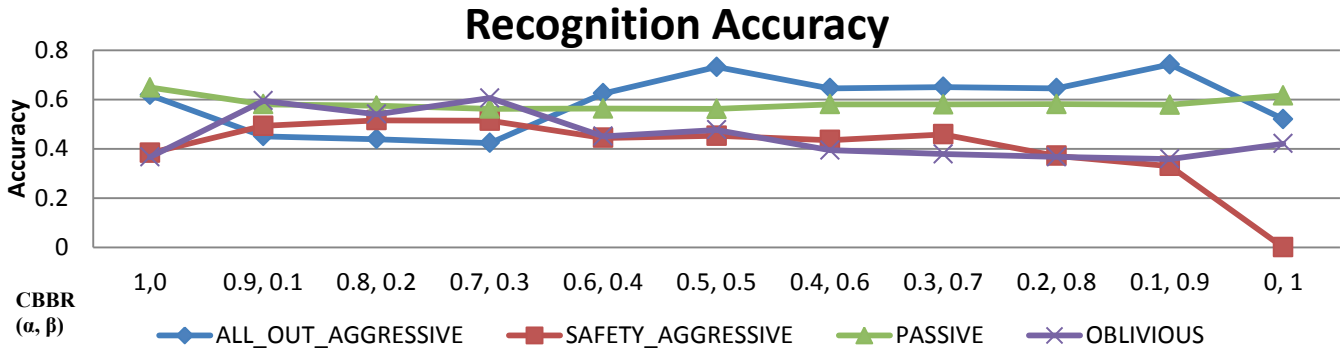


Figure 3: CBBR’s average recognition accuracies per behavior for each combination of weight settings tested

flanking maneuver from the right and left of the friendlies, respectively. For each trial the starting position, starting heading, and behavior of the agents were randomized within bounds to ensure *valid* scenarios, which are scenarios where the hostile agents nearly always enter radar range of the UAV. (Agents operating outside of the UAV’s radar range cannot be sensed by the UAV, which prevents behavior recognition.) During the case acquisition process we created case bases from a pool using the same base scenarios used during the experiment but with different randomized trials.

We set the pruning thresholds as follows:  $\tau_b = .97$ ,  $\tau_d = .973$ , and  $\tau_\alpha = .99$ . The thresholds for the similarity calculation were set at  $\tau_c = 0.1$  and  $\tau_s = 0.8$ . These values were hand-tuned based on insight from subject matter experts and experimentation. In the future we plan to tune these weights using an optimization algorithm.

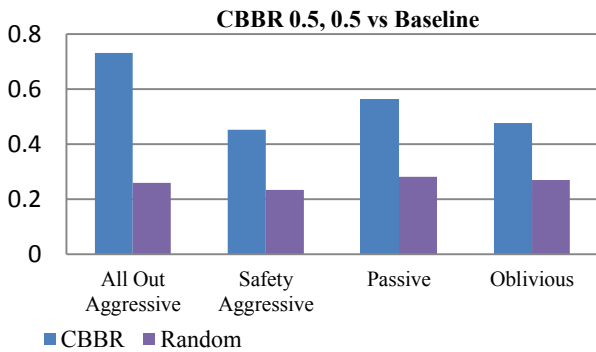


Figure 4: Comparing the average recognition accuracies of CBBR, using equal global weights, with the Random baseline

## 4.2 Results

We hypothesized that the CBBR algorithm would increase recognition performance in comparison to the baseline (**H1**) and that recognition performance should be higher than an ablation that does not use global features (**H2**).

Figure 3 displays the results for recognition accuracy. When testing CBBR with a variety of global weight settings, its best performance was obtained when  $\alpha = 0.5$

and  $\beta = 0.5$ , which has an average recognition accuracy of 0.55 over all behaviors (Figure 4), whereas this average is only 0.50 when  $\alpha = 1.0$  and  $\beta = 0.0$ . Figure 4 also shows that CBBR with  $\alpha = 0.5$  and  $\beta = 0.5$  outperforms Random for behavior recognition accuracy. Another observation (from Figure 3) is that the *Safety Aggressive* behavior clearly relies on time step features.

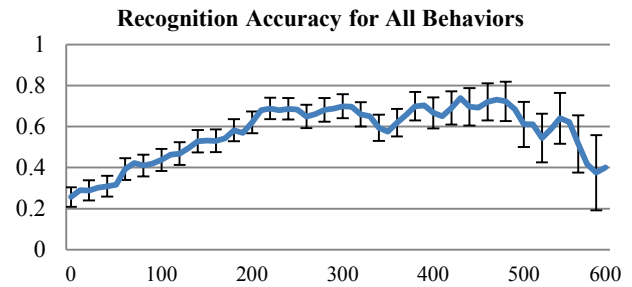


Figure 5: Average recognition accuracy (with standard error bars) for all behaviors versus simulation time

The experimental results support **H1**; for all values of  $\alpha$  and  $\beta$ , CBBR had better recognition accuracy than Random on a paired *t*-test ( $p > 0.08$ ). In particular, CBBR with weights  $\alpha = 0.5$  and  $\beta = 0.5$  significantly outperformed Random on a paired *t*-test ( $p > 0.01$ ) for all behaviors.

The results also provide some support for **H2**. In particular, we found that the average recognition accuracy of CBBR when  $\alpha = 0.5$  and  $\beta = 0.5$  is significantly higher than when  $\alpha = 1.0$  and  $\beta = 0.0$  ( $p > 0.04$ ). Therefore, this suggests that the inclusion of global features, when weighted appropriately, increases recognition performance.

Figure 5 displays CBBR’s ( $\alpha = 0.5$  and  $\beta = 0.5$ ) average recognition accuracy with standard error bars (across all behaviors) as the fair recognition time increases. This showcases how recognition accuracy varies with more observation time, and in particular demonstrates the impact of global features, given that their values accrue over time (unlike the time step features). In more detail, recognition accuracy starts out the same as would be expected of a random guess among the four behaviors (0.25), increases

over time, and finally becomes erratic due to the spurious “triggering” of the global features. For example an agent which is Safety Aggressive may during its’ execution be within radar range and missile range of another agent while actively trying to disengage (i.e. without meaning to be) and will then be categorized as *All Aggressive* because it triggered the INDANGER feature.

## 5. Conclusion

We presented a case-based behavior prediction (CBBR) algorithm for the real time recognition of agent behaviors for simulations of Beyond Visual Range (BVR) Air Combat. In an initial empirical study, we found that CBBR outperforms a baseline strategy, and that memory of agent behavior can increase its performance.

Our CBBR algorithm has two main limitations. First, it does not perform online learning. For our domain, this may be appropriate until we can provide guarantees on the learned behavior, which is a topic for future research. Second, CBBR cannot recognize behaviors it has not previously encountered. To partially address this, we plan to conduct further interviews with subject matter experts to identify other likely behaviors that may arise during BVR combat scenarios.

Our future work includes completing an integration of our CBBR algorithm as a component within a larger goal reasoning (Aha et al 2013) system, where it will be used to provide a UAV’s agent with state expectations that will be compared against its state observations. Whenever expectation violations arise, the GR system will be triggered to react, perhaps by setting a new goal for the UAV to pursue. We plan to test the CBBR’s contributions in this context in the near future.

## Acknowledgements

Thanks to OSD ASD (R&E) for sponsoring this research. The views and opinions contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of NRL or OSD. Additional thanks to Josh Darrow and Michael Floyd for their insightful conversations/feedback.

## References

Aha, D.W., Molineaux, M., & Ponsen, M. (2005). Learning to win: Case-based plan selection in a real-time strategy game. *Proceedings of the Sixth International Conference on Case-Based Reasoning* (pp. 5-20). Chicago, IL: Springer.

Aha, D.W., Cox, M.T., & Muñoz-Avila, H. (Eds.) (2013). *Goal reasoning: Papers from the ACS Workshop* (Technical Report CS-TR-5029). College Park, MD: University of Maryland, Department of Computer Science.

Auslander, B., Apker, T., & Aha, D.W. (2014). Case-based parameter selection for plans: Coordinating autonomous vehicle teams. *Proceedings of the Twenty-Second International Conference on Case-Based Reasoning* (pp. 189-203). Cork, Ireland: Springer.

Auslander, B., Lee-Urban, S., Hogg, C., & Muñoz-Avila, H. (2008). Recognizing the enemy: Combining reinforcement learning with strategy selection using case-based reasoning. *Proceedings of the Ninth European Conference on Case-Based Reasoning* (pp. 59-73). Trier, Germany: Springer.

Borck, H., Karneeb, J., Alford, R., & Aha, D.W. (2014). Case-based behavior recognition to facilitate planning in unmanned air vehicles. In S. Vattam & D.W. Aha (Eds.) *Case-Based Agents: Papers from the ICCBR Workshop* (Technical Report). Cork, Ireland: University College Cork.

Jaidee, U., Munoz-Avila, H., & Aha, D.W. (2013). Case-based goal-driven coordination of multiple learning agents. *Proceedings of the Twenty-First International Conference on Case-Based Reasoning* (pp. 164-178). Saratoga Springs, NY: Springer.

Kabanza, F., Bellefeuille, P., Bisson, F., Benaskeur, A. R., & Irandoust, H. (2014). Opponent behavior recognition for real-time strategy games. In G. Sukthankar, R.P. Goldman, C. Geib, D.V. Pynadath, & H.H. Bui (Eds.) *Plan, Activity, and Intent Recognition*. Amsterdam, The Netherlands: Elsevier.

Laviers, K., Sukthankar, G., Klenk, M., Aha, D.W. & Molineaux, M. (2009). Opponent modeling and spatial similarity to retrieve and reuse superior plays. In S.J. Delany (Ed.) *Case-Based Reasoning for Computer Games: Papers from the ICCBR Workshop* (Technical Report 7/2009). Tacoma, WA: University of Washington Tacoma, Institute of Technology.

Nielsen, P., Smoot, D., & Dennison, J.D. (2006). *Participation of TacAir-Soar in RoadRunner and Coyote exercises at Air Force Research Lab, Mesa AZ* (Technical Report). Ann Arbor, MI: Soar Technology Inc.

Rao, A.S., & Murray, G. (1994). Multi-agent mental-state recognition and its application to air-combat modelling. In M. Klein (Ed.) *Distributed Artificial Intelligence: Papers from the AAAI Workshop* (Technical Report WS-94-02). Menlo Park, CA: AAAI Press.

Smith, R.E., Dike, B.A., Mehra, R.K., Ravichandran, B., & El-Fallah, A. (2000). Classifier systems in combat: Two-sided learning of maneuvers for advanced fighter aircraft. *Computer Methods in Applied Mechanics and Engineering*, 186(2), 421-437.

Smyth, B., & Keane, M.T. (1995). Remembering to forget. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 377-382). Montreal, Quebec: AAAI Press.

Sukthankar, G., & Sycara, K. (2006). Simultaneous team assignment and behavior recognition from spatio-temporal agent traces. *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (pp. 716-721). Boston, MA: AAAI Press.

Sukthankar, G., & Sycara, K.P. (2011). Activity recognition for dynamic multi-agent teams. *ACM Transactions on Intelligent Systems Technology*, 18, 1-24.

Weber, B. G., Mateas, M., & Jhala, A. (2010). Case-based goal formulation. In D.W. Aha, M. Klenk, H. Muñoz-Avila, A. Ram, & D. Shapiro (Eds.) *Goal-Directed Autonomy: Notes from the AAAI Workshop (W4)*. Atlanta, GA: AAAI Press.