Case-Based Behavior Recognition
to Facilitate Planning in Unmanned Air Vehicles

Hayley Borck[1], Justin Karneeb[1], Ron Alford[2], David W. Aha[3]

[1]Knexus Research Corporation; Springfield, VA; USA
[2]ASEE Postdoctoral Fellow; Naval Research Laboratory (Code 5514); Washington DC; USA
[3]Navy Center for Applied Research in Artificial Intelligence;
Naval Research Laboratory (Code 5514); Washington, DC; USA
{*first.last*}@knexusresearch.com | {david.aha, ronald.alford.ctr}@nrl.navy.mil

Abstract. An unmanned air vehicle (UAV) can operate as a capable team member in mixed human-robot teams if the agent that controls it can intelligently plan. However, planning effectively in an air combat scenario requires understanding the behaviors of hostile agents in that scenario, which is challenging in partially observable environments such as the one we study. 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 can then be given as input to an air combat simulation, along with the UAV's plan, to predict hostile actions and determine the effectiveness of the given plan. We describe an initial implementation of a CBBR prototype in the context of a goal reasoning agent designed for UAV control.

1   Introduction

Unmanned air vehicles (UAVs) can be capable wingmen in air combat scenarios when given an accurate plan to execute [1]. However, planning may be ineffective if the behaviors of the other agents operating in the world are unknown. To effectively account for hostile and allied agents we will use a Case-based Behavior Recognition (CBBR) algorithm that, in combination with a predictive planner, can effectively evaluate UAV plans in real time. 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.

We define a *behavior* as an overarching tendency or policy of the agent. Behaviors are encoded in a directed graph where each node is an action, such as 'fly to target' or 'fire missile'. The domain we are working with is *Beyond Visual Range Air Combat*, which entails precise tactics at large distances. In this domain we have little data about the hostile agents, and what we do have is partially observable. Yet if the UAV can identify a hostile agent's behavior or plan it can use that information when creating and evaluating its own plan.

We hypothesize that behavior recognition is more effective than plan recognition in domains where information is scarce. We designed our CBBR implementation so that, by discretizing state information over time, it can identify a hostile agent's current behavior. CBBR currently operates in two 2 vs 2 scenarios (i.e., each scenario involves two 'friendly' aircraft versus two 'enemy' aircraft). In our first scenario a pilot and their UAV wingman are conducting an attack, while in the second they are defending a specified area.

In the rest of this paper we describe our agent for intelligent control of UAVs in the Beyond Visual Range Air Combat domain, focusing on its CBBR component. In Section 2 we summarize related work. In Section 3 we provide a model of the Tactical Battle Manager (TBM), which includes our CBBR component. In Section 4, we describe its case structure and similarity function. Section 5 details a simple example, and Section 6 concludes and describes potential future work.

2 Related Work

Our behavior recognition component, which lies within a larger goal reasoning (GR) agent (i.e., the TBM), can determine if a UAV wingman's plan is effective. In recent years, case-based reasoning (CBR) has been an active area of research for GR agents. For example, Weber et al. [2] use a case base to formulate new goals for an agent. Jaidee et al. [3] uses 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.

Opponent agents can be recognized as a team or as a single agent. Team composition can be dynamic [4], resulting in a more complex version of the plan recognition problem [5]. Another approach to team dynamics involves setting multiagent planning parameters, as addressed by Auslander et al. [6], which are then given to a plan generator. Recognizing higher-level behaviors encompasses these team behaviors. For example, two hostile agents categorized as 'all out aggressive' in our system could, acting according to the 'all out aggressive' graph, execute a pincer maneuver (a maneuver in which two agents attack both flanks of an opponent).

A challenging task in agent planning is inferring the states of any adversarial agents because their strategies can change over time. Auslander et al. [7] use a case-based reinforcement learner to combat changing conditions and overcome slow learning by employing a case base of winning policies. Rao and Murray [8] store the mental states of the agent representing their beliefs, desires, and intentions and use those to synthesize plans. Similarly, Jaidee et al. [9] use dual case bases to learn goals and agent policies, making their approach more flexible than either case-based learning or RL alone. Smith et al. [10] use a genetic algorithm (GA) system to develop effective tactics for their agents in a two-sided learning experiment. Aha et al. [11] employed a case base to select sub-plans for agents at each state and keep the opponent agents at bay. To ensure our case-based solutions are robust to dynamic behaviors, we use global features in our cases to serve as a memory of past actions

and tendencies. We also frequently update the agent's behaviors, which enables the most recent information to be used for future planning.

3   Tactical Battle Manager

The TBM (Figure 1) is a set of systems for pilot-UAV interaction and autonomous UAV control. The UAV's intelligent controller, which is the focus of this paper, takes as input an incomplete world state and outputs, and subsequently executes, a plan for the UAV. Each known agent in a scenario is represented in the world model, which contains the agent's past observed states and future predicted states as well as its capabilities and currently recognized behavior. A complete state contains, for each time step in the simulation, the position and actions for each known agent. An example of an action in our system is 'fire missile' or 'fly to target'. 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 or the radar of one of its allies. Also, a hostile agent's actions are never known and must be inferred from the potentially incomplete past states. The capabilities of an aircraft are currently given, though in future work they will be inferred through observations. In Section 4 we describe the behaviors and how they are modeled by the CBBR algorithm.

The updated world model is passed to the Goal Management System (GMS). This follows the normal goal reasoning cycle and is complemented by a *desire* system similar to a Belief Desire Intention (BDI) [12] architecture. The GMS maintains a set of goals based on the world model; it adds, removes, and reprioritizes them as necessary. These goals are used to generate a plan for the UAV with a corresponding set of predicted states for all agents. We refer to the system that performs these tasks as the Predictive Planner. Currently this planner is simple. However, we use a more sophisticated Plan Expectation Predictor (PEPR) to generate the predicted states; it runs an instance of the Air Force Simulator (AFSIM), which is a mature air combat simulation engine that is used by the USAF. AFSIM simulates the plan for the UAV and the other agents in a scenario by projecting their behaviors to determine the effectiveness of the UAV's plan. Thus, the predicted future states are only as accurate as the behaviors contained in our models.
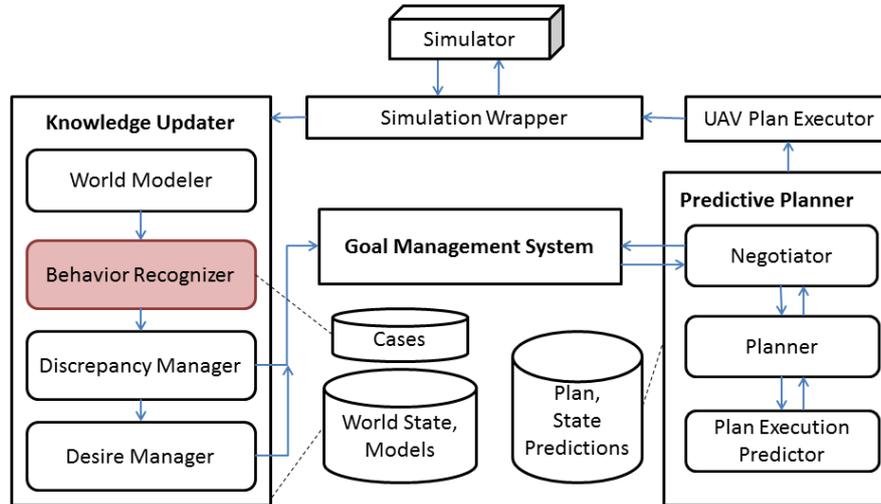
Fig. 1. Tactical Battle Manager (TBM) Architecture

4 Case-Based Behavior Recognition

The following subsections describe the CBBR algorithm in detail. The traditional CBR cycle consists of four steps: retrieval, reuse, revise, and retain. Currently our algorithm employs only steps for retrieval and reuse. In future work we plan to expand the algorithm to include steps for revision and retention.

4.1 Case Representation

A case in our system represents an agent over time. Cases are represented as ⟨problem, solution⟩ pairs. A *problem* is represented by a set of features that discretize the agent's model, while a *solution* is the behavior the agent was employing. The feature set contains two feature types: features that occur at a specific time step and global features (Figure 2). 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. Features can be represented as a boolean value or a percentage. We represent some features using a percentage value because it more fully describes a situation than does a boolean. For example the hasTrack feature, which describes whether an agent has another agent in its radar, is defined as the ratio of agents it has in its radar versus the total number of agents it currently knows exists in a scenario.

The currently modeled behaviors are:

- *All Out Aggressive*: an agent attacks and is not concerned for its safety.
- *Safety Aggressive*: an agent that attacks but has concern for its safety.

- *Defensive*: an agent that only attacks when a hostile agent is within a certain area.
- *Oblivious*: an agent that acts as if hostile agents are not near.
- *Passive*: an agent that knows hostile agents are near but does not attack.
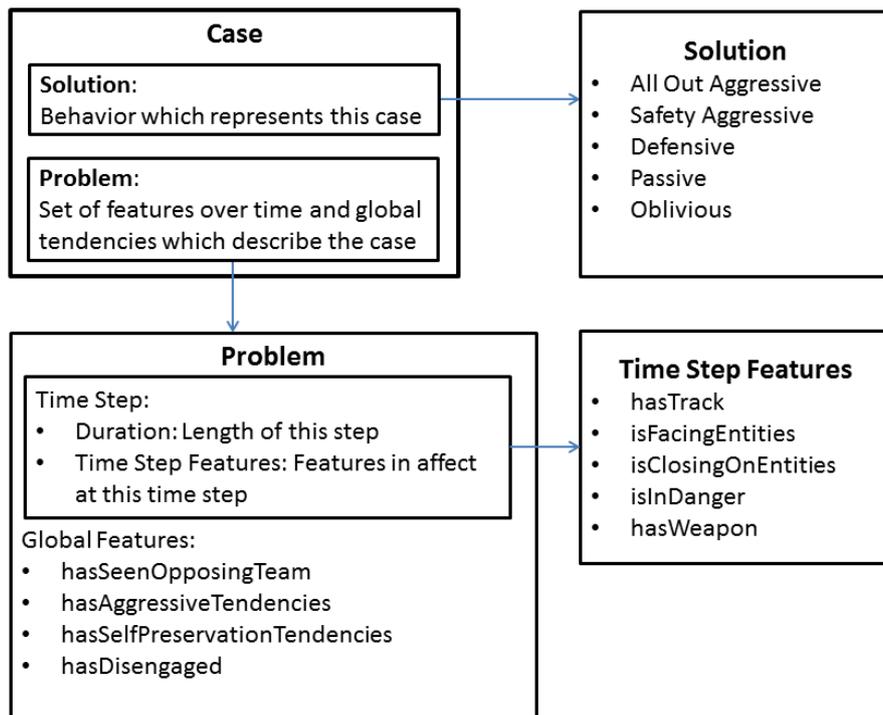

Fig. 2. A case's design, including problem features and solution behaviors

4.2   Case Base Population

We populated our case base by running several 2 vs 2 scenarios in AFSIM, where the hostiles were encoded with explicit behaviors to exhibit. For example, a 2 vs 2 scenario was run where both hostiles had *all out aggressive* behaviors and the pilot and UAV ran simple *passive* behaviors (in which they try to keep the hostiles in radar range but do not attack). Cases are created from the hostiles in the scenario and recorded in an xml file. We prune the cases twice; first during case generation and also after all the scenarios have been run. The first stage of pruning prevents cases with the same problem features and solution behavior to be added to the case base. The second stage deletes cases from the case base if their problem features are identical but their behaviors differ.

4.3 Case Similarity

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 in cases. We use a weight of α for time step features and β for global features, where α and β are both non-negative numbers that sum to 1. If a query contains mismatched features to a case (features that are not present in the case, or features in a case that are not present in the query) then those features are ignored in the similarity equation. Similarity is calculated in reverse chronological order, with a discount factor δ applied based on how far in the past the feature occurred. The full equation is shown below, where $\sigma(q_f, c_f)$ is the distance between two values for (matching) feature f, N is the set of time step features, and M is the set of global features.

$$\text{sim}(q, c) = -\alpha \frac{\sum_{f \in N} \left(\delta \cdot \sigma(q_f, c_f) \right)}{N} - \beta \frac{\sum_{f \in M} \sigma(q_f, c_f)}{M} \qquad (1)$$

We are currently identifying values to use for these weights and the discount factor. Future work will include optimizing these variables.

Once the case with the most similar problem description is found its (solution) behavior is retrieved and used as the predicted behavior of the currently observed agent. The world model is also updated with that agent's predicted behavior, which is used by PEPR to predict future states.

5 Discussion

In Section 5.1 we present a simple example of the case structure and similarity metric in the domain of Beyond Visual Range Air Combat. Following that, in Section 5.2 we briefly describe the evaluations we intend to conduct in the future.

5.1 A Simple Example

In a simple example of the CBBR system, we have a case base in a 2 vs 2 scenario. The agents are modeled using discrete time step and global features. Here we define each case to have time steps of 5 seconds (i.e., a trace of 15 seconds of observed states is split into three time steps). Global features are extracted from the entire trace's observed states. Below we show an example of a query q_1, where a hostile agent followed an agent friendly to the UAV for two time steps and then turned away at the third time step.

```
[q₁] Behavior: ?
List<TimeStep> timeSteps =
  {d=5, hasTrack(.5), isFacing(.5) hasWeaponLeft(T)}
  {d=5, hasTrack(.5), isFacing(.5), hasWeaponLeft(T),
       isClosingOnEntities(.5)}
  {d=5. hasTrack(.5), hasWeaponLeft(T)}
```

```
List<GlobalFeature> gFeatures =
  {hasSeenOpposingTeam(.5), hasAggressiveTendencies(.5)}
```

In query q_1 we can see the hostile agent is following a friendly agent because it has a friendly in its radar (*hasTrack*), is facing a friendly agent, and is closing on a friendly agent. Since there are two friendly agents in the scenario but the hostile is only following one of them the features have a value of 0.5. We do not record which agent this hostile is following, but only that it is following one of them. This is because knowing which friendly the hostile agent is following will not affect which behavior the agent is exhibiting. The *hasWeaponLeft* time step feature is the only one shown that is represented by a boolean value. (In this example we did not infer that the hostile fired a weapon, and therefore believe it still has one or more weapons remaining.)

For this example consider two cases in the case base, c_2 and c_3. Case c_2 is an example of a *passive* behavior, which often involves flying away from an enemy and avoiding conflict. Case c_3 is an example of an *all out aggressive* behavior, which is similar to the query q_1. The case retrieval step would return case c_3 due to the similarity of the features in their first two time steps, and their global features. As mentioned previously the mismatched features at the third time step do not count against the similarity between q_1 and either of the other cases. Thus, for this situation the agent described by query q_1 would be predicted to be an *all out aggressive* agent.

```
[C₂] Behavior: Passive
List<TimeStep> timeSteps =
  {d=5, hasWeaponLeft(T)}
  {d=5, hasWeaponLeft(T)}
List<GlobalFeature> gFeatures =
  {hasSelfPreservationTendencies(.5)}

[C₃] Behavior: All Out Aggressive
List<TimeStep> timeSteps =
  {d=5, hasTrack(1), isFacing(1), hasWeaponLeft(T)}
  {d=5, hasTrack(1), isFacing(1), hasWeaponLeft(T),
      isClosingOnEntities(1)}
List<GlobalFeature> gFeatures =
  {hasSeenOpposingTeam(.5), hasAggressiveTendencies(.5)}
```

5.2   Future Empirical Studies

To evaluate our CBBR component we plan to conduct several experiments. The objective of the first experiment will be to determine the effectiveness of CBBR as compared to other behavior recognizers, including baseline algorithms. To do this we will compare CBBR to a random behavior choice, a random behavior choice based on a predetermined percentage, and a rule-based system. Additionally, since we hypothesize a behavior recognizer is more robust than a plan recognizer in a domain with partial information we will compare the two approaches empirically. Lastly we

plan to assess the effectiveness of the UAV's plan, since the end goal of CBBR is to help identify whether a UAV's plan will succeed as predicted by PEPR.

6   Summary

In this paper we presented a Case-Based Behavior Recognizer that, in our domain (Beyond Visual Range Air Combat), facilitates planning in unmanned air vehicles. This behavior recognizer is given a trace of spatio-temporal information, which may be incomplete. Our CBBR component is designed to identify overarching behaviors (e.g., aggressive or passive) rather than plans. In our future work we will empirically compare CBBR versus other behavior and plan recognizers, and also assess the effectiveness of the plan. We will also expand the behavior recognizer to reason with possibly mislabeled state information and more complex team tactics.

Acknowledgements

References

1. Nielsen, P., Smoot, D., Dennison, J.D.: Participation of TacAir-Soar in RoadRunner and Coyote Exercises at Air Force Research Lab, Mesa AZ. Technical report (2006)
2. Weber, B. G., Mateas, M., Jhala, A.: Case-based Goal Formulation. In: AAAI Workshop on Goal-Driven Autonomy (2010)
3. Jaidee, U., Munoz-Avila, H., & Aha, D.W.: Case-based goal-driven coordination of multiple learning agents. In: 21st International Conference on Case-Based Reasoning. (2013) pp. 164-178.
4. Sukthankar, G., Sycara, K.P.: Simultaneous Team Assignment and Behavior Recognition from Spatio-Temporal Agent Traces. In: 21st National Conference on Artificial Intelligence. (2006) 716-721
5. Sukthankar, G., Sycara, K.P.: Activity Recognition for Dynamic Multi-Agent Teams. ACM Trans. on Intell. Syst. Technol. (2011) 18
6. Auslander, B., Apker, T., Aha, D.W.: Case-Based Parameter Selection for Plans: Coordinating Autonomous Vehicle Teams. In: 22nd International Conference on Case-Based Reasoning. (2014) 189-203
7. Auslander, B., Lee-Urban, S., Hogg, C., Muñoz-Avila, H.: Recognizing the Enemy: Combining Reinforcement Learning with Strategy Selection Using Case-Based Reasoning. In: 9th European Conference on Case-Based Reasoning. (2008) 59-73
8. Rao, A.S., Murray, G.: Multi-agent Mental-State Recognition and its Application to Air-Combat Modelling. In: 13th International Workshop on Distributed Artificial Intelligence (1994) 283-304

9. Jaidee, U., Muñoz-Avila, H., Aha, D.W.: Case-based Learning in Goal-driven Autonomy Agents for Real-Time Strategy Combat Tasks. In: 19th International Conference on Case-Based Reasoning. (2011) 43-52

10. Smith, R.E., Dike, B.A., Mehra, R.K., Ravichandran, B., El-Fallah, A.: Classifier Systems in Combat: Two-Sided Learning of Maneuvers for Advanced Fighter Aircraft. Computer Methods in Applied Mechanics and Engineering 186(2) (2000) 421-437

11. Aha, D.W., Molineaux, M., Ponsen, M.J.V.: Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game. In: International Conference on Case-Based Reasoning. (2005) 5-20

12. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: 2nd International Conference on Principles of Knowledge Representation and Reasoning. (1991) 473-484