

# Goal Reasoning: Foundations, Emerging Applications, and Prospects

*David W. Aha*

■ *Goal reasoning has a bright future as a foundation for the research and development of intelligent agents. Goal reasoning is the study of agents that can deliberate on and self-select their objectives, which is a desirable capability for some applications of deliberative autonomy. This capability is of interest to several AI subcommunities and applications. Our group has focused on how goal reasoning can assist with controlling autonomous systems. The importance of how agents reason about goals is growing and it merits increased attention, particularly from the perspective of research on AI safety. In this article, I introduce goal reasoning, briefly relate it to other AI topics, summarize some of our group's work on goal reasoning foundations and emerging applications, and describe some current and future research directions.*

## An Omnipresent Phenomenon

The popular 1946 film *It's a Wonderful Life* focuses on the trials and tribulations of George Bailey, who decides on the objective of suicide as the best remedy for his seemingly insurmountable situation. At the film's crucial turning point, an event occurs that compels George to instead save Clarence Odbody, his guardian angel. Clarence helps George to see the great value of his life, which leads George to abandon death and instead choose life. This dramatic change in goals has many other examples in storytelling and cinema. In the 2015 film *The Force Awakens*, for example, we encounter FN-2187, a reluctant stormtrooper (later dubbed "Finn") who, after observing a horrific event, decides to rebel against his superiors by shifting his allegiance from the evil Galactic Empire to the good Resistance. Again, this represents not only a change in plans, but a change in objectives.

Dynamic goal reprioritization is not limited to the entertainment realm. Before the events of December 7, 1941, for example, the USA pursued the stated objective of peace with the Axis powers, but this soon changed. Governments continue to alter their goals based on the results of elections, on international incidents, and on other motivating factors.

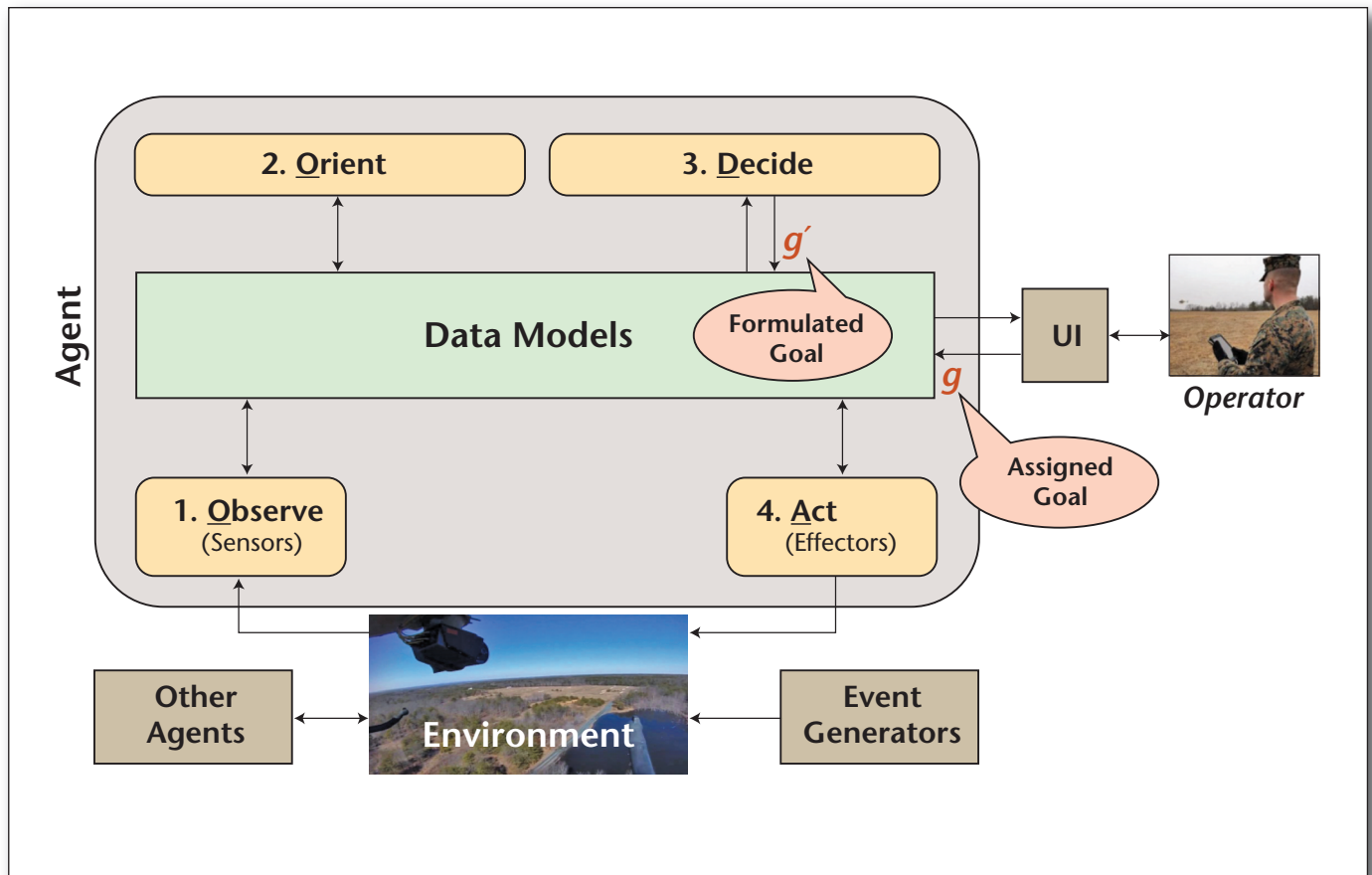


Figure 1. Goal Reasoning Agents Can Formulate Their Own Goals.

There is also evidence that we humans continually manage our own goals as we react to our evolving situation (Altmann and Trafton 2002). We may suspend current goals to pursue others (for example, that relate to biological, emotional, or social needs), or even abandon goals we deem to be unachievable or of lower priority. Such goal switching may occur, for example, when we receive a request from our supervisor, when we unexpectedly encounter an old friend, or when we are given tickets to attend a favorite event.

Similarly, intelligent agents may benefit from deliberating about, and changing, their active goals when warranted. This flexibility may allow them to behave competently when they are not preencoded with a model that dictates what goals they should pursue in all encounterable situations.

I refer to *goal reasoning* (GR) as the process by which intelligent agents continually reason about the goals they are pursuing, which may lead to goal change (Cox 2007; Muñoz-Avila et al. 2010; Klenk et al. 2013; Vattam et al. 2013). This general topic has been studied, using different terminology, in multiple disciplines for several decades. In this article, I summarize our group's research on GR, which has

been strongly influenced by perspectives on cognitive architectures and symbolic task planning.

## Situating Goal Reasoning Agents

Figure 1 highlights a key property of interactive GR agents, where we use an observe, orient, decide, act (OODA) loop to frame the agent's decision cycle.<sup>1</sup> In this figure, we assume a human operator can interact with the agent, at least to provide an initial objective or objectives. In contrast to some others, GR agents can deliberate on a space of goals, dynamically adjust goal priorities, and perform goal-management functions (for example, formulation, commitment, and suspense).

Many dimensions for goals exist. For example, borrowing and adapting from van Riemsdijk et al.'s (2008) taxonomy, these include (among others) type, specificity, duration, purpose, condition, and persistence.

*Type:* Goals can be declarative (referring to belief states) or procedural (referring to actions).

*Specificity:* Goals may refer to a concrete instance or an abstraction (for example, region of belief states, sequence of actions).

| Environment Dimension | Simple        | Complex                      |
|-----------------------|---------------|------------------------------|
| Operator Availability | Constant      | Intermittent or Inaccessible |
| Goal Model            | Complete      | Partial                      |
| Accessibility         | Full          | Partial                      |
| Updates               | Static        | Dynamic                      |
| Action Effects        | Discrete      | Continuous                   |
| Action Outcomes       | Deterministic | Stochastic                   |
| Agents                | Single        | Multiple                     |

Table 1. Goal Reasoning Agents Are Most Appropriate for Complex Environments.

*Duration:* Goals may refer to a static time point or be durative.

*Purpose:* Some goals are designed to learn world knowledge (that is, query or knowledge goals [Bengfort and Cox 2015]), while others are attainment goals (that is, they exploit such knowledge).

*Condition:* Goals can be unconditional, or conditioned on beliefs or other goals.

*Persistence:* Goals may, or may not, be interruptible.

Here I focus on goals that are declarative, that are specific points in belief space not involving knowledge acquisition, that are unconditional and that can be interrupted, though extensions of GR should be amenable to other types of goals and dimensions.

GR agents are intended for use in complex environments (Russell and Norvig 2016), meaning that they are characterized by some of the properties highlighted in table 1.

Most of these properties are well studied — for example, a dynamic environment is one in which other agents and effectors (such as weather) can modify the environment during the agent’s decision-making cycle. However, two properties are especially relevant to goal reasoning: operator availability and goal model.

*Operator availability:* If the agent’s human operator is always available, then they could potentially provide continuous control, alleviating any need for agent self-control. GR is relevant in situations where the operator is not always available.<sup>2</sup>

*Goal model:* If the agent is given a complete function  $F_g: S \rightarrow G$  that determines what goal  $g \in G$  should be pursued for all encounterable situations  $s \in S$ , then

there is no need to perform dynamic inference in support of goal reasoning. Rather, GR becomes a (degenerate) retrieval task. We are interested in rich environments where it is infeasible to provide such a complete function.

GR agents are not intended for all environments and scenarios. In particular, their ability to perform goal reprioritization is not useful unless they encounter situations that warrant goal reprioritization (for example, impasses or affordances requiring goal deliberation).

## Inspirations

Topics relevant to goal reasoning have been studied in computer science (for example, software engineering, AI), cognitive science (for example, cognitive modeling, cognitive architectures), robotics, and philosophy, among other disciplines. We briefly summarize a small subset of related work in symbolic planning, cognitive architectures, and intelligent agents that has motivated our group’s research.

## Planning

Most research on symbolic task planning pertains to the following problem: given initial and goal states  $i, g \in S$  from a set of states  $S$ , and a model  $M_x$  of actions that can be applied to traverse among these states, generate a plan  $\pi$  that can be applied in  $i$  to traverse into  $g$ . In this classical formulation, no monitoring of the plan’s execution takes place, and the agent cannot change  $g$ . Many investigations have relaxed these assumptions, as can be seen, for example, in

continual planning (Chien et al. 2000), where human operators can provide an agent with additional goals during run time, and oversubscription planning (Smith 2004), where the planner must reason about which among conflicting goal(s) it should attempt to achieve. More recent work has described planning in the context of a class of conditional goals by reasoning about trade-offs among sensing costs and goal rewards (Talamadupula et al. 2010). These and other dynamic deliberation tasks closely relate to GR, and one perspective is that GR is a methodology for plan monitoring in the context of planning and acting (Ghallab et al. 2014). In the section on foundations, I describe a process model of goal reasoning that borrows heavily from symbolic task planning models.

### Cognitive Architectures

Solutions for GR have frequently been included in cognitive architectures. For example, Soar’s universal subgoaling provides a process for responding to impasses during problem solving by posting a new subgoal to solve (Laird and Newell 1983). To use this approach, TacAir-Soar was provided with a top-down goal hierarchy that encodes doctrine, missions, and tactics for its simulated air vehicles to perform, along with a bottom-up hierarchy of rules to guide interrupt processing (Jones and Laird 1997). An extreme form of this approach would provide an agent with a complete goal hierarchy, relieving any need for goal deliberation but requiring a complete model  $M_g$  for dynamic goal selection. Marinier, van Lent, and Jones (2010) later discussed the use of appraisal theories in SOAR to support GR processes. Choi (2011) instead extended Icarus (Langley et al. 2004) to nominate top-level goals (from a long-term memory of general goals) and continuously manage them through a prioritization function. Altmann and Trafton (2002) describe the integration of a model in Act-R (Anderson and Lebiere 1998) to replace its architectural goal stack for managing goals, and show that goal-directed behavior can be explained using mechanisms of goal activation and associative priming. Finally, Cox et al. (2016) describe MIDCA, which models a metacognitive process for goal change and a cognitive process for goal generation. It manages unexpected events in dynamic environments. Our work differs from this body of research in that we investigate a variety of GR process models independent of a broad psychological theory that guides and constrains cognitive architecture design.

### Intelligent Agents

Many GR contributions have been proposed in the context of intelligent agents. Cox (2007) describes a perpetual self-aware cognitive agent that was designed for continuous autonomous operation in complex environments. Its integration of planning, execution, and goal generation components directly

inspired our research on GR. Coddington and Luck (2004) describe how an agent’s context (that is, its environment situation) can be used to constrain goal selection and prioritization. Coddington (2006) later describes MADbot, an agent that dynamically generates goals in response to its internal motivation model. Research on motivated agents (Hawes 2011), largely inspired by belief-desire-intention (BDI) architectures, has addressed the representation of goal types, their properties, and reasoning lifecycles (for example, Braubach et al. 2004; Dastani and van der Torre 2004; Harland et al. 2014). Many agent programming languages support automated planning in the context of BDI architectures (Meneguzzi and de Silva 2015); they relax several environment assumptions that are common to earlier research. While we share interest in such environments, we do not always use a BDI-inspired framework for our GR agents, nor encode them in a specific agent programming language.

## Foundations

Cox (2018) presents a formal model of goal reasoning, defining goal transformation function  $\beta(s, g) \rightarrow g'$  to return a (possibly new) goal  $g'$  given current state  $s$  and goal  $g$ .  $\beta(\cdot)$  maps (state,goal) pairs to a new goal, thus modeling a GR agent’s function for dynamically selecting goals. Cox further introduces a model of goal change  $\Delta = \{\delta \mid \delta: G \rightarrow G\}$  that represents a set of potential transformations on goals (for example, null, deletion, subgoaling) that the agent may select. For a sequence of decisions  $\delta_1, \delta_2, \dots, \delta_n \in \Delta$ , the result will be

$$\delta_n(\dots\delta_2(\delta_1(g))) = g'$$

Cox uses this model as the basis to formalize notions of planning, acting, and interpretation. Cox also shows how this formalization can be applied to describe the GR model we introduce next.

Our GR research began with investigations of goal-driven autonomy (GDA) (Molineaux, Klenk, and Aha 2010), which is a simple anomaly-driven agent model. GDA extends Nau’s (2007) framework for online planning by introducing a process for its controller. In figure 2, we display this extended framework by decomposing Nau’s controller into *orient* and *decide* subprocesses. As is common, we model the environment as a tuple  $M_\Sigma = (S, A, E, \gamma)$  with states<sup>3</sup>  $S$ , actions  $A$ , exogenous events  $E$ , and state transition function  $\gamma: S \times (A \cup E) \rightarrow 2^S$ , which describes how an action’s execution (or an event’s occurrence) transforms the environment from one state to another in  $S$ . (We do not assume that  $M_\Sigma$  is complete or correct.) The initial state, obtained from Observe’s sensors, is stored in  $s$ , while the operator provides goal  $g$ . The planner, given both of these and  $M_\Sigma$ , generates plan  $\pi = \langle a_1, \dots, a_n \rangle$  (that is, a sequence of actions, each in  $A$ ) and

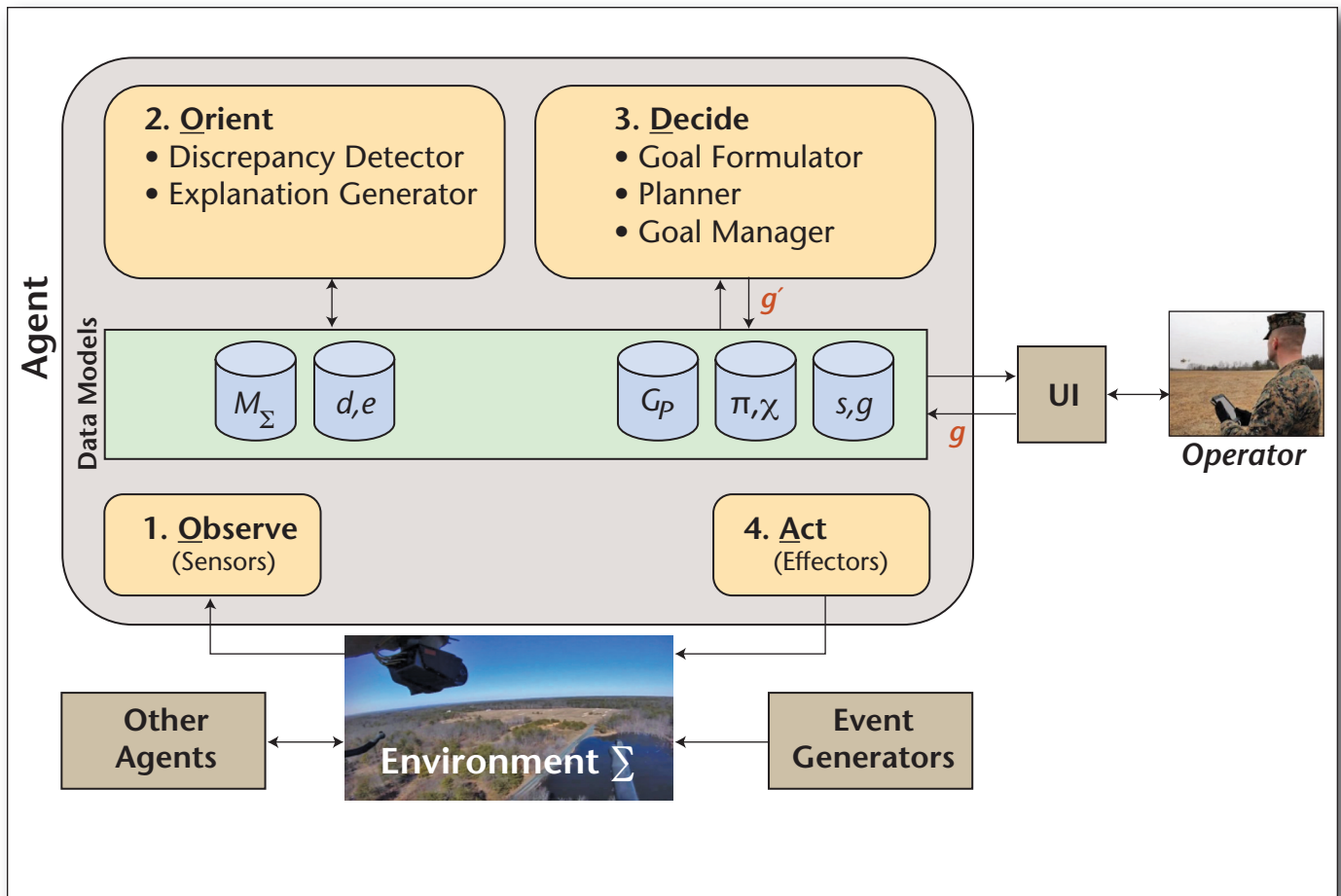


Figure 2. A Depiction of the Goal-Driven Autonomy (GDA) Model of Goal Reasoning.

expectations  $\chi = \langle \chi_1, \dots, \chi_n \rangle$  (that is, distributions of expected states in  $S$  after executing each sequential action  $a_1 \in \pi$ , starting in  $s$ ).

The GDA model monitors  $\pi$ 's execution. This involves four steps: (1) discrepancy detection, (2) explanation generation, (3) goal formulation, and (4) goal management.

Step 1, *discrepancy detection*, compares the observations obtained from executing action  $a_i$  in belief state  $s_i$  with expectation  $\chi_i$  (that is, this tests whether any constraints are violated, corresponding to unexpected observations). If a discrepancy  $d$  is found, then it is given to step 2.

Step 2, *explanation generation*. Given next state  $s_{(i+1)}$  (provided by  $\gamma$ ),  $s_i$ , and  $d$ , this process hypothesizes an explanation  $e \in E$  (not shown in figure 2) of its cause.

Step 3, *goal formulation*. Given  $d$ ,  $e$ , and  $s_{(i+1)}$ , this process generates a goal  $g' \in G$  (not shown).

Step 4, *goal management*. Given a set (initially empty) of pending goals  $G_p \subseteq G$  and  $g'$ , this process may update  $G_p$  and will select the next goal  $g$  to feed to the planner.

GDA does not specify what algorithms to use for these processes or what representations to use for

these data models. Details of our initial GDA agent, ARTUE, and its analysis are given in Klenk et al. (2013).

Several groups have used GDA as a starting point for research on GR. For example, Molineaux and Aha (2015) describe an abductive method for continuous explanation generation that employs a constrained heuristic search to identify plausible explanations, given unexpected observations.<sup>4</sup> This method may result in modifying initial state assumptions or the action models of other agents in the environment. Revised models can be learned and used to interpret future similar occurrences (Molineaux and Aha 2014). Powell et al. (2011) use active learning techniques with GDA to acquire a function that maps states to goals. Weber, Mateas, and Jhala's (2012) EIS-Bot plays a complete real-time strategy (RTS) game that uses GDA to select objectives (that is, which units to produce). Jaidee, Muñoz-Avila, and Aha (2013) describe a GDA variant that uses reinforcement learning to learn a goal selection function for each unit type in an RTS game. Paisner et al. (2014) describe how to model GDA in MIDCA. Dannenhauer, Muñoz-Avila, and Cox (2016) instead extend

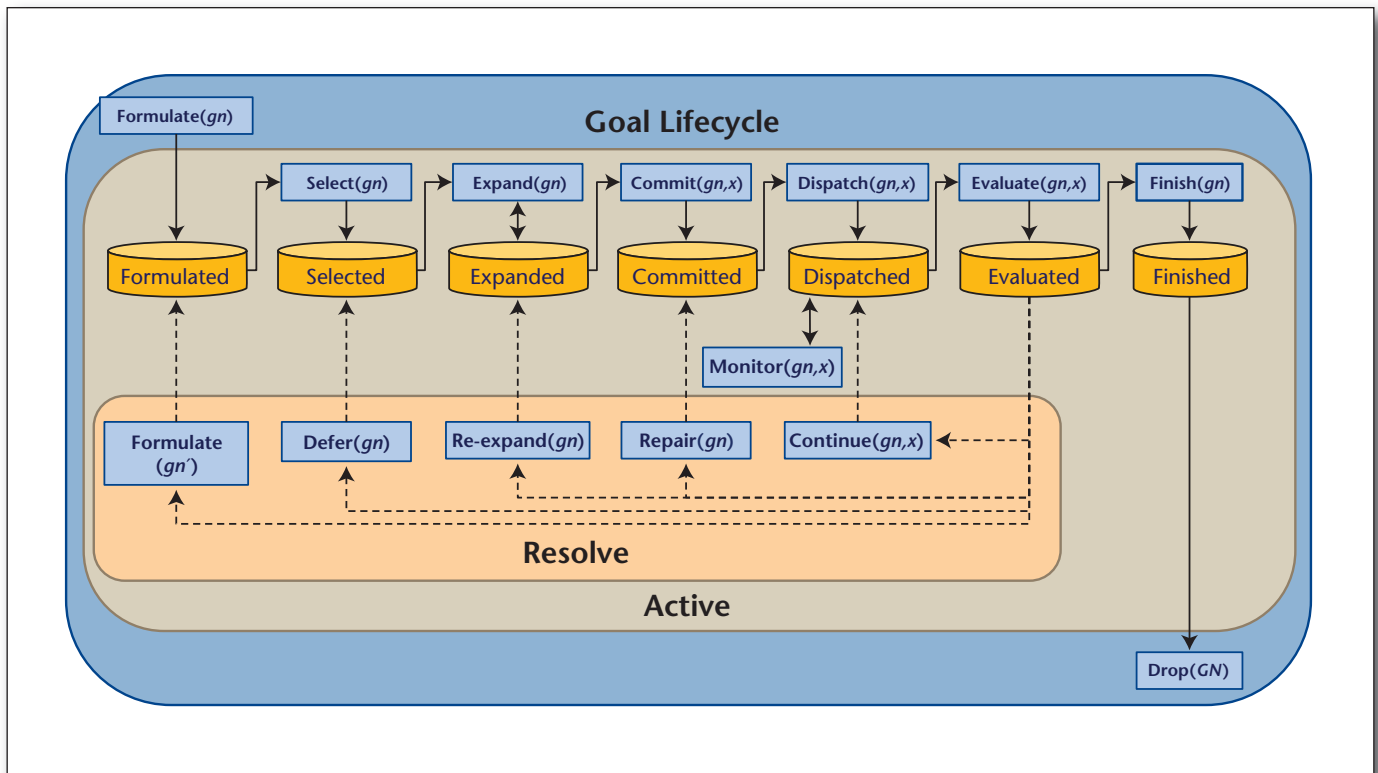


Figure 3. The Goal Lifecycle — A Goal Refinement Model of Goal Reasoning.

the GDA model to reason about sensing actions that have associated costs and about the way in which different methods for generating expectations impact a GDA model's performance.

While GDA can model simple GR processes, it does not explicitly model goal constraints, the relation of goals to tasks for achieving them, or processes for suspending or revising goals whose plans are not executable. This limitation motivated the development of a more comprehensive process model for GR. In Roberts et al. (2014), we introduced such a model, based on goal refinement, an extension of plan refinement (Kambhampati, Knoblock, and Yang 1995) that models the progressive refinement of goals through the addition of constraints. Goal refinement can represent the context in which a goal is pursued by a GR agent. Our goal refinement process model is the Goal Lifecycle. Figure 3 displays a simplified version of it. This model transitions a goal node (that is, a pairing  $GN = (g, C)$  of goal  $g$  with constraint set  $C$ ) through increasingly detailed modes (for example, formulated, selected) by applying constraint-refinement strategies that progress goal nodes toward completion. The strategies include formulate, select, expand, commit, and dispatch.

*Formulate* creates a new goal node and enters it into the Goal Lifecycle by defining its initial constraints, criteria, and prerequisites.

*Select* chooses which goal(s) to actively pursue; it ensures that the goals' prerequisites are met and that the agent has the resources to pursue them.

*Expand* generates a set of expansions  $X$  (for example, plans, decompositions of nonprimitive goals, or trajectories of primitive goals) to achieve a goal  $g$  in goal node  $GN$ , and a set of expectations for each.

*Commit* picks an expansion  $x \in X$  to pursue from those generated by *expand*.

*Dispatch* executes the committed expansion and defines the criteria by which  $g$  can be evaluated during execution.

The Goal Lifecycle also includes strategies for reacting to events and changes during execution. After being dispatched, a goal expansion is monitored and, if a discrepancy is detected, it can be evaluated. As a result, the GR agent may continue executing the expansion, it may drop  $GN$  (as either completed or failed), or it may try to resolve the discrepancy through one of several strategies (for example, repair, defer) that transition  $GN$  to an earlier mode before execution resumes. This approach supports goal adaptation, deferment, and even reformulation. The Goal Lifecycle captures decision points during a goal's activation, and can be represented as a set of *decide* subprocesses (figure 4) where this lifecycle's strategies subsume the decision processes denoted in figure 2, and we introduce a data struc-

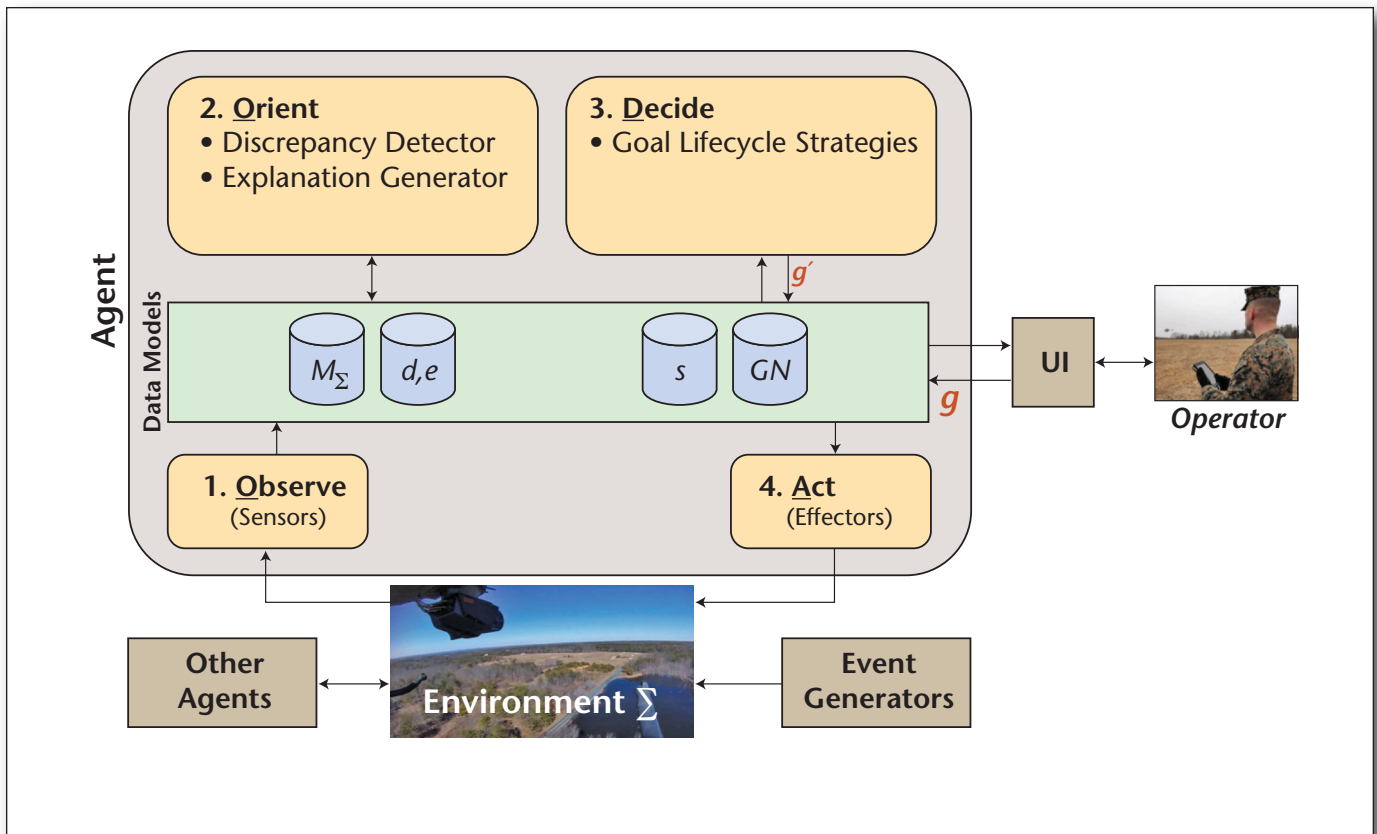


Figure 4. A Depiction of a Goal Lifecycle Model of Goal Reasoning.

ture (GN) that records substantial information associated with each goal node (for example, goal, associated constraints, mode, selected expansion/plan, plan expectation, associated discrepancies).

Goal refinement is only one extension of plan refinement, which equates multiple planning algorithms in plan-space and state-space planning. Other extensions incorporate different forms of planning and clarify issues in the Modal Truth Criterion (Kambhampati and Nau 1994). More recent formalisms (for example, Angelic Hierarchical Plans [Marthi, Russell, and Wolfe 2008] and Hierarchical Goal Networks [Shivashankar, Alford, and Aha 2017]) can also be viewed as leveraging plan refinement. Employing constraints in plan refinement allows a natural extension to the many integrated planning and scheduling systems that use constraints for temporal and resource reasoning.

Our Goal Lifecycle resembles the one proposed by Harland et al. (2014) for BDI agents, which they provide operational semantics for and demonstrate on a Mars rover scenario. Winikoff, Dastani, and van Riemsdijk (2010) have linked linear temporal logic (LTL) to the expression of goals. As described later, we have as well, though our work with Goal Reasoning with Information Measures (GRIM) focuses on agent teams rather than single agents.

In summary, the Goal Lifecycle provides a formal structure for goal refinement, such that the GR agent can deliberate on and adapt its goals in response to dynamic and unpredictable events. As described next, our GR agents employ variants of GDA or more comprehensive Goal Lifecycle models.

## Emerging Applications

To date, our GR applications have focused on controlling autonomous unmanned vehicles, either simulated or hardware. This section summarizes three such applications: the first employs the GDA model, the second uses a substantial modification of it, and the third instantiates the Goal Lifecycle.

### Underwater Vehicles

This section briefly summarizes initial studies on using GR to control an unmanned underwater vehicle (UUV). Details on the implementation and results can be found in Wilson et al. (2018).

UUVs can perform several important missions (for example, surveillance, mine countermeasures, plume source localization, hull inspection), which motivate a high demand for robust UUV control methods. These vehicles must operate in mission environments that, unlike others (for example, ground, air,

near-earth space), may prevent their communication with human operators for long durations (for example, weeks). Also, most UUV sensors (for example, acoustic sonar) cannot provide rich situation assessment information, which exacerbates the challenge of operating in an environment subject to several hazards (for example, fishnets, marine life, mines, uneven bathymetry, other vehicles) and effectors (for example, currents, sea state). This combination of factors motivates the investigation of intelligent control strategies, such as GR techniques, to assist with ensuring that a UUV operates competently when unexpected states arise.

While some studies apply GR agents to robotics tasks (for example, Cox et al. 2016), these agents have rarely been applied to control UUVs. Similarly, while many researchers use AI techniques for UUV control (for example, Cashmore et al. 2015; McMahon and Plaku 2016; Rajan and Py 2012), few employ GR techniques. One exception is the recent work by Oxenham and Green (2017), who use a GDA model (in MIDCA) for dynamic power management of a UUV's multicore processor during long-duration missions. Their initial work concerns simulated scenarios involving a discrete domain; in-water tests are planned in the future. In contrast, our GR agent, also based on GDA, performs deliberative mission management and can support multiple UUV missions.

We installed our agent in a dedicated single-core CPU aboard an OceanServer Iver2,<sup>5</sup> a low-cost, lightweight UUV (of diameter 14.7 centimeters, length 12 centimeters, and weight 19 kilograms, with a top speed of about 2.0 meters per second). The Iver2's Underwater Vehicle Console (UVC) processes raw sensor data. We use MOOS-IvP (Benjamin et al. 2010) to connect our agent with the UVC, and specifically iOceanServerComms.<sup>6</sup> MOOS-IvP receives updates from the UVC and posts them to MOOSDB, a centralized server that coordinates message passing among autonomy applications. Our agent receives updates to relevant variables in MOOSDB, from iOceanServerComms or MOOS-IvP simulation software, to populate its state. For actuation, our agent sends commands to IvP Helm, a reactive, behavior-based motion controller that sets navigation parameters to generate collision-free trajectories. IvP Helm uses an interval programming technique that performs multiobjective optimization over the active behaviors' objective functions in the navigation space (typically, heading, speed, and depth). To match the dynamics of MOOS-IvP's simulator with our vehicle, we use several default MOOS-IvP values (for example, buoyancy rate, maximum acceleration) that accurately capture the UUV's dynamics in field trials (McMahon and Plaku 2016).

Figure 5 displays our GR agent architecture, which is an adaptation of ARTUE (Molineaux, Klenk, and Aha 2010). Our agent employs a variant of the Planning Domain Definition Language (PDDL) for plan-

ning and explanation, based on features of PDDL+, which is a symbolic language for specifying planning domains with discrete actions (Coles and Coles 2014). It can represent and reason with continuous state values, exogenous events, and continuous processes, all of which are pertinent to the underwater environment.

Our agent's discrepancy detector compares expected with observed states (constructed from processed sensor data and simulated environmental elements). It compares them using set-difference for facts and value comparison with a floating-point tolerance for numerically valued fluents, using the same predicate-logic representation of states as our planner. To explain a discrepancy, it uses a C++ reimplementation of DiscoverHistory (Molineaux and Aha 2015). The goal manager formulates a goal using a simple rule-based system, where rules evaluate the current state and, when triggered, generate a priority value with each formulated goal. Our hierarchical task network planner is PHOBOS, which we designed to model expectations in noisy environments. PHOBOS generates fluent values by incorporating acceptable ranges of possible values (per action) so as to reduce false discrepancy rates (Wilson, McMahon, and Aha 2014). Finally, the goal manager simply ranks (by priority) all formulated goals, and the agent generates a plan for the highest-priority goal.

We have completed initial in-water demonstrations of our agent's in situ capability at NRL's Chesapeake Bay Detachment (CBD) facility. Our objective was to show that it can respond competently to basic maritime events. Our scenario involves reacting to unexpected observations during execution of a survey task that would typically be performed in mine countermeasures missions. We tasked the vehicle with three goals: (1) reach a start location (after departing a launch point), (2) complete the survey of a predefined region from that location, and (3) reach its launch point.<sup>7</sup> During this scenario, we also simulated the transit of an unexpected unmanned surface vehicle (USV) that emits engine noise and, in some cases, active sonar pings (indicating that it is searching for the UUV with hostile intent). Both noise and pings generate discrepancies, and for pings we defined as the UUV's appropriate response to pursue the goal of reaching a "safe point" (that is, retreat). Figure 6 depicts the UUV's area of operations, showing the survey region, the UUV's launch point, and the endpoint regions for the simulated USV.

For this scenario, the agent's PDDL domain description included (1) the vehicle's location, depth, speed, and heading; (2) notional processed input from passive sonar sensors (classified as "engine noise" or "active pings"); (3) actions for traversing to a waypoint and surveying a region (which causes the vehicle to execute "lawnmower" motion patterns); and (4) exogenous events for state changes as com-



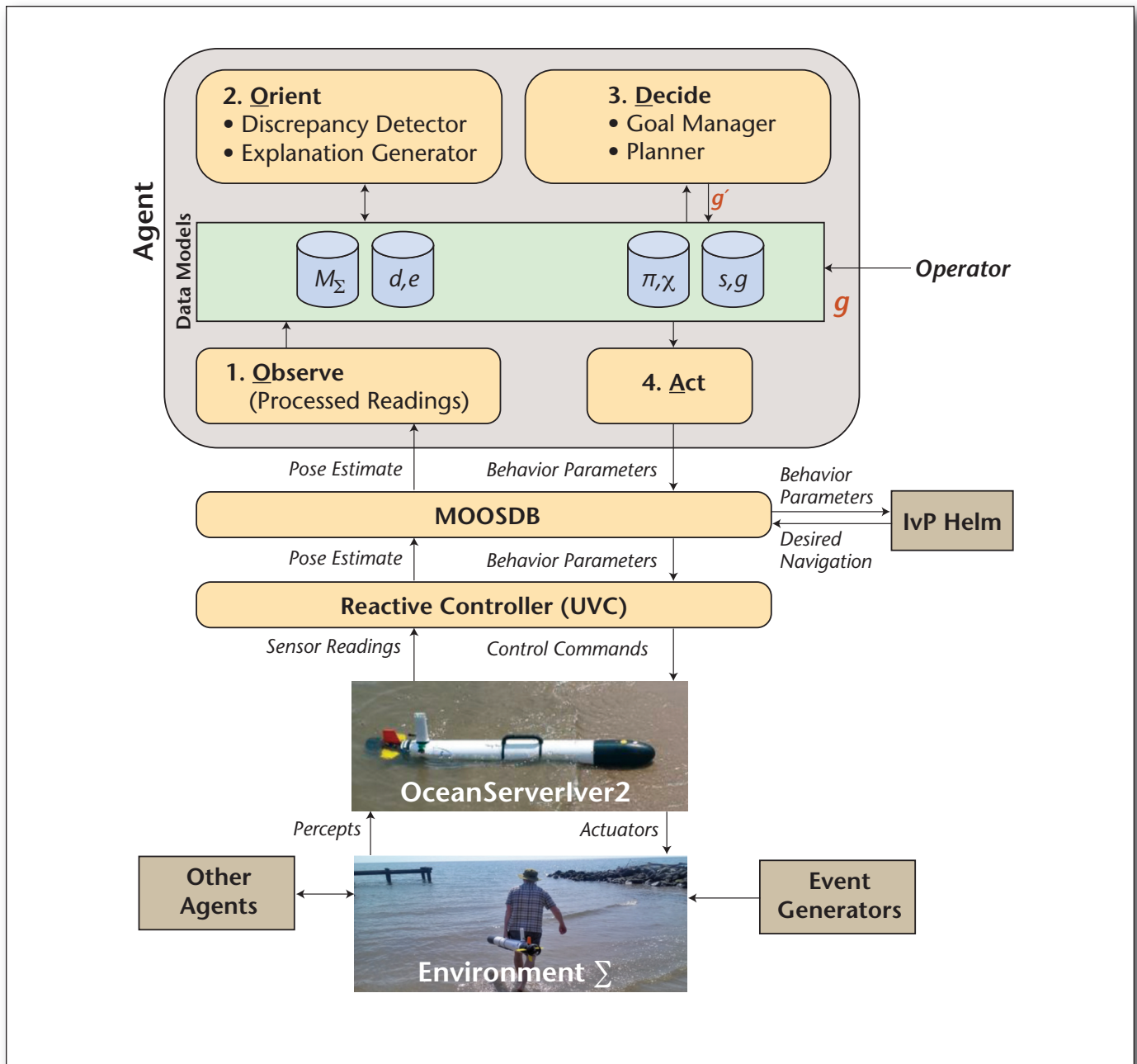


Figure 5. A GDA Model of Goal Reasoning for Unmanned Underwater Vehicle (UUV) Control.

municated by the UVC. These events may be anticipated by the planner or inferred during explanation generation.

We first tested our agent in simulated neutral and hostile mode scenarios, where only the latter included active sonar pings. The simulated USV starts in the center of the UUV's target survey region, picks a heading to one of the two endpoint regions, loiters for a specified time (to ensure the UUV encounters it), and then begins traversing. The USV emits an

engine noise with a detectable radius. Meanwhile, the UUV departs its launch point toward its survey region. It detects the unexpected engine noise and interprets it as a discrepancy that triggers the explanation generator to identify that a contact is within range, but without detecting a (hostile) ping, no new goal will be formulated. In contrast, when a ping is encountered, the explanation generator concludes that there is a hostile vehicle within range, and goal formulation recommends (with high priority) a goal

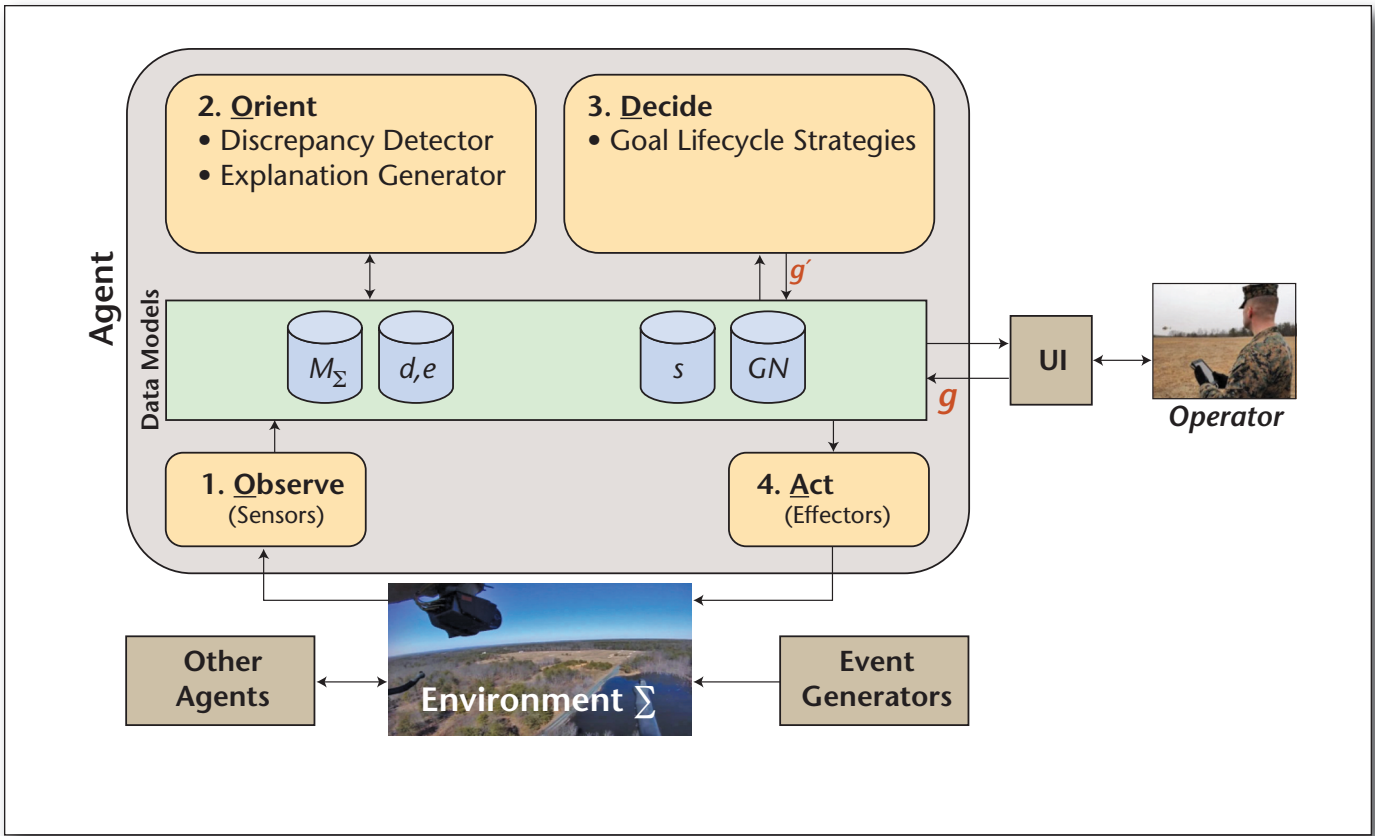


Figure 6. Overhead View of NRL's Chesapeake Bay Detachment (CBD) with Key Locations Highlighted Pertaining to Our In-Water Tests.

to retreat to the safe point. When the pinging is no longer detected, the explanation generator will conclude that there is no hostile vehicle in that region. At this time, goal generation directs the UUV to resume its prior goal, and it completes its mission. In 25 trials, in which the mode was randomly varied along with other independent variables (for example, the simulated USV's route), the UUV responded appropriately each time.

We then tested our agent at CBD using the same scenarios. From six trials we collected data with the UUV traversing at the sea surface or maintaining a depth of 0.75 meters. Equal numbers of trials were used for hostile and neutral USVs. In each trial, the UUV correctly detected the USV and its active pinging (for the hostile condition), and reacted by explaining the discrepancy and formulating the correct goal in response. Due to our calm marine environment, we did not encounter significant positional sensor drift during the relatively short mission. Generously modeling our expectations using PHOBOS's ranged values provided sufficient tolerance for the noise that we did observe in the surface and underwater trials. Figure 7 depicts traces from a hostile and nonhostile trial, respectively.

In summary, our GR agent can successfully control

an Iver2 UUV in at-sea tests in simple scenarios using only the limited computational resources typically available on UUV platforms. It can formulate goals and execute plans based on the user's input. It can recognize completion of prior goals, detect discrepancies, and formulate (and act on) new goals in response. Our future work plans include testing more advanced goal formulation techniques (Wilson, Molineaux, and Aha 2013), which will reduce the need for domain-specific knowledge. We will also integrate a more advanced motion path planner that can provide cost (that is, time, energy) estimates for plan execution (McMahon and Plaku 2016), which will allow our agent to make more informed goal selection decisions. Finally, we will explore more challenging scenarios that include noise in sensor models; maritime sources of sensor interference; real (rather than simulated) sensors; more advanced motion behavior (for example, vehicle tracking); UUV hardware faults; multiple surface vehicles; mine-like objects that can be discovered during surveys; and a larger space of goals (for example, for communicating events of interest to human operators, patrolling a region, or gathering information about encountered objects).



Figure 7. Traces from a Hostile and NonHostile Trial.

The left image depicts mission traces when a hostile USV's active sonar pings are encountered, in which case the UUV retreated to its safe point and circled until the USV departed, while the right image depicts a mission trace with a neutral or nonhostile USV.

### Beyond-Visual Range Air Combat

Unlike with UUVs, our work on unmanned air vehicles (UAVs) is in the context of human-agent teams that may include multiple vehicles, and it is focused on one mission type. We briefly summarize these efforts. Additional details can be found, for example, in Floyd, Karneeb, and Aha (2017) and Karneeb et al. (2018).

The United States Air Force (USAF) envisions future roles for autonomous platforms (for example, UAVs) (James and Welsh 2015), including in human-machine combat teaming operations. Among these is beyond-visual-range (BVR) air combat (or, simply, BVR), which is a modern form of air-to-air warfighting. In BVR, opposing teams of aircraft engage over large distances (that is, over 100 kilometers), where each team attempts to destroy their enemy (using active radar homing missiles with ranges of approximately 50 kilometers) or to force them to retreat. Similar to close-range dogfighting, BVR engagements can involve multiple aircraft (teammates and adversaries) operating in a contested airspace. However, BVR is less reactive and involves more deliberation, with positioning and timing being more important than motion planning. Still, the BVR environment is continuous, partially observable (due to limited sensor ranges), and noisy (due to sensor errors); and aircraft behaviors must satisfy tight real-time constraints to evade opponent attacks and avoid dangerous maneuvers (for example, flying too low, colliding with teammates). BVR scenarios also incur substantial uncertainty, as the adversary's assets (air and ground), configurations, and preferred tactics are not always known a priori. Finally, as BVR scenarios unfold, the battle situation can change rapidly, which can present opportunities and problems that encourage changes in mission objectives. Thus, controlling wingmen UAVs in future mixed human-UAV

BVR teams motivates the development of GR-controlled agents.

We studied this task in the context of two high-fidelity simulators, NGTS (NAVAIR 2013) and AFSIM (Clive et al. 2015), which are used for research and operations analysis and which include models of weapons, sensors, and communication networks. Few other applications of AI agents have been reported for this context. One exception is RIPR (Clive et al. 2015), which was designed with the assistance of subject-matter experts (that is, ex-fighter pilots). RIPR performs competently in BVR scenarios across all aspects of an encounter (for example, target pursuit, attacking, escaping danger). Another example is Alpha (Ernest et al. 2016), which employs tactics, represented as fuzzy trees, that are learned by genetic algorithms. Alpha performs well against expert pilots, and is designed as an AFSIM "red" team for training. Unlike our GR agent, RIPR and Alpha are not designed to deliberate about dynamic unexpected events. In research with low-fidelity simulators (that is, simple 2D environments with no sophisticated flight and aircraft models), genetic algorithms have been used to optimally assign targets to each aircraft (Luo et al. 2005) and to select initial team formations (Mulgund et al. 1998). These approaches support a subset of our agent's behaviors, they are performed only before a scenario starts, and they do not respond to changes in the environment or to unexpected opponent behavior.

Our GR agent, the tactical battle manager (TBM), is designed to control one or more UAVs in mixed human-UAV BVR mission scenarios, where the UAVs serve as wingmen to the human pilot. At the start of a scenario, the TBM receives a mission briefing containing information about its team, which is assumed to be correct, and the opponents, which may be incorrect. Team information includes team leader, capabilities (for example, each teammate's air-

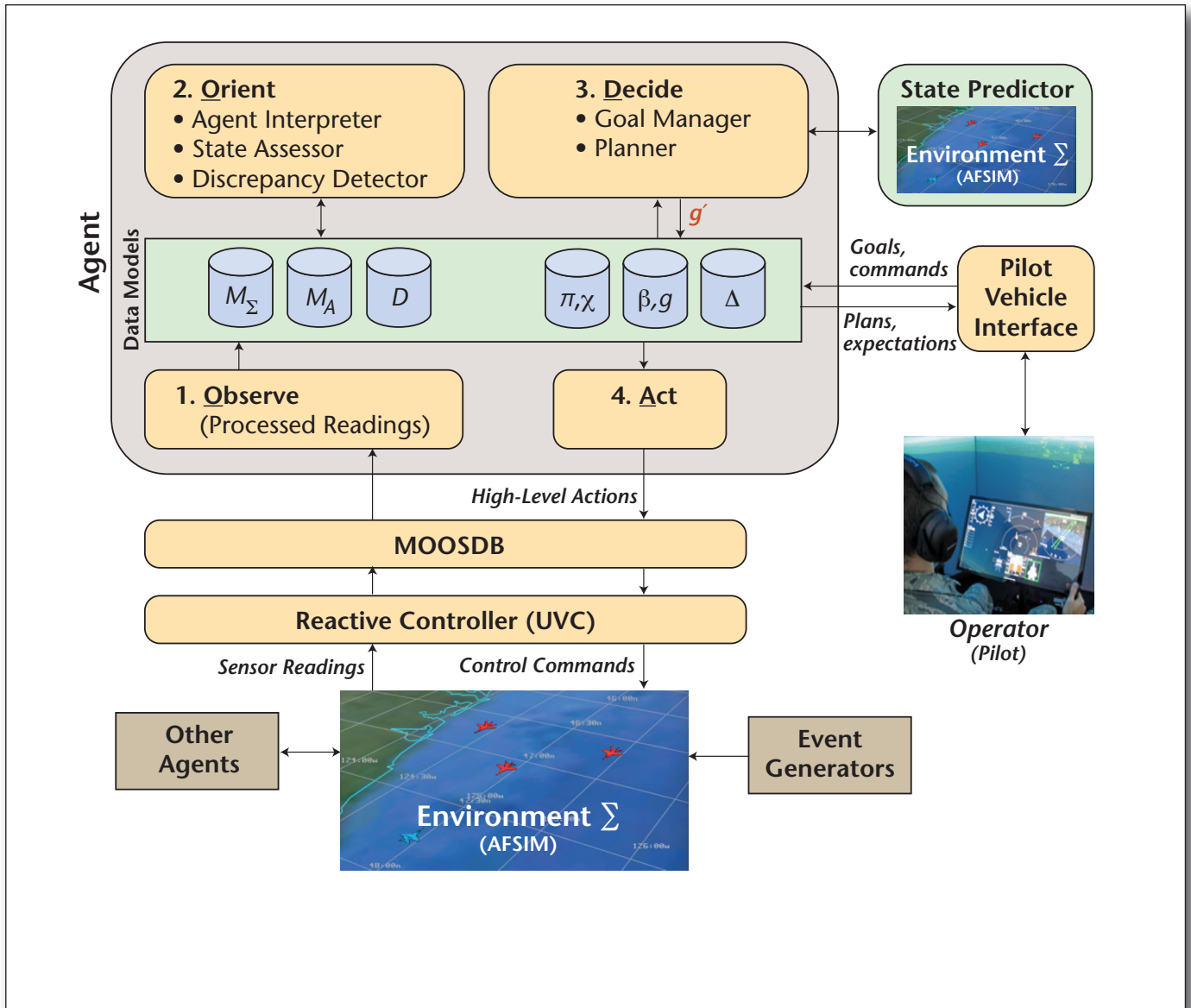


Figure 8. A Modified GDA Model of Goal Reasoning for Controlling a UAV Wingman in a Mixed Human / UAV Beyond-Visual-Range (BVR) Air Combat Team.

craft type, missiles), tactics (for example, preferred altitudes for engagement, preferred approach angles), speed (for example, passive, approach, engagement, escape), and weapons (for example, distances at which missiles are expected to hit, distance from an opponent that is considered dangerous). Information about the opponents includes their number of aircraft, aircraft type, and weapons capabilities. Many challenging situations can arise during these scenarios. For example, the human pilot may become unavailable (for example, distracted or departed), the hostiles' capabilities may exceed expectations, or additional friendly assets may become available.

To address these challenges, the TBM integrates several components that can (in parallel) access, cre-

ate, and modify information in the shared data models (figure 8), where  $\beta$  is the current belief state. This design allows the components to process information in real time and avoids delays caused by slower components. The TBM's agent interpreter recognizes the behaviors of hostile aircraft using its models of them ( $M_A$ ). We showed that a case-based reasoning technique (Borck et al. 2015), which uses features that model an agent's prior behaviors, performs well on this task. We later extended it with an active planner that "teases out" a hostile's behavior (to disambiguate it from others) and showed that this active planner further improves performance (Alford et al. 2015). After belief revision takes place (not shown in figure 8), the TBM's state assessor calculates the

degree to which the current beliefs satisfy the desires data model ( $\Delta$ ), where example desires are to maintain the TBM's safety or disrupt hostile movements. Inspired by BDI theory (Rao and Georgeff 1991), these (criticality-weighted) desires ( $\Delta = \langle \Delta_1, \dots, \Delta_m \rangle$ ) are functions that map the current sensor stream to  $[0,1]$ , where higher values indicate higher satisfaction. The TBM represents a goal as a set of preferred desire values  $g = \langle pref_1, \dots, pref_m \rangle$ , and it attempts to achieve environment states that satisfy its active goal's desires. The TBM's discrepancy detector tests the following discrepancies ( $D$ ), some of which refer to preferred desires: *Incoming Missile*; *Model Changed*; *Flanking Hostile*; *Expectations Violated*; *Out of Ammo*; *Low on Fuel*; and *Opportunistic Target* (Karneeb et al. 2018). For example, *Incoming Missile* identifies unexpected hostile missiles (which allows the TBM to dynamically respond to an attack and attempt to evade the missile), *Low on Fuel* tests whether that resource is running low, and *Expectations Violated* tests for violations of any of the current plan's expectations, which are generated by the state predictor.

The TBM's goal manager will change the active goal upon receiving a command from the human pilot or when it generates a new goal. The latter can occur when the TBM successfully achieves its current goal, when it predicts it will fail to achieve it (for example, when unanticipated hostile behaviors are expected to thwart the current plan), or when it recognizes an opportunistic situation that will benefit its mission (for example, a more appropriate target). The TBM continuously monitors for discrepancies  $D$  and uses a set of rules to determine whether a goal change is warranted. If so, the planner takes as input  $\beta$ , the new goal, the environment model  $M_{\Sigma}$ , and the recognized plans of other entities to generate a new plan to execute. Ours is a plan library planner (Borajao, Roubicková, and Serina 2015) that uses a library of ungrounded plan templates, provided by BVR domain experts, to represent desirable air combat tactics. The planner generates instantiations of applicable plans, abiding by constraints defined in the mission briefing (for example, maximum attack speed, attack angle) or resource availability (for example, number of remaining missiles). To evaluate each candidate plan, the TBM predicts its outcome using the state predictor (which uses AFSIM), calculates how well each plan achieves each desire, and selects the plan that is predicted to best achieve the TBM's current goal. For some discrepancies, the TBM invokes the planner to find a new plan to satisfy the current goal. If one is found, then this replanning obviates the need for goal formulation.

We conducted several simulation studies to test the TBM's capabilities, including the component ablation study summarized here. We wanted to test whether the TBM can outperform RIPR and whether each key TBM component contributes positively to overall mission performance. To do this, we aug-

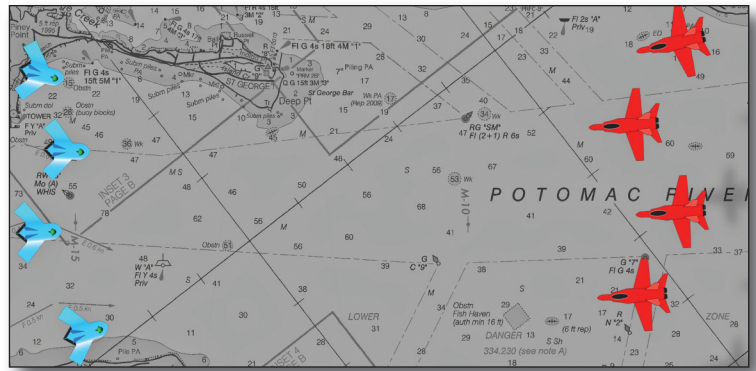


Figure 9. Starting Conditions for Testing the TBM GR Agent's Team (blue) versus a Team Controlled by a Different Agent (red) in a Constrained Random 4v4 Scenario.

AFSIM screenshot; aircraft size is not to scale.

mented a CMASI message set (Duquette 2011) to enable communication between the TBM and AFSIM's reactive UAV controller. We developed TBM ablations where we replaced one component with a simplified (but still functional) version (for example, the default discrepancy detector detects only *Incoming Missile*). We defined constrained random 4v4 scenarios (figure 9), where the TBM controlled the blue team, and RIPR or an ablated TBM variant controlled the red team. In each scenario, each team is aligned in a column with opposing teams spaced 4.5 times the distance among team members. In 100 random scenarios, each aircraft's 2D position was modified by a small amount, and the scenarios were repeated with teams in switched positions. Each of the 200 trials ended when one team was completely destroyed (a "win") or 20 minutes elapsed. For this study, we created a representation for the weapon engagement zone (WEZ) (for example, flight characteristics of a missile at different altitudes and flight durations, and aircraft rate of escape). We used a tree-induction algorithm to learn weapon models from observations, and applied them to facilitate more accurate planning (of controlled UAVs) and prediction of hostile asset behavior. We used the WEZ to calculate the expected range for a missile strike and to reason about future hostile engagements (for example, if the TBM's UAV is flanking, and the targeted hostile has an inferior WEZ model, then engage it at a closer distance). We also created a model of several cooperative tactics (for example, decoy, scatter, bracket, grinder) that the TBM's planner could select to accomplish a given goal.

Figure 10 summarizes the results of the ablations in the 4x4 scenarios. The TBM significantly outperformed its ablations (single-tailed  $t$ -test,  $p < 0.01$ ), where the average increase in win percentage varied between 58 and 154 percent. For example, versus a fully-ablated version (All Ablations), the TBM won 127 times and lost 50 times.

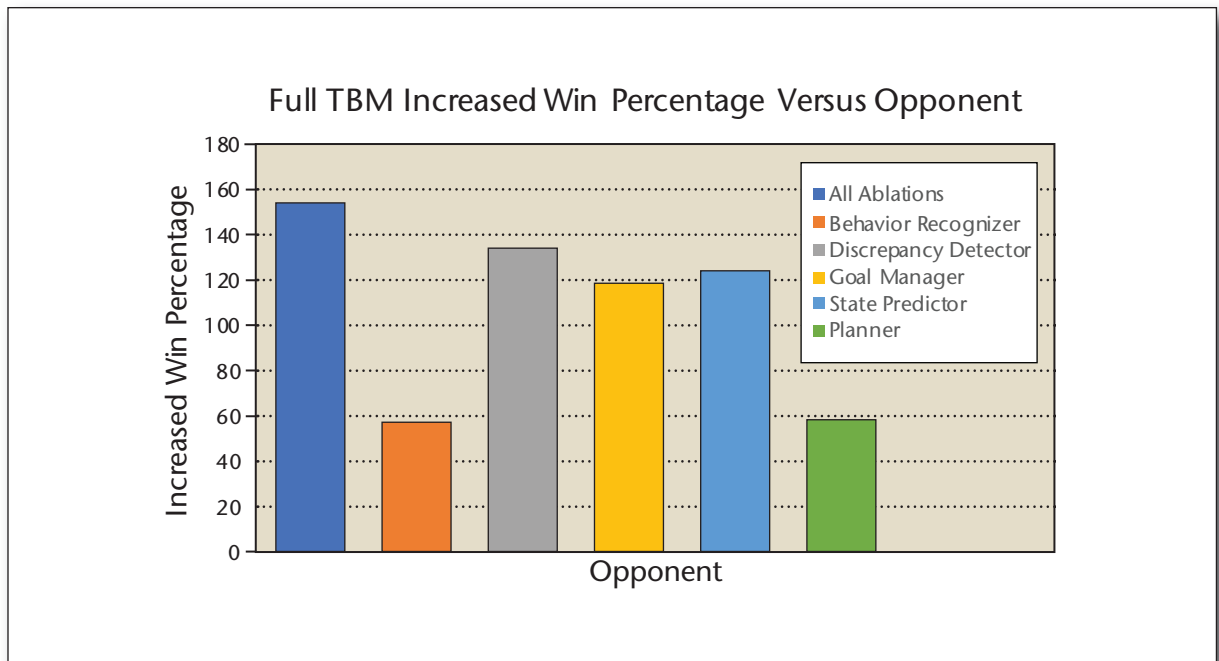


Figure 10. The TBM Outperformed Its Ablations in 4v4 Scenarios

50 times. Performance improvement versus RIPR (not shown in the figure) was more dramatic: 3200 percent. The TBM won 165 times and lost only 5 times. This result suggests that the TBM was performing fairly well and that its components were useful, at least for these scenarios. The TBM changed its goals approximately five times more frequently than when the default discrepancy detector was used, and the most frequent discrepancies for triggering goal formulation were (in order) *Expectations Violated* (due to a plan prediction failure), *Opportunistic Target* (due to opportunities to switch to a more viable target), and *Flanking Hostile*.

We assisted AFRL with integrating the TBM with their pilot-vehicle interface, which permitted testing the TBM with subjects (fighter pilots) in BVR scenarios using AFSIM. Studies were run with each pilot leading a team of one to three TBM-controlled UAVs. Afterward, pilots were asked to assess the TBM's impact on mission performance and its usability. They reported that the TBM reduced their workload and helped them complete their missions, and they generally had an overall positive assessment of TBM-controlled teammates.

In summary, dynamically modifying its own goals allowed the TBM to respond to unanticipated events while relaxing, to some extent, reliance on pre-coded/instantiated plans for all possible contingencies. The TBM combines techniques from GR, automated planning, opponent behavior recognition, state prediction, and discrepancy detection. Our empirical study demonstrated that it significantly

outperforms an expert-authored BVR agent in a set of combat scenarios and that each reasoning component positively influenced mission performance. System performance required real-time execution of all components, which the TBM supports. Future work should focus on integrating learning techniques into the discrepancy detection process. As one example of such work, the TBM should dynamically learn models of opponent aircraft and missiles (for example, using a strategy similar to the one described in Molineaux and Aha [2014]) and use them to detect novel hardware configurations (for example, types of aircraft or more advanced missiles). We also plan to add capabilities that identify opportunistic targets and communicate with other UAVs to more effectively perform small-team tactics (for example, surround the opponent, create a diversion).

### Foreign Disaster Response

I next summarize our application of GR to foreign disaster response (FDR) mission scenarios, which focuses on the centralized control of a set of agents using an instantiation of the Goal Lifecycle. Additional details can be found in Roberts et al. (2015), Apker, Johnson, and Humphrey (2016), and Johnson et al. (2016).

An FDR mission's objective is to provide, across the globe, humanitarian aid after a natural disaster strike, when many lives can be in peril and first responders must react quickly (DOD 2011). These missions can benefit from a heterogeneous team of UAVs and unmanned ground vehicles (UGVs) that rapidly sur-

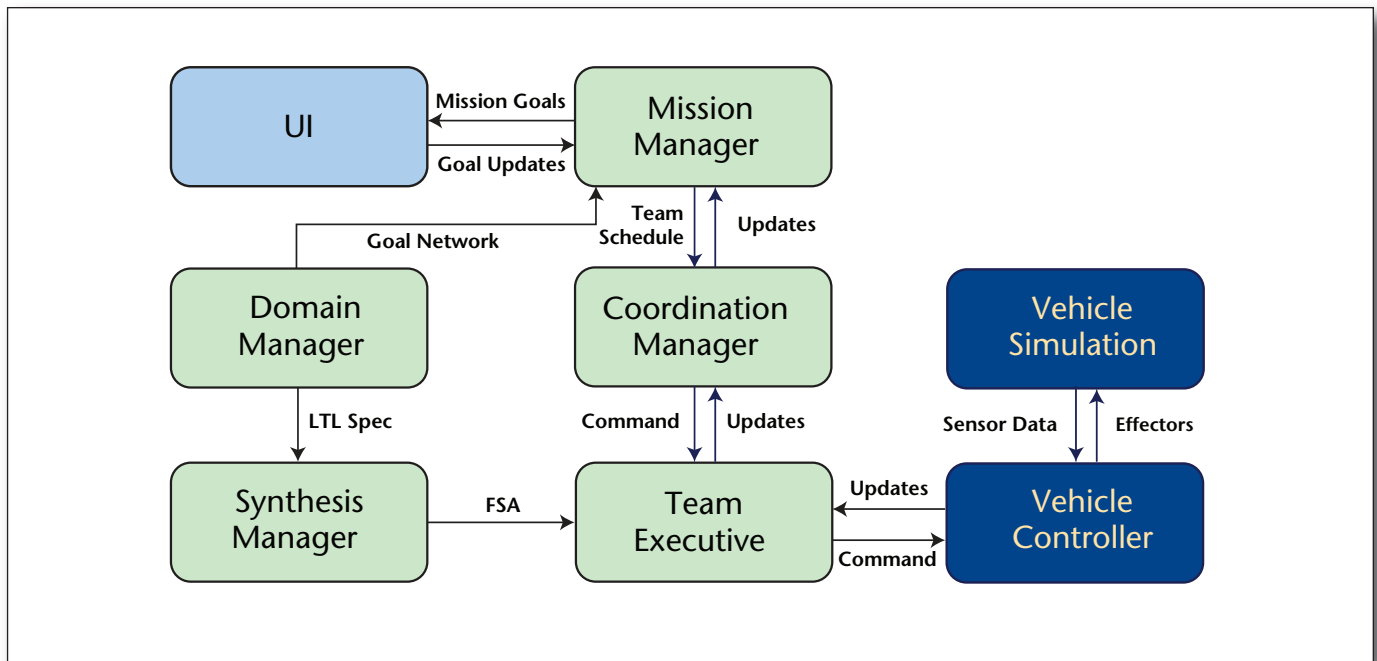


Figure 11. Conceptual Design of the Situated Decision Process (SDP).

The Mission Manager performs goal reasoning, creating a schedule of actions for a team of vehicles, each of which executes a synthesized FSA.

vey the area, that identify key locations (for example, of survivors, damaged infrastructure) and traversable routes for ingress and egress, that locate VIPs, and that serve as mobile communication relays. Human coordination of such a team can be challenging, as commands must be translated to actions by appropriate team members, the team must keep the human operator informed, and the robots must react intelligently to changes in their environment (for example, unexpected situations or events) or their internal state. These dynamic conditions may cause them to change their tasks or even their objectives (for example, if the current one is unachievable).

To address this problem, we designed the situated decision process (SDP), which (under operator guidance) uses a GR approach to control and coordinate a robot team (that is, managing and executing their goals) and which is designed for use by a forward operating base. The SDP uses a centralized control approach that provides commands to independent vehicles. The SDP (figure 11) takes as input an operator's commands (that is, goal updates with constraints) and passes them to the mission manager, whose GR agent (informed by our subject matter expert's goal network, encoded in the domain manager) selects a goal<sup>8</sup>  $g$ , expands  $g$  (that is, generates plans to achieve  $g$ ), commits to one such plan  $\pi_g$  (with expectations  $\chi_g$ ), creates a schedule for executing  $\pi_g$ , and passes it to the coordination manager. This module interprets the schedule and passes applicable com-

mands to a team executive, which assigns the commands to each vehicle. Each vehicle interprets their command as an input to a finite state automaton (FSA), which the synthesis manager automatically synthesizes from a variant of an LTL specification, called *general reactivity 1* (Bloem et al. 2012), and is guaranteed to satisfy it (Kress-Gazit et al. 2009). The FSA specifies the regions for executing the behaviors and mission sensors that cause a behavior change when a vehicle observes a notable event. This strategy yields a play-calling architecture (Apker et al. 2016) that provides guarantees on the execution of goals chosen by the mission manager. Health sensors establish the conditions before a vehicle can pursue a goal (for example, it requires sufficient fuel to reach a goal location), while contingency behaviors ensure that it maintains a safe posture (for example, landing an air vehicle) or attends to the health sensor (for example, returns to a base station to refuel). In summary, the SDP adjusts its goals autonomously and pursues them while constrained to abide by specific guarantees.

Our group implemented and applied, in simulated FDR scenarios, a variant of the SDP named Goal Reasoning with Information Measures (GRIM),<sup>9</sup> which employs a single measure to assess goal utility and communicates this utility to an operator (Johnson et al. 2016). GRIM controls a team of two vehicles that must survey a set of regions to locate a VIP and establish communications (figure 12). Each region (an airport and two office buildings) corresponds to a sur-

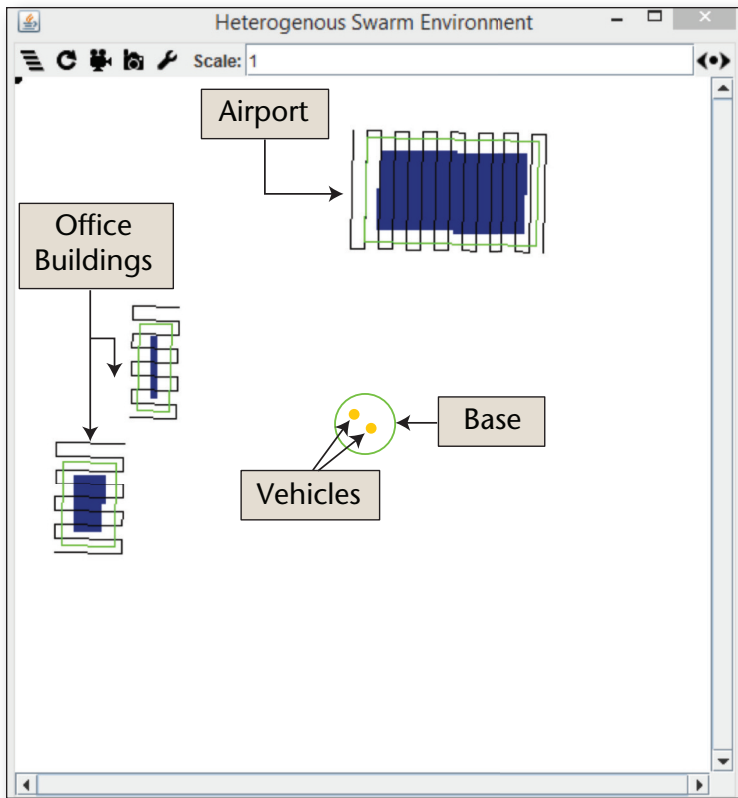


Figure 12. Scenario for Testing GRIM's Ability to Control Two Vehicles in a Simulated Foreign Disaster Relief Operation.

The vehicles' goals include completing a survey of the airport and office buildings, and establishing a communications relay for any VIPs found.

vey goal, where surveys follow a waypoint sequence. The remaining uncertainty in an area survey is the length of the search pattern that has yet to be traversed by an assigned vehicle.

Figure 13 shows the plots of the four Goal Lifecycle strategies. Graph 13a depicts that each survey goal is formulated by generating constraints on the maximum allowable uncertainty over time. GRIM selects a single goal (the airport survey goal) to pursue and expands it (that is, generates plans to achieve it). These plans' expectations (depicted as a change in the uncertainty over time) are shown in graph 13b. GRIM commits to a single expansion and dispatches it to the vehicles. Graph 13c depicts the corresponding expectations and performance bounds that are generated by the dispatch strategy. Graph 13d displays execution performance over time, as obtained by the monitor strategy. During execution, when the vehicle's performance is predicted to violate a goal constraint (for example, in graph 13d, when its execution reaches the worst-case time bound), GRIM triggers the evaluate strategy to determine what violation occurred. If the execution satisfies the completion criteria, the goal is marked as completed and

dropped. If it instead violates the goal's constraints, it is marked as failed and dropped. Otherwise, if the performance violates the execution bounds, a resolve strategy is activated to adjust the goal (for example, its expansion) before continuing execution. The selected resolve strategy can transition the goal back to an earlier Goal Lifecycle mode (see figure 3) (for example, it may repair the committed expansion by adjusting parameters that affect the expectations and bounds). Alternately, a resolve strategy may force GRIM to expand the goal again and then commit to and dispatch one of the new expansions.

We conducted an ablation study with GRIM's resolve strategies (Johnson et al. 2016) on simulated FDR scenarios. We found that they allow GRIM to perform GR during execution, that they improve its performance, and that they enable it to successfully complete more goals under uncertain and changing conditions. By associating the Goal Lifecycle strategies with a single measure, GRIM can define clear decision points that increase the transparency of its decision process. For an agent that can change its goals and plans, transparency in how those decisions are made is critical for promoting operator trust.

GRIM automatically synthesizes FSAs whose execution by individual vehicles is guaranteed to satisfy their LTL specification. Balch et al. (2006) also use FSAs for mobile robot guidance. Hand-coding an FSA for each execution of a robot is tedious and error prone. Kress-Gazit, Fainekos, and Pappas (2009) instead synthesize an FSA from an LTL specification using a game-theoretic approach in which the robot acts to achieve its goals versus actions taken by an adversary. This strategy guarantees correct behavior if the LTL specification is never violated, but synthesis is quadratic in the number of goals (Bloem et al. 2012) and is thus intractable for large robot teams. GRIM instead preselects missions for vehicles prior to FSA synthesis, which reduces the size of the LTL specification and the computation time required for synthesis.

Goal Lifecycle strategies are themselves important research topics, and each can be accomplished using a variety of algorithms. For example, the goal selection method can vary widely, from domain-specific rule-based selection (Thangarajah et al. 2010) to the evaluation of domain-independent heuristics (Wilson, Molineaux, and Aha 2013), or goal priorities (Young and Hawes 2012). Many planning algorithms can be used for goal expansion, including the sophisticated hierarchical (Shivashankar, Alford, and Aha 2017) and temporal planners (To et al. 2017) that our group developed and plan to integrate with future GR agents. Finally, in many cases the planner can generate plan execution expectations, but in some situations additional simulations and deliberation may be required (Auslander et al. 2015).

Future extensions of GRIM will investigate additional goal types (for example, a communications



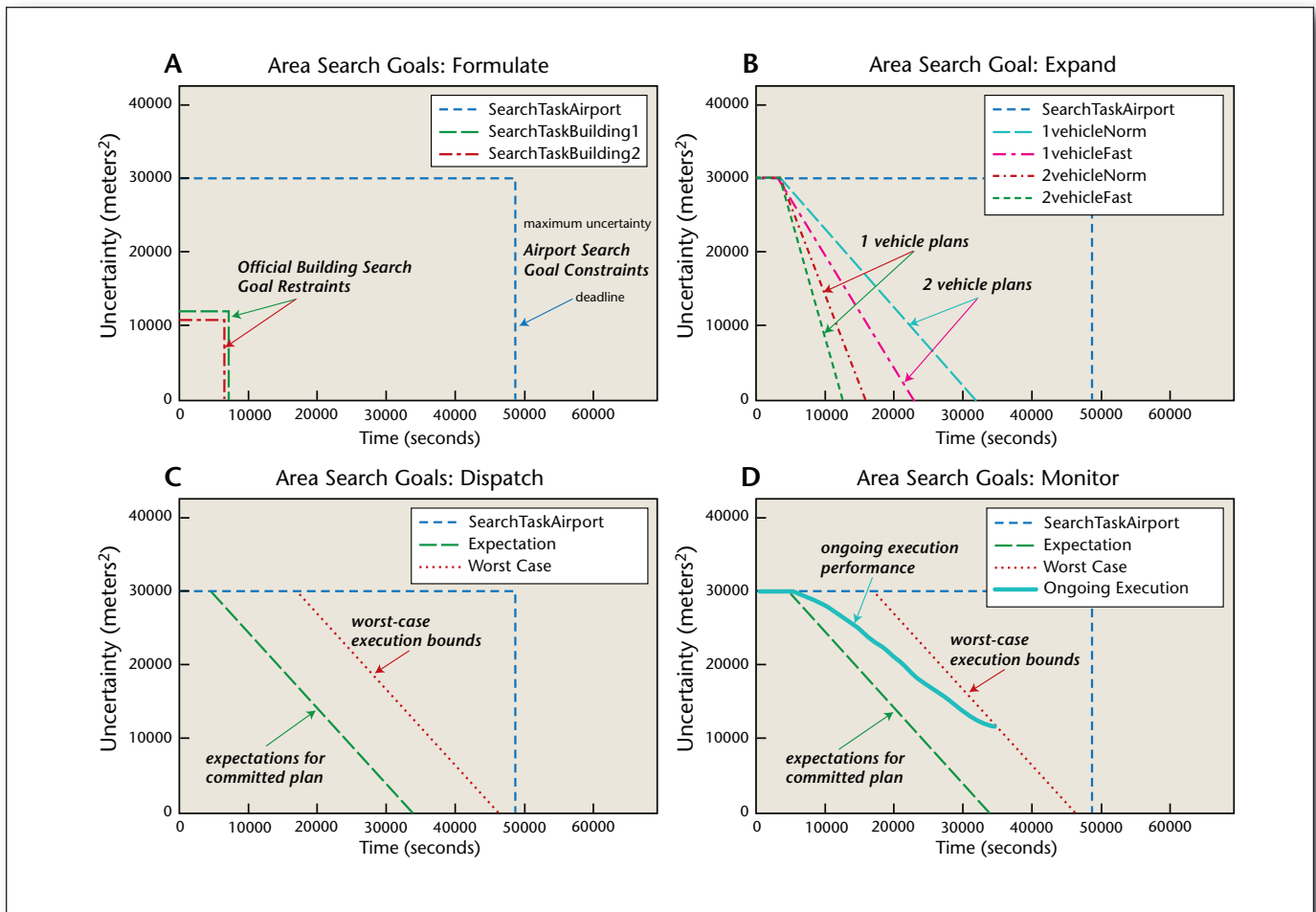


Figure 13. Plots of Four Goal Lifecycle Strategies from the Execution of GRIM on Simulated FDR Scenarios.

relay for a discovered VIP) and more comprehensive algorithms and information measures. The uncertainty of an area survey was approximated, for expectations and progress of the goal, by the length of the lawnmower search pattern. A more accurate measure of uncertainty (for example, tracking the total area covered by the vehicle's sensors) and corresponding expectations in an area survey would allow GRIM to improve its performance estimates and react accordingly. Also, a more intelligent goal selection approach should consult with the expansion strategy on the likelihood of discovering a VIP in each region. Likewise, adapting plan expectations (for example, recognizing that the vehicles are not completing the survey at the expected rate, and changing the expectations accordingly) would enable GRIM to more quickly identify and evaluate problems, and thus improve the likelihood that it could resolve any discrepancies. Finally, our studies to date have been in simulation; we intend to test extensions of GRIM in robotics scenarios in the future.

## Prospects

Many interesting research issues deserve attention in the study of GR agents. Our group has made some progress, as described below. Information on other relevant work has been reported, for example, at the Goal Reasoning Workshop series<sup>10</sup> and in a recent special issue in *AI Communications*.

While our focus has been on autonomous vehicle control, GR can be a foundation for (unembodied) proactive decision aids in collaborative decision-making contexts. One compelling example is in support of a military command staff, where the agent could monitor an ongoing mission's status (for example, the area of operations, the status of friendly and other known assets, the probability of achieving the mission objectives), observe interactions among staff members as they consider alternative courses of action, provide information when prompted, formulate its own recommendations, and recognize when to share these with the staff. These capabilities could

also help civilian groups, such as those responsible for marketing, financial investment, and budgeting decisions.

A more sophisticated GR agent would be required to assist these human-agent teams. For example, it would need to perform automated scene understanding and situation assessment. Although we integrated deep learning (for image recognition) (Bonanno et al. 2016) and natural language understanding techniques (Gillespie et al. 2015) with GR agents, these have been in constrained settings. Our GR agent would also need to detect and reason about observations not anticipated by its action, event, or agent models. For this purpose, we developed DiscoverHistory, which performs continuous explanation generation by using heuristic constraints to search the space of plausible explanations (Molineaux and Aha 2015). Event models can be learned from these explanations and applied to more quickly reason about future similar occurrences (Molineaux and Aha 2014). The actions of agents in the environment (that is, team members and others) will need to be monitored by our GR agent to help identify their goals and plans. We developed a plan recognition algorithm for noisy environments (Vattam and Aha 2015), but this algorithm requires extension to dynamically recognize changes to an observed agent's plans. Our agent would also benefit from algorithms that reason with mental models to infer information about a given situation (Khemlani, Hinterecker, and Johnson-Laird 2017). We will soon integrate, in a GR agent, an extension of this approach that can reason with qualitative spatial representations (that is, the relation of objects in a scene).

Our agent would also need extensions for more robust decision making. For example, our GR agents that instantiate the Goal Lifecycle assume there exists only a single goal node, and only one algorithm for each of its strategies (for example, selection, expansion). This approach is limiting: we would like our agents to compare the utilities of different goal nodes before selecting which one to process, and not necessarily discard ones that are not immediately selected. Also, different algorithms for a lifecycle strategy (for example, planners for the expansion strategy) may be appropriate for different problem-solving contexts and could be made available to a GR agent for selection. Thus, we are investigating a metareasoning method for selecting a goal node (or nodes) and a strategy algorithm (or algorithms) to apply. We also seek methods that reason about how to proceed if a goal cannot be processed in its current form (for example, can more specific versions of it be processed that would satisfy a human operator's intent?). This goal-changing behavior may be accomplished by incorporating a process that can transform unachievable goals into ones that can be pursued by a GR agent (Cox, Dannenhauer, and Kondrakunta 2017). This process must learn a model for determin-

ing when to perform goal transformation and which transformation operator to apply in a given context. Although we have collaboratively investigated learning algorithms for goal selection (for example, Powell, Molineaux, and Aha 2011; Jaidee, Muñoz-Avila, and Aha 2013), learning goal transformation knowledge has received less attention.

Finally, a GR agent that serves as a proactive intelligent decision aid must be able to explain its models, its reasoning for a recommendation (and other decisions), and the expected outcomes from applying a recommendation to its human teammates. That is, a GR agent should be *transparent* so that its teammates can calibrate their trust in it and, in doing so, make appropriate decisions. Indeed, Selkowitz, Lakhmani, and Chen (2017) reported that a user interface designed to expose a GR agent's models and reasoning can increase an operator's situation awareness and trust in the agent's decision making. These findings relate to research on Explainable AI (XAI),<sup>11</sup> which has focused narrowly on machine learning techniques and more broadly on AI (Aha et al. 2017). In future work, we will develop and assess the utility of explainable GR agents in human-agent teaming contexts.

Most GR agents accept commands from a human operator and do not deviate from these commands unless they are in a fully autonomous mode (for example, when the operator is unavailable for consultation), at which time they may consider alternative objectives so long as they are constrained by operator intent. However, there is need for more proactive agents that can object to, or even reject, an operator-provided command when the operator is accessible. Such rebel agents may rebel for several reasons. For example, they may have access to information that is unavailable to the operator which indicates that a current action, plan, or objective will fail; they may recognize that an operator (or others) is in an unsafe situation; they may object due to an ethical dilemma; or they may prioritize another objective more highly for reasons of social justice. Acting rebelliously in these contexts can impact operator trust in the GR agent, either positively or negatively. If designed appropriately, there are substantial benefits for modeling rebel agents, both to assist human operators directly and to notify operators about the predicted rebellious behavior of other agents (that is, that may act contrary to the operators' objectives). Our group developed a framework for AI rebellion, identified its stages, described factors that motivate or support it (Aha and Coman 2017), and discussed how social rebel agents can benefit by producing and using alternative narratives to justify their rebellion (Coman and Aha 2017). Our current work includes demonstrating the utility of agent rebellion in a variety of mission scenarios.

GR is inherently dangerous if not properly constrained. How can we ensure that an agent's decisions

to change its objectives, plans, or actions are consistent with operator/command intent? Guarantees are required that a GR agent will operate correctly, both when an operator provides direct oversight and when it acts autonomously, which is complicated when operating in complex environments where not all situations can be predicted. This complexity is exacerbated if these agents use online learning techniques to acquire or refine models of their environment and other agents. Related to the motivations for and study of safe AI (Vassev 2016; Omohundro 2014), additional research is needed to develop best practices for safe goal reasoning so that GR agents can be deployed confidently as productive and appropriately trusted partners in human-agent teams and in autonomous settings.

## Conclusion

Intelligent agents that dynamically deliberate on, reprioritize, and self-select their goals have a long history of study (for example, Norman and Long 1996; Altmann and Trafton, 2002; Cox, 2007; Talamadupula et al. 2010; Thangarajah et al. 2010; Weber, Mateas, and Jhala 2012; Jaidee, Muñoz-Avila, and Aha 2013; Klenk, Molineaux, and Aha 2013; Harland et al. 2014; Dannenhauer and Muñoz-Avila 2015; Cox, Dannenhauer, and Kondrakunta 2017). Researchers studying these types of agents are motivated by the challenges of deploying them in complex environments, including to serve as members of human-agent teams. Our group refers to these as goal reasoning (GR) agents, and in this article I described some of our inspirations, foundations, and emerging applications. Although few applications of these agents exist, demand for them should increase because GR can serve as the foundation of highly autonomous and proactive approaches for vehicle control and intelligent decision aids. Many important research directions on GR require further attention, in addition to those I noted previously. These include, for example, representing and reasoning with additional goal types (van Riemsdijk, Dastani, and Winikoff 2008), dynamically recognizing other agents' goals (Vered and Kaminka 2017), recognizing team intent (Franke et al. 2000), and methods for learning goal priorities (Young and Hawes 2012). Suitably constrained GR agents have tremendous potential for applications of critical interest, but the task of designing and developing them is AI-complete (Shapiro 1992), as such agents must perform comprehensive situation assessment and decision-making tasks. For this reason I encourage AI researchers to consider how their work relates to GR, and to contribute to this interesting topic.

## Acknowledgements

This article is based on my Robert S. Engelmore Memorial Lecture, given at IAAI 2017 in honor of

Engelmore's extraordinary service to AAAI and contributions to applied AI. I did not survey the broader topic of GR, including many contributions from, for example, Daniel Borrajo, Nick Hawes, Tom Hinrichs, Gal Kaminka, Mary Lou Maher, and Okun Topçu. For more information, please see, as a start, the 2018 *AI Communications* special issue on goal reasoning and the proceedings from GR workshops held at AAAI-10, ACS-13, ACS-15, IJCAI-16, and IJCAI-17.

Thanks to the many colleagues who have contributed to our group's work, including Ron Alford, Tom Apker, Bryan Auslander, Dave Bonanno, Hayley Borck, Dongkyu Choi, Alexandra Coman, Dustin Dannenhauer, Michael Floyd, Keith Frazer, Kellen Gillespie, Brian Houston, Ulit Jaidee, Ben Johnson, Justin Karneeb, Matt Klenk, Michael Leece, Michael Maynard, Jim McMahan, David Menager, Matt Molineaux, Phil Moore, Héctor Muñoz-Avila, Jay Powell, Mak Roberts, Vikas Shivashankar, Christine Task, Son To, Swaroop Vattam, Mark Wilson, and Artur Wolek. Thanks also to our sponsors (AFOSR, DARPA, NRL, ONR, OSD ASD (R&E)), with special thanks to Michael Cox for steering us toward this topic and to AAAI for providing this opportunity.

## Notes

1. Although I use an OODA loop ([pogoarchives.org/m/dni/john\\_boyd\\_compendium/essence\\_of\\_winning\\_losing.pdf](http://pogoarchives.org/m/dni/john_boyd_compendium/essence_of_winning_losing.pdf)), I do not intend this as a constraint. GR can be expressed in many other agent reasoning frameworks.
2. I'm referring to agreeable, rather than rebel, GR agents here. That is, while GR agency can be useful when an operator is available, it is particularly well motivated when the operator is inaccessible during complex environment scenarios.
3. This should read *belief state* throughout, but is shortened for brevity.
4. Using the situation calculus, Task et al. (2018) provide a formalization of the solution space through which this search takes place so as to inform the selection of future heuristic approaches.
5. [www.iver-auv.com/Iver2\\_AUV\\_Brochure.pdf](http://www.iver-auv.com/Iver2_AUV_Brochure.pdf).
6. [oceanai.mit.edu/moos-ivp/docs/GuideTo\\_iOceanServer-Comms.pdf](http://oceanai.mit.edu/moos-ivp/docs/GuideTo_iOceanServer-Comms.pdf).
7. Returning is important. In 2010, four Navy AUVs, with a collective value of one million dollars, were lost during a training exercise. They were found only after an intense search effort.
8. I often use *goal* as shorthand for *goal node* in this section.
9. GRIM was implemented using our group's ActorSim platform ([makro.ink/actorsim](http://makro.ink/actorsim)).
10. For example, the IJCAI-17 Workshop on Goal Reasoning ([makro.ink/ijcai2017grw](http://makro.ink/ijcai2017grw)).
11. [www.darpa.mil/program/explainable-artificial-intelligence](http://www.darpa.mil/program/explainable-artificial-intelligence).

## References

- Aha, D. W., and Coman, A. 2017. The AI Rebellion: Changing the Narrative. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 4826–4830. Palo Alto, CA: AAAI Press.

- Aha, D. W.; Darrell, T.; Pazzani, M.; Reid, D.; Sammut, C.; and Stone, P., eds. 2017. Explainable AI: Papers from the IJCAI Workshop. Unpublished proceedings. Melbourne, Australia, August 20. [www.intelligentrobots.org/files/IJCAI2017/IJCAI-17\\_X\\_AI\\_WS\\_Proceedings.pdf](http://www.intelligentrobots.org/files/IJCAI2017/IJCAI-17_X_AI_WS_Proceedings.pdf).
- Alford, R.; Borck, H.; Karneeb, J.; and Aha, D. W. 2015. Active Behavior Recognition in Beyond Visual Range Air Combat. In *Proceedings of the Third Conference on Advances in Cognitive Systems*. Palo Alto, CA: Cognitive Systems Foundation.
- Altmann, E. M., and Trafton, J. G. 2002. Memory for Goals: An Activation-Based Model. *Cognitive Science* 26(1): 39–83. doi.org/10.1207/s15516709cog2601.
- Anderson, J. R., and Lebiere, C., eds. 1998. *The Atomic Components of Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Apker, T. B.; Johnson, B.; and Humphrey, L. 2016. LTL Templates for Play-Calling Supervisory Control. In *Proceedings of the AIAA S&T Forum and Exposition*. Reston, VA: AIAA Press. doi.org/10.2514/6.2016-0917
- Auslander, B.; Floyd, M. W.; Apker, T.; Johnson, B.; Roberts, M.; and Aha, D. W. 2015. Learning to Estimate: A Case-Based Approach to Task Execution Prediction. In *Proceedings of the 23rd International Conference on Case-Based Reasoning*, 15–29. Berlin: Springer. doi.org/10.1007/978-3-319-24586-7.
- Balch, T.; Dellaert, F.; Feldman, A.; Guillory, A.; Isbell, C. L.; Khan, Z.; Pratt, S. C.; Stein, A. N.; and Wilde, H. 2006. How Multirobot Systems Research Will Accelerate Our Understanding of Social Animal Behavior. *Proceedings of the IEEE* 94(7): 1445–1463. doi.org/10.1109/JPROC.2006.876969
- Bengfort, B., and Cox, M. T. 2015. Interactive Knowledge-Goal Reasoning. In *Goal Reasoning: Papers from the ACS Workshop*, edited by D. W. Aha. Technical Report GT-IRIM-CR-2015-001. Atlanta, GA: Georgia Institute of Technology Institute for Robotics and Intelligent Machines.
- Benjamin, M. R.; Schmidt, H.; Newman, P. M.; and Leonard, J. J. 2010. Nested Autonomy for Unmanned Marine Vehicles with Moos-IvP. *Journal of Field Robotics* 27(6): 834–875. doi.org/10.1002/rob.20370
- Bloem, R.; Jobstmann, B.; Piterman, N.; Pnueli, A.; and Sa’r, Y. 2012. Synthesis of Reactive(1) Designs. *Journal of Computer and System Sciences* 78(3): 911–938. doi.org/10.1016/j.jcss.2011.08.007
- Bonanno, D.; Roberts, M.; Smith, L.; and Aha, D. W. 2016. Selecting Subgoals Using Deep Learning in Minecraft: A Preliminary Report. In *Deep Learning for Artificial Intelligence: Papers from the IJCAI Workshop*, edited by D. Aha, A. Wagner, A. Gordon, and Y. Aloimonos. Unpublished proceedings. New York, NY, July 10. [home.earthlink.net/~dwaha/research/meetings/ijcai16-dlai-ws/](http://home.earthlink.net/~dwaha/research/meetings/ijcai16-dlai-ws/).
- Borck, H.; Karneeb, J.; Floyd, M. W.; Alford, R.; and Aha, D. W. 2015. Case-Based Policy and Goal Recognition. In *Proceedings of the 23rd International Conference on Case-Based Reasoning*, 30–43. Berlin: Springer. doi.org/10.1007/978-3-319-24586-7.
- Borrajó, D.; Roubícková, A.; and Serina, I. 2015. Progress in Case-Based Planning. *ACM Computing Surveys* 47(2): 35:1–39. doi.org/10.1145/2674024
- Braubach, L.; Pokahr, A.; Moldt, D.; and Lamersdorf, W. 2004. Goal Representation for BDI Agent Systems. In *Proceedings of the Second International Workshop on Programming Multi-Agent Systems*, 44–65. Berlin: Springer.
- Cashmore, M.; Fox, M.; Long, D.; Magazzeni, D.; Ridder, B.; Carrera, A.; Palomeras, N.; Hurtós, N. and Carreras, M. 2015. Rosplan: Planning in the Robot Operating System. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling*, 333–341. Palo Alto, CA: AAAI Press.
- Chien, S. A.; Knight, R.; Stechert, A.; Sherwood, R.; and Rabideau, G. 2000. Using Iterative Repair to Improve the Responsiveness of Planning and Scheduling. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, 300–307. Palo Alto, CA: AAAI Press.
- Choi, D. 2011. Reactive Goal Management in a Cognitive Architecture. *Cognitive Systems Research* 12(3–4): 293–308. doi.org/10.1016/j.cogsys.2010.09.002
- Clive, P. D.; Johnson, J. A.; Moss, M. J.; Zeh, J. M.; Birkmire, B. M.; and Hodson, D. D. 2015. Advanced Framework for Simulation, Integration and Modeling (AFSIM). Paper presented at the International Conference on Scientific Computing. Las Vegas, NV, July 27–30.
- Coddington, A. M. 2006. Motivations for MADbot: A Motivated and Goal Directed Robot. Paper presented at the Twenty-Fifth Workshop of the UK Planning and Scheduling Special Interest Group. Nottingham, UK, December 14–15. [www.cs.nott.ac.uk/~pszrq/PlanSIG/papers.htm](http://www.cs.nott.ac.uk/~pszrq/PlanSIG/papers.htm)
- Coddington, A. M., and Luck, M. 2004. A Motivation-Based Planning and Execution Framework. *International Journal on Artificial Intelligence Tools* 13(01): 5–25. doi.org/10.1142/S0218213004001399
- Coles, A., and Coles, A. 2014. PDDL+ Planning with Events and Linear Processes. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling*, 74–82. Palo Alto, CA: AAAI Press.
- Coman, A., and Aha, D. W. 2017. Cognitive Support for Rebel Agents: Social Awareness and Counternarrative Intelligence. In *Proceedings of the Fifth Conference on Advances in Cognitive Systems*. Palo Alto, CA: Cognitive Systems Foundation.
- Cox, M. T. 2007. Perpetual Self-Aware Cognitive Agents. *AI Magazine* 28(1): 32–45. doi.org/10.1609/aimag.v28i1.2027
- Cox, M. T. 2018. A Goal Reasoning Model of Planning, Action, and Interpretation. *Advances in Cognitive Systems* 5: 59–78.
- Cox, M. T.; Alavi, Z.; Dannenhauer, D.; Eyorokon, V.; Muñoz-Avila, H.; and Perlis, D. 2016. MIDCA: A Metacognitive, Integrated Dual-Cycle Architecture for Self-Regulated Autonomy. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 3712–3718. Palo Alto, CA: AAAI Press.
- Cox, M. T.; Dannenhauer, D.; and Kondrakunta, S. 2017. Goal Operations for Cognitive Systems. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 4385–4391. Palo Alto, CA: AAAI Press.
- Dannenhauer, D., and Muñoz-Avila, H. 2015. Raising Expectations in GDA Agents Acting in Dynamic Environments. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2241–2247. Palo Alto, CA: AAAI Press.
- Dannenhauer, D.; Muñoz-Avila, H.; and Cox, M. T. 2016. Informed Expectations to Guide GDA Agents in Partially Observable Environments. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.
- Dastani, M.; and van der Torre, L. 2004. What Is a Normative Goal? In *Proceedings of the First International Workshop on Regulated Agent-Based Social Systems*, 210–227. Berlin: Springer. doi.org/10.1007/978-3-540-25867-4.
- US Department of Defense (DOD) 2011. *Support to Foreign Disaster Relief: Handbook for JTF Commanders and Below*. Technical Report GTA 90–01-030. Washington, DC: Department of Defense, Office of the Assistant Secretary of Defense.
- Duquette, M. 2011. The Common Mission Automation Services Interface. In *Proceedings of the Infotech@Aerospace Conference*. Reston, VA: AIAA Press. doi.org/10.2514/6.2011-1542
- Ernest, N.; Carroll, D.; Schumacher, C.; Clark, M.; Cohen, K.; and Lee, G. 2016. Genetic Fuzzy Based Artificial Intelligence for Unmanned Combat Aerial Vehicle Control in Simulated Air Combat Missions. *Journal of Defense Management* 6(1). doi.org/10.4172/2167-0374.1000144
- Floyd, M. F.; Karneeb, J.; and Aha, D. W. 2017. Case-Based Team Recognition Using Learned Opponent Models. In *Proceedings of the 25th International Conference on Case-*

- Based Reasoning*, 123–138. Berlin: Springer. doi.org/10.1007/978-3-319-61030-6.
- Franke, J.; Brown, S. M.; Bell, B.; and Mendenhall, H. 2000. Enhancing Teamwork Through Team-Level Intent Inference. In *Proceedings of the International Conference on Artificial Intelligence*. Las Vegas, NV: CSREA Press.
- Ghallab, M.; Nau, D.; and Traverso, P. 2014. The Actor's View of Automated Planning and Acting: A Position Paper. *Artificial Intelligence* 208: 1–17. doi.org/10.1016/j.artint.2013.11.002
- Gillespie, K.; Molineaux, M.; Floyd, M. W.; Vattam, S. S.; and Aha, D. W. 2015. Goal Reasoning for an Autonomous Squad Member. In *Goal Reasoning: Papers from the ACS Workshop*, edited by D. W. Aha. Technical Report GT-IRIM-CR-2015-001. Atlanta, GA: Georgia Institute of Technology Institute for Robotics and Intelligent Machines.
- Harland, J.; Morley, D.; Thangarajah, J.; and Yorke-Smith, N. 2014. An Operational Semantics for the Goal Life-Cycle in BDI Agents. *Autonomous Agents and Multi-Agent Systems* 28(4): 682–719. doi.org/10.1007/s10458-013-9238-9
- Hawes, N. 2011. A Survey of Motivation Frameworks for Intelligent Systems. *Artificial Intelligence* 175(5–6): 1020–1036.
- Jaidee, U.; Muñoz-Avila, H.; and Aha, D. W. 2013. Case-Based Goal-Driven Coordination of Multiple Learning Agents. In *Proceedings of the 21st International Conference on Case-Based Reasoning*, 164–178. Berlin: Springer. doi.org/10.1007/978-3-642-39056-2.
- James, D. L., and Welsh, M. A. 2015. USAF Strategic Master Plan. Office of the Secretary of the Air Force, Washington, DC: US Air Force. www.af.mil/Portals/1/documents/Force%20Management/Strategic\_Master\_Plan.pdf
- Johnson, B.; Roberts, M.; Apker, T.; and Aha, D. W. 2016. Goal Reasoning with Information Measures. In *Proceedings of the Fourth Conference on Advances in Cognitive Systems*. Palo Alto, CA: Cognitive Systems Foundation.
- Jones, R. M., and Laird, J. E. 1997. Constraints on the Design of a High-Level Model of Cognition. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, 358–363. Mahwah, NJ: Lawrence Erlbaum Associates.
- Kambhampati, S.; Knoblock, C. A.; and Yang, Q. 1995. Planning as Refinement Search: A Unified Framework for Evaluating Design Tradeoffs in Partial-Order Planning. *Artificial Intelligence* 76: 168–238. doi.org/10.1016/0004-3702(94)00076-D
- Kambhampati, S., and Nau, D. 1994. On the Nature of Modal Truth Criteria in Planning. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 67–97. Menlo Park, CA: AAAI Press.
- Karneeb, J.; Floyd, M. W.; Moore, P.; and Aha, D. W. 2018. Distributed Discrepancy Detection for Beyond-Visual-Range Air Combat. *AI Communications*. doi.org/10.3233/AIC-180757
- Khemlani, S.; Hinterecker, T.; and Johnson-Laird, P. N. 2017. The Provenance of Modal Inference. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*. Somerville, MA: Cognitive Science Society, Inc.
- Klenk, M.; Molineaux, M.; and Aha, D. W. 2013. Goal-Driven Autonomy for Responding to Unexpected Events in Strategy Simulations. *Computational Intelligence* 29(2): 187–206. doi.org/10.1111/j.1467-8640.2012.00445.x
- Kress-Gazit, H.; Fainekos, G. E.; and Pappas, G. J. 2009. Temporal-Logic-Based Reactive Mission and Motion Planning. *IEEE Transactions on Robotics* 25(6): 1370–1381. doi.org/10.1109/TRO.2009.2030225
- Laird, J. E., and Newell, A. 1983. A Universal Weak Method: Summary of Results. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 771–773. San Mateo, CA: Morgan Kaufmann.
- Langley, P.; Cummings, K.; and Shapiro, D. 2004. Hierarchical Skills and Cognitive Architectures. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society*, 779–784. Mahwah, NJ: Lawrence Erlbaum Associates.
- Luo, D.-L.; Shen, C.-L.; Wang, B.; and Wu, W.-H. 2005. Air Combat Decision-Making for Cooperative Multiple Target Attack Using Heuristic Adaptive Genetic Algorithm. In *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, 473–478. Berlin: Springer.
- Marinier, B.; van Lent, M.; and Jones, R. 2010. Applying Appraisal Theories to Goal Directed Autonomy. Paper presented at the AAAI 2010 Workshop on Goal-Directed Autonomy. Atlanta, GA, July 12.
- Marthi, B.; Russell, S.; and Wolfe, J. 2008. Angelic Hierarchical Planning: Optimal and Online Algorithms. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 222–231. Menlo Park, CA: AAAI Press.
- McMahon, J., and Plaku, E. 2016. Mission and Motion Planning for Autonomous Underwater Vehicles Operating in Spatially and Temporally Complex Environments. *Journal of Oceanic Engineering* 41(4): 893–912. doi.org/10.1109/JOE.2015.2503498
- Meneguzzi, F., and de Silva, L. 2015. Planning in BDI Agents: A Survey of the Integration of Planning Algorithms and Agent Reasoning. *Knowledge Engineering Review* 30(1): 1–44. doi.org/10.1017/S0269888913000337
- Molineaux, M., and Aha, D. W. 2014. Learning Unknown Event Models. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 395–401. Palo Alto, CA: AAAI Press.
- Molineaux, M., and Aha, D. W. 2015. Continuous Explanation Generation in a Multi-Agent Domain. In *Proceedings of the Third Conference on Advances in Cognitive Systems*, 1–18. Palo Alto, CA: Cognitive Systems Foundation.
- Molineaux, M.; Klenk, M.; and Aha, D. W. 2010. Goal-Driven Autonomy in a Navy Strategy Simulation. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.
- Mulgund, S.; Harper, K.; Krishnakumar, K.; and Zacharias, G. 1998. Air Combat Tactics Optimization Using Stochastic Genetic Algorithms. In *Proceedings of the International Conference on Systems, Man, and Cybernetics*, 3136–3141. Piscataway, NJ: Institute for Electrical and Electronics Engineers. doi.org/10.1109/ICSMC.1998.726484
- Muñoz-Avila, H.; Jaidee, U.; Aha, D. W.; and Carter, E. 2010. Goal Directed Autonomy with Case-Based Reasoning. In *Proceedings of the 18th International Conference on Case-Based Reasoning*, 228–241. Berlin: Springer.
- Nau, D. S. 2007. Current Trends in Automated Planning. *AI Magazine* 28(4): 43–58. doi.org/10.1609/aimag.v28i4.2067
- NAVAIR. 2013. Next Generation Threat System. Washington, DC: US Navy Naval Air Warfare Center Training Systems Division.
- Norman, T. J., and Long, D. 1996. *Alarms: An Implementation of Motivated Agency. Intelligent Agents: Theories, Architectures, and Languages*, edited by M. Wooldridge, J. P. Muller, and M. Tambe. Berlin: Springer. doi.org/10.1007/3540608052.
- Omohundro, S. 2014. Autonomous Technology and the Greater Human Good. *Journal of Experimental and Theoretical Artificial Intelligence* 26(3): 303–315. doi.org/10.1080/0952813X.2014.895111
- Oxenham, M., and Green, R. 2017. From Direct Tasking to Goal-Driven Autonomy for Underwater Vehicles. Paper presented at the IJCAI 2017 Workshop on Goal Reasoning. Melbourne, Australia, August 19.
- Paisner, M.; Cox, M. T.; Maynard, M.; and Perlis, D. 2014. Goal-Driven Autonomy for Cognitive Systems. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, 2085–2090. Somerville, MA: Cognitive Science Society, Inc.
- Powell, J.; Molineaux, M.; and Aha, D. W. 2011. Active and Interactive Learning of Goal Selection Knowledge. In *Proceedings of the 24th Florida Artificial Intelligence Research*

- Society Conference. Palo Alto, CA: AAAI Press.
- Rajan, K., and Py, F. 2012. T-Rex: Partitioned Inference for AUV Mission Control. In *Further Advances in Unmanned Marine Vehicles*, 171–199. London: The Institution of Engineering and Technology.
- Rao, A. S., and Georgeff, M. P. 1991. Modeling Rational Agents Within A BDI-Architecture. In *Proceedings of the Second Conference on Principles of Knowledge Representation and Reasoning*, 473–484. San Mateo, CA: Morgan Kaufmann Publishers.
- Roberts, M.; Apker, T.; Johnson, B.; Auslander, B.; Wellman, B.; and Aha, D. W. 2015. Coordinating Robot Teams for Disaster Relief. In *Proceedings of the 28th Florida Artificial Intelligence Research Society Conference*. Palo Alto, CA: AAAI Press.
- Roberts, M.; Vattam, S.; Alford, R.; Auslander, B.; Karneeb, J.; Molineaux, M.; Apker, T.; Wilson, M.; McMahan, J.; and Aha, D. W. 2014. Iterative Goal Refinement for Robotics. Paper presented at the 2014 ICAPS Workshop on Planning and Robotics. 22–23 June, Portsmouth, NH.
- Russell, S., and Norvig, P. 2016. *Artificial Intelligence: A Modern Approach*. London: Pearson.
- Selkowitz, A. R.; Lakhmani, S. G.; and Chen, J. Y. 2017. Using Agent Transparency to Support Situation Awareness of the Autonomous Squad Member. *Cognitive Systems Research* 46: 13–25. doi.org/10.1016/j.cogsys.2017.02.003
- Shapiro, S. C. 1992. Artificial Intelligence. In *Encyclopedia of Artificial Intelligence*, edited by S. C. Shapiro. New York: John Wiley.
- Shivashankar, V.; Alford, R.; and Aha, D. W. 2017. Incorporating Domain-Independent Planning Heuristics in Hierarchical Planning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press
- Smith, D. E. 2004. Choosing Objectives in Over-Subscription Planning. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling*, 393–401. Palo Alto, CA: AAAI Press.
- Talamadupula, K.; Benton, J.; Kambhampati, S.; Schermerhorn, P.; and Scheutz, M. 2010. Planning for Human-Robot Teaming in Open Worlds. *ACM Transactions on Intelligent Systems and Technology* 1(2): 14:1–24 doi.org/10.1145/1869397.1869403
- Task, C.; Wilson, M.; Molineaux, M.; and Aha, D. W. 2018. An Illustrated Situation Calculus Abstraction for Iterative Explanatory Diagnosis. *AI Communications*. doi.org/10.3233/AIC-180759
- Thangarajah, J.; Harland, J.; Morley, D.; and Yorke-Smith, N. 2010. Operational Behaviour for Executing, Suspending, and Aborting Goals In BDI Agent Systems. In *Proceedings of the Eighth International Workshop on Declarative Agent Languages*, 1–21. Berlin: Springer.
- To, S. T.; Johnson, B.; Roberts, M.; and Aha, D. W. 2017. A New Approach to Temporal Planning with Rich Metric Temporal Properties. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling*. Palo Alto, CA: AAAI Press.
- van Riemsdijk, M. B.; Dastani, M.; and Winikoff, M. 2008. Goals In Agent Systems: A Unifying Framework. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multi-Agent Systems*, 713–720. New York: Association for Computing Machinery.
- Vashev, E. 2016. Safe Artificial Intelligence and Formal Methods. In *Proceedings of the Seventh International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation*, 704–713. Berlin: Springer. doi.org/10.1007/978-3-319-47166-2.
- Vattam, S. S., and Aha, D. W. 2015. Case-Based Plan Recognition Under Imperfect Observability. In *Proceedings of the 23rd International Conference on Case-Based Reasoning*, 381–395. Berlin: Springer. doi.org/10.1007/978-3-319-24586-7.
- Vattam, S.; Klenk, M.; Molineaux, M.; and Aha, D. W. 2013. Breadth of Approaches to Goal Reasoning: A Research Survey. In *Goal Reasoning: Papers from the ACS Workshop*, edited by D. W. Aha, M. T. Cox, and H. Muñoz-Avila. Technical Report CS-TR-5029. College Park, MD: University of Maryland, Department of Computer Science.
- Vered, M., and Kaminka, G. A. 2017. Online Recognition of Navigation Goals Through Goal Mirroring. In *Proceedings of the 16th Conference on Autonomous Agents and Multi-agent Systems*, 1748–1750. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Weber, B.; Mateas, M.; and Jhala, A. 2012. Learning from Demonstration for Goal-Driven Autonomy. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.
- Wilson, M. A.; McMahan, J.; and Aha, D. W. 2014. Bounded Expectations for Discrepancy Detection in Goal-Driven Autonomy. In *AI and Robotics: Papers from the AAAI Workshop*, edited by A. Saffiotti, N. Hawes, G. Konidaris, and M. Tenorth. Technical Report WS-01-14. Palo Alto, CA: AAAI Press.
- Wilson, M. A.; McMahan, J.; Wolek, A.; Aha, D. W.; and Houston, B. H. 2018. Goal Reasoning for Autonomous Underwater Vehicles: Responding to Unexpected Agents. *AI Communications*. doi.org/10.3233/AIC-180755
- Wilson, M.; Molineaux, M.; and Aha, D. W. 2013. Domain-Independent Heuristics for Goal Formulation. In *Proceedings of the 26th Florida Artificial Intelligence Research Society Conference*, 160–165. Palo Alto, CA: AAAI Press.
- Winikoff, M.; Dastani, M.; and van Riemsdijk, M. B. 2010. A Unified Interaction-Aware Goal Framework. In *Proceedings of the European Conference on Artificial Intelligence*, 1033–1034. Amsterdam, The Netherlands: IOS Press.
- Young, J., and Hawes, N. 2012. Evolutionary Learning of Goal Priorities in a Real-Time Strategy Game. In *Proceedings of the Eighth Annual International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 87–92. Palo Alto, CA: AAAI Press.

**David W. Aha** (PhD, University of California, Irvine, 1990) leads NRL's Adaptive Systems Section, within the Navy Center for Applied Research in AI, in Washington, DC. His interests include mixed-initiative intelligent agents (for example, that employ goal reasoning models), deliberative autonomy, explainable AI, case-based reasoning, and machine learning, among other topics. He has co-organized 35 events on these topics, launched the UCI Repository for ML Databases, served as an AAAI councilor, cocreated AAAI's AI video competition, and led evaluation teams for four DARPA or ONR programs.