

Goal Reasoning and Trusted Autonomy

Benjamin Johnson, Michael W. Floyd, Alexandra Coman, Mark A. Wilson, and David W. Aha

Abstract This chapter discusses the topic of Goal Reasoning and its relation to Trusted Autonomy. Goal Reasoning studies how autonomous agents can extend their reasoning capabilities beyond their plans and actions, to consider their goals. Such capability allows a Goal Reasoning system to more intelligently react to unexpected events or changes in the environment. We present two different models of Goal Reasoning: Goal-Driven Autonomy (GDA) and goal refinement. We then discuss several research topics related to each, and how they relate to the topic of Trusted Autonomy. Finally, we discuss several directions of ongoing work that are particularly interesting in the context of the chapter: using a model of inverse trust as a basis for adaptive autonomy, and studying how Goal Reasoning agents may choose to rebel (i.e., act contrary to a given command).

Benjamin Johnson
NRC Research Associate at the US Naval Research Laboratory; Washington, DC; USA
e-mail: benjamin.johnson.ctr@nrl.navy.mil

Michael W. Floyd
Knexus Research Corporation; Springfield, VA; USA
e-mail: michael.floyd@knexusresearch.com

Alexandra Coman
NRC Research Associate at the US Naval Research Laboratory; Washington, DC; USA
e-mail: alexandra.coman.ctr.ro@nrl.navy.mil

Mark A. Wilson
Navy Center for Applied Research in AI, US Naval Research Laboratory; Washington, DC; USA
e-mail: mark.wilson@nrl.navy.mil

David W. Aha
Navy Center for Applied Research in AI, US Naval Research Laboratory; Washington, DC; USA
e-mail: david.aha@nrl.navy.mil

1 Introduction

An important consideration for any autonomous system is the need to react intelligently to unplanned events and observations. Doing so requires that the system be given the freedom and ability to adjust its behavior, without being commanded to do so by a human operator or external system. A good deal of research in robotics and autonomy focuses on developing systems that can change their actions and plans, in order to more reliably or more optimally achieve their goals. However, it is also important to develop systems that autonomously deliberate on the goals themselves. Goal Reasoning (e.g., [38]) is the study of how autonomous agents can dynamically reason about and adjust their goals. Doing so enables agents to adapt intelligently to changing conditions and unexpected events, allowing them to address a wider variety of complex problems.

Goal Reasoning capabilities, in one form or another, may prove useful in many applications. In domains where the agents must operate autonomously for substantial periods of time or with limited communications, such as in unmanned underwater operations, it is imperative that the agents have the freedom to act on the information they gather during the mission; they cannot rely on timely operator input, and must be able to adjust their goals autonomously, should the need arise. Goal Reasoning can also be used to allow autonomous agents to deliberate about the goals of other agents, such as an operator. This can lower the burden on the operator and avoid information overload, allowing the operator to focus on other, immediate tasks (e.g., operating their own vehicle, or observing their surroundings). Similarly, Goal Reasoning can be useful in systems that involve multiple, collaborating autonomous agents. In such a system, it is valuable for each agent to be able to adjust its own goals, based on the goals or actions of the other agents with which it is collaborating.

Goal Reasoning also raises interesting questions with regard to the topic of Trusted Autonomy. One definition of Trusted Autonomy, taken from [1], refers to it as:

[T]he ability to form teams of humans and/or machines that make educated and conscious decisions to delegate risky tasks among team members seamlessly and symbiotically.

Trust, then, is inherent in any well-functioning autonomous system in which tasks or goals are delegated to agents within the system. In particular, a Goal Reasoning agent requires a large degree of trust to have the freedom to autonomously determine and adjust its goals. As such, Goal Reasoning pushes the scope of Trusted Autonomy, as the trust of other agents must extend beyond the completion of known tasks or towards achieving a known goal, to trusting the agent's ability to make decisions regarding its goals.

There are a number of open questions with respect to Goal Reasoning agents and their relation to Trusted Autonomy. What motivates the agent to change its goals? How do humans interact with robotic agents and delegate tasks? When will the agents choose to overrule an operator's commands, and why? How can the Goal Reasoning process be framed to promote transparency? How can one ensure that certain safety conditions and guarantees are maintained, despite the additional free-

dom of autonomy provided to a Goal Reasoning agent? These are just a few of the questions raised by the relationship between Goal Reasoning and Trusted Autonomy. In this chapter we describe Goal Reasoning and elaborate on some of these important questions. While we focus on a few selected topics of research, there is a large and growing body of work on Goal Reasoning and related topics. For additional reading, see the survey papers [20, 38], or the proceedings of several Goal Reasoning workshops [2, 3, 4, 34].

This chapter is structured as follows. In Section 2, we describe a simple model of Goal Reasoning called Goal-Driven Autonomy (GDA), as well as a domain-independent method for goal selection in GDA, and an application of GDA in a human-robot teaming task. Section 3 focuses on a more comprehensive model of Goal Reasoning based on goal refinement, an architecture for ensuring the behaviors of Goal Reasoning agents that use this model, and its application in a distributed robotics task. For both models, we describe the importance of transparency to engendering operator trust. We then describe two extensions to Goal Reasoning in Section 4, first on how inverse trust can be used as a basis for adaptive autonomy, and then on rebel agents and their relation to Trusted Autonomy. We then conclude in Section 5.

2 Goal-Driven Autonomy Models

This section describes the Goal-Driven Autonomy (GDA) model of Goal Reasoning, which has been studied by several groups (e.g., [11, 12, 13, 22, 25, 29, 39]). We discuss only some of our own group's work on GDA, and its relation to Trusted Autonomy. We start with an introduction to GDA in Section 2.1, describe an approach for goal selection in GDA in Section 2.2, and an application of the GDA model in a human-robot teaming task in Section 2.3.

2.1 Goal-Driven Autonomy

One proposed model for Goal Reasoning is that of Goal-Driven Autonomy (GDA) [25, 28], which allows an autonomous agent to introduce new goals, manage existing goals, and preempt active goals. An early instantiation of GDA is in an agent called Autonomous Response to Unexpected Events (ARTUE). Molineaux *et al.* show that the Goal Reasoning capabilities provided by GDA allow ARTUE to better react to unexpected events, and improve performance, versus an on-line planning system [28].

In the GDA model (shown in Figure 1), an agent performs Goal Reasoning via a repeated 4-step sequence:

1. First, the agent uses a *Discrepancy Detector* to compare its observations with a set of expected observations (given by the planner). Any differences between the expected and actual observations are used to define a set of discrepancies.
2. Next, the *Explanation Generator* creates one or more possible explanations for the set of discrepancies. Each explanation hypothesizes a possible cause for the set of discrepancies, based on the current and prior observed states.
3. Third, the *Goal Nominator* nominates a set of potentially appropriate goals, based on the generated explanation(s).
4. Finally, given the current set of pending goals and the set of newly nominated goals, the *Goal Manager* selects a subset of goals to be passed to the planner as pending goals. This may involve adding, deleting, and/or modifying the pending goals.

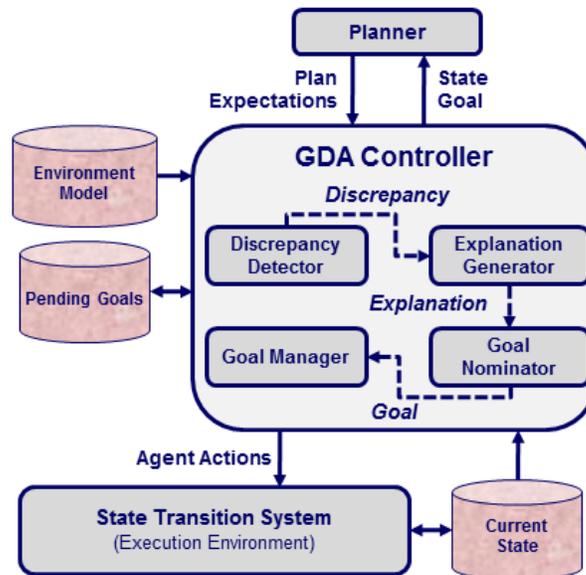


Fig. 1: Conceptual diagram of the Goal-Driven Autonomy model.

These four steps (Discrepancy Detection, Explanation Generation, Goal Nomination, and Goal Management) form the core of the GDA model of Goal Reasoning. Molineaux *et al.* [28] also contrasted this model with a conceptual model of on-line planning [30]. In both cases, the planner uses a model of the environment, a current state, and an active goal, and generates a plan to achieve that goal. However, GDA's planner also generates a set of expectations. For both models, the controller uses the generated plan to apply an action to the state transition system, and updates the current state. The primary difference between GDA and the on-line planning model

is in the ability of the GDA controller to reason over a set of goals, rather than being limited to pursuing a single, set goal.

We relate GDA to the topic of Trusted Autonomy as follows. In Section 2.2 we describe a domain-independent method for goal selection, which is a subtask of goal management, and how it can be biased to choose goals that engender operator trust. Then in Section 2.3 we describe the use of a GDA agent in a simulated mission involving human-robot teaming, and the importance of transparency in that mission.

2.2 Goal Selection

An integral part of Goal Reasoning is the problem of goal selection (i.e., a subtask of goal management in which one or more goals are chosen for subsequent execution). Agents that are teamed directly with humans can receive operator-selected goals, or seek approval from operators to pursue their self-selected goals. However, some agents may not have access to operators in a timely fashion. For instance, an autonomous underwater vehicle (AUV) cannot communicate with operators unless it surfaces, and an interplanetary robot may require excessive time to consult operators on mission-critical decisions. In these and similar contexts, the agents' ability to intelligently select their goals is critical.

One approach for goal selection involves manually constructing knowledge bases that dictate what goals to formulate based on the agent's beliefs about the world (e.g., ARTUE [28] uses an engineered rule set that governs goal formulation). However, this approach requires extensive domain-specific knowledge engineering. It is also limiting, as the agent knows only how to respond to situations that were anticipated by the designer. The greater control afforded by hand-tuned goal selection mechanisms may appear at first to offer greater predictability for operators. However, an agent employing such a system cannot be expected to respond intelligently to situations outside its programmed knowledge.

For instance, during tests applying GDA on an AUV, we found that low-level software responsible for controlling the vehicle was at one point unable to correctly determine the vehicle's motion [41]. The vehicle deviated significantly from its intended trajectory without reporting that deviation to the GDA agent. More robust goal selection abilities offer the possibility of mitigating such failures through autonomous responses.

Some agents address the need for more robust goal selection through the use of learning. For example, agents may learn goal selection knowledge from criticism and query-answer interaction with a human expert [31], from demonstrations by human experts [40], or from Q-learning [22]. These approaches are more adaptable than manually engineered systems, but they rely on the availability of human experts or demonstration data for training. Also, agents using these approaches may perform poorly when confronted with new situations for which they were not trained.

Another approach is to control goal selection through the application of *motivators*, which encode high-level, domain-independent desires the agent wants to fulfill.

We adopted this approach in Motivated ARTUE (M-ARTUE) [42], an extension of ARTUE. M-ARTUE expresses these motivators in terms of the agent’s planning model; thus the motivator functions themselves are domain-independent and do not require further domain knowledge than that encoded for the agent’s planner. M-ARTUE applies the following motivators to guide its behavior:

- **Social:** This encodes the desire to pursue goals provided by the agent’s human operators or teammates.
- **Opportunity:** This encodes the desire to gather and conserve resources, as well as preserve the agent’s possible actions in future states.
- **Exploration:** This encodes the desire to visit states that the agent has not visited previously.

To achieve broadly applicable, domain-independent implementations of these motivators, we first introduced two subfunctions:

- **Urgency:** $u_m(s_c)$ returns a numeric value representing the agent’s need to fulfill a particular motivator m in the current state s_c .
- **Fitness:** $f_m(X_g)$ returns a numeric value representing how well a plan (to achieve a given goal g) fulfills a particular motivator m , using the sequence of states X_g the agent expects to visit while executing the plan.

We defined these functions to embody the following properties for each motivator:

- *Social:* This motivator’s urgency increases as time passes without the agent achieving any operator goals. Its fitness expresses a high value for any plan that achieves an operator goal, and a low value for other plans.
- *Opportunity:* This motivator’s urgency increases as the agent expends resources and as fewer actions are available in the current state. Its fitness expresses higher values for plans that retain high quantities of resources and many available actions, and lower values for plans that do not.
- *Exploration:* This motivator’s urgency is initially high and decreases as time passes, to encourage the agent to explore early but prioritize other desires after it has gathered more information. Its fitness expresses higher values for plans that visit more new states, and lower values for plans that visit more known states.

During goal selection, M-ARTUE prefers the goal with the highest overall fitness, which is defined for each goal in the current state as the sum of the motivators’ fitness scores, weighted by urgency:

$$F(X_g, s_c) = \sum_m u_m(s_c) f_m(X_g)$$

M-ARTUE’s approach for goal selection was tested in a simulated Mars rover domain with hazards, with the objective of successfully maneuvering up to three rovers to given destinations around a map within a fixed number of actions. In three different levels of difficulty (controlled by the prevalence of hazards) it achieved

comparable performance to ARTUE without requiring domain engineering for goal selection [42].

Other researchers have investigated the use of motivations or drives as goal selection mechanisms. For instance, Sun [36] uses drives analogous to our motivators in CLARION; however, the drives are domain-specific (e.g., Thirst and Hunger drives in CLARION contrast with resource management in our Opportunity motivator), and their preferred implementation is using a supervised back-propagating neural network. Also, the experimental focus of [36] is on cognitive plausibility, not agent performance. Merrick and Shafi [27] focus on three motivations (Achievement, Affiliation, and Power), extending a psychological theory of achievement that models competing impulses of success and failure. Unlike our work, this forgoes planning as a mechanism, instead modeling probability of success using past experiences and limited domain knowledge. The authors propose inverse probability of success or socially-determined value of goals as simpler alternatives to modeling how well a goal satisfies a particular motivation; but our motivators address the problem of determining direct value of a goal. Finally, this work proposes "motive profiles" for agents (encompassing different sets of values for the model's parameters) and focuses on testing those motive profiles against the expected responses for corresponding human profiles in certain psychological tests. Baldassarre and Mirolli [6] focus on the use of the psychological theory of *intrinsic motivation* as a basis for long-term learning through exploration. This work bears some resemblance to our Exploration motivator; however, we focus primarily on acquisition of knowledge about the world and do not address the acquisition of new skills. Moreover, our system is guided by novelty of state, as contrasted with metrics of predictability or acquisition of competence.

Goal selection methods based on motivators (or other primitives) allow autonomous agents to make their own decisions in situations their designers did not anticipate. This may be viewed as a step toward greater autonomy, but does not necessarily establish a mutual understanding of trust as a concept [1]. Introducing an additional motivator that represents the desire to establish trust between the agent and its operators (or other agents) could establish a basis of understanding for trustworthiness, as well as a means for the agent to determine courses of action that would maximize human-machine trust. For example, in [17], an autonomous agent applies "inverse trust" metrics that guide a robot's behavior towards increasing the trust it receives from human operators. A motivator utilizing a similar metric, applied in goal selection, might enable a Goal Reasoning agent to become more trusted by its operators. In future research, we plan to investigate an extension of M-ARTUE that incorporates trust-related motivators.

2.3 An Application for Human-Robot Teaming

A solitary Goal Reasoning agent can be the sole determiner of its own tasks and goals. However, if the agent collaborates with other agents or humans as a mem-

ber of a team, it needs to consider both the goals of individual teammates as well as the overall team goals. Failure to do so could lead to teammates viewing the agent as selfish (e.g., never assisting any of its teammates) or hinder the efficient achievement of team objectives (e.g., performing actions that create more work for teammates). This contrasts with a traditional multi-agent setting where other entities may provide a Goal Reasoning agent with motivations for goal change (e.g., saving an injured civilian, defending against an aggressive enemy), but they do not share any squad-level goals with the agent. It may be necessary to reason about their goals or motivations, which requires sharing team goals with the agent.

Our Autonomous Squad Member (ASM) project [18] focuses on the design and development of an extended GDA agent that controls a simulated unmanned ground vehicle, which is embedded with a detached squad that is performing surveillance tasks in a rural environment. The ASM agent (Figure 2) observes the other squad members and controls its behavior accordingly.

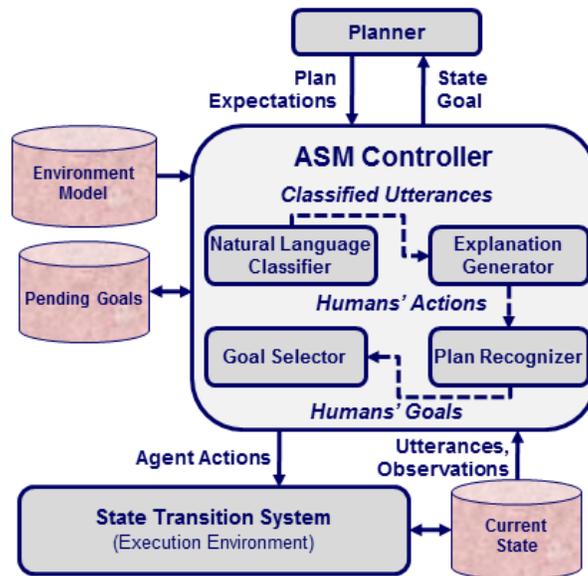


Fig. 2: The Autonomous Squad Member (ASM) agent's conceptual design

More specifically, the ASM agent continuously monitors the behavior of its teammates to identify their current goals. Using its current sensory inputs (i.e., observations of the environment or spoken dialog detected by its *Natural Language Classifier*), the agent's *Explanation Generator* attempts to explain what actions each teammate must have performed for the environment to be in its current state (i.e., explain the most recent actions that were observed). The actions of each teammate are used by the agent's *Plan Recognizer* to recognize their respective plans and associated goals (i.e., predict their future actions based on previously observed actions).

If the ASM agent determines that some, or all, of the teammates have changed their goals, it can change its goal in response, using its *Goal Selector*. For example, if the teammates were *patrolling* but are now *retreating*, it can use that information to reason that there must be a threat that the teammates have observed but the agent has not, so it should also retreat. Similarly, the agent can also modify its goals in response to an opportunity or an unexpected external event (i.e., something it perceives in the environment). The ASM agent's current goal is used to control its behavior by generating a plan (i.e., using the *Planner*) and executing its actions in the environment.

The ASM model can be compared to the GDA model introduced in Section 2.1 and shown in Figure 1, with several notable differences. The ASM model specifically differentiates natural language utterances from other state observations and uses the *Natural Language Classifier* to pre-process them before they are provided as input to the *Explanation Generator*. Explanation in the ASM model is not a single process but is done at two levels, with the *Explanation Generator* explaining what actions each teammate must have performed and the *Plan Recognizer* reasoning about what plans they must be performing. Similarly, in the ASM model, discrepancy detection is not a single module but happens throughout the system. For example, discrepancies about expected and observed states are handled in the *Explanation Generator*, while discrepancies about past and current goals of teammates are handled in the *Goal Selector*. The *Goal Selector* itself differs from the GDA model since it only performs a subset of the duties of the GDA model's *Goal Manager*.

To support Trusted Autonomy, we are integrating the ASM agent with a user interface that adds transparency between the agent and a human operator. This interface uses the Situation awareness-based Agent Transparency (SAT) model [9], a transparency model that attempts to reduce user overhead, provide situational awareness, and allow for appropriate calibration of trust in the agent. The SAT interface provides three levels of transparency information: the agent's status (e.g., current state, goals, plans, physical location), the agent's reasoning process (e.g., what motivated it to perform its current task), and the agent's projections (e.g., future environment states, future resource levels). Each transparency element is presented using icons on the user interface, allowing the user to quickly and intuitively process information and identify changes in the agent's behavior.

In this application of SAT, we are displaying to the user interface information pertaining to the ASM agent's location in its environment, its current goal, its current task, and influence factors pertaining to why it selected that goal and is operating on that task. Our objective is to demonstrate the benefits of the SAT model in a real-time environment in mission-critical situations.

3 Goal Refinement

Section 2 described some of our group’s work on GDA, a simple model of Goal Reasoning. In this section, we describe our work on goal refinement, which is a more comprehensive Goal Reasoning model. We begin in Section 3.1 by describing its basic concepts and realization in the Goal Lifecycle. In Section 3.2 we present an architecture for ensuring guarantees on the behavior of agents that employ this model. Finally, we describe its application to a distributed robotics task in Section 3.3.

3.1 Goal Lifecycle

Our group defined a second model for Goal Reasoning, based on the concept of goal refinement, which we call the Goal Lifecycle [33]. Goal refinement, an extension of plan refinement [24], models the progressive refinement of goals through the addition of constraints. This is visualized in the Goal Lifecycle shown in Figure 3 [33]. In this model, individual goals transition through stages of increasingly detailed *modes* by activating a series of refinement *strategies*. For example, goals and their initial constraints are introduced using the *formulate* strategy, while the *expand* strategy concerns the automated generation of plans for a given goal.

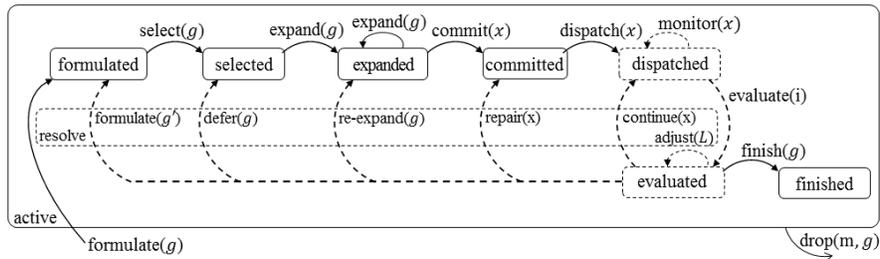


Fig. 3: The Goal Lifecycle [33] depicts the application of strategies to transition goals (and their associated constraints) through a sequence of modes. While the top-level strategies progress goals towards completion, the set of *resolve* strategies (e.g., *re-expand*) support goal adaptation, deferment, and reformulation.

Briefly, the refinement strategies are:

1. *Formulate*: This creates a new goal and enters it into the Goal Lifecycle by defining its initial constraints, criteria, and prerequisites.
2. *Select*: This chooses which goal(s) to actively pursue; it ensures that the goals’ prerequisites are met and that the agent has the needed resources to pursue them.
3. *Expand*: This generates one or more expansions (i.e., plans) to achieve a given goal, along with a set of expectations for each.

4. *Commit*: This picks a single expansion to pursue from the set of expansions created by the expand strategy.
5. *Dispatch*: This executes the committed expansion and defines the criteria by which a goal can be evaluated during execution.

In addition to these strategies, which progressively add detail to the goal's definition, the Goal Lifecycle includes a set of strategies for detecting and reacting to events and changes during execution. After being dispatched, each goal can be actively *monitored* and, if problems are detected, or if an unexpected event occurs, the goal can be *evaluated*. As a result of this evaluation, the system may elect to *continue* the goal as is, *drop* the goal (as either completed or failed), or attempt to resolve the detected problems through one of several strategies (e.g., *repair*, *defer*). *Resolve* strategies transition a goal to an earlier mode before execution resumes.

In contrast to the GDA model, goal refinement provides a more explicit representation of the context in which a goal is pursued by a Goal Reasoning agent. This has benefits that relate to Trusted Autonomy. For example, contextual constraints can be used to guarantee that agents will behave according to a given specification, and can also be used to more clearly deliberate on and communicate details of their reasoning (i.e., for selecting the next strategy to apply). We discuss these topics in the following subsections.

3.2 *Guaranteeing the Execution of Specified Behaviors*

Once a Goal Reasoning agent is provided with or self-selects a goal to pursue, it must use some combination of planning and control algorithms to undertake the actions necessary to achieve it. Careful design of these components can provide valuable capabilities for a Goal Reasoning agent. Here we describe the Situated Decision Process (SDP), which manages and executes goals for a team of autonomous vehicles. A more thorough description of the SDP and its components can be found in [33], and we describe an application of the SDP in Section 3.3.

The SDP (Figure 4) takes as input goal updates (e.g., commands) from an operator and passes them to the Mission Manager, which performs Goal Reasoning operations using the Goal Lifecycle described in Section 3.1. Once the Mission Manager selects a goal, it dispatches an expansion to the vehicles by creating a schedule of commands for them and passing that schedule to the Coordination Manager. The Coordination Manager interprets the schedule and passes the applicable commands to a Team Executive, which then assigns the commands to individual vehicles.

Each vehicle interprets their command as an input to a Finite State Automaton (FSA), which is automatically synthesized using a template. This template specifies the regions where the behaviors are to be executed, ensuring they are active in only appropriate areas, as well as any *mission sensors* that cause an automatic switching between behaviors when the vehicle observes a particular event. This yields a play-calling architecture, detailed in [5], that provides guarantees on the execution of the goals chosen by the Mission Manager. The execution of each command is predicated



Fig. 4: A conceptual design of the Situated Decision Process (SDP). The Mission Manager performs Goal Reasoning operations. It creates a schedule of actions for a team of vehicles, each of which operates a synthesized Finite State Automaton.

on the satisfaction of a set of pre-defined *health sensors*, which establish required conditions before the vehicle can pursue the commanded goal (e.g., the vehicle must have a sufficient amount of fuel to reach the goal location). If one or more of the health sensors are not satisfied, the FSA activates a *contingency behavior* that causes the vehicle to engage in a behavior aimed towards maintaining safety (e.g., landing an air vehicle) or fixing the health sensor (e.g., returning to a base station to refuel).

The FSAs used by the vehicles are synthesized from a temporal logic specification, and are guaranteed to satisfy this specification. Full details on the synthesis process we use in the SDP can be found in [26]. Briefly, the behavior of the vehicle is specified as a Linear Temporal Logic (LTL) formula:

$$\phi = \varphi_e \rightarrow \varphi_s, \quad (1)$$

where the behavior of system φ_s is specified in reaction to changes in environment φ_e . The specified behavior of the environment and system includes three components:

- The initial state: $\varphi_i^{\{e,s\}}$
- A set of safety constraints that restrict the transitions of the system: $\varphi_r^{\{e,s\}}$
- A set of goal conditions that must be satisfied infinitely often: $\varphi_g^{\{e,s\}}$

Formula ϕ is specified over a set of Boolean propositions that represent the state of the environment, as sensed by the vehicle, and the state of the system. The resulting FSA is synthesized automatically, in a manner that guarantees that the transitions given by the FSA will satisfy ϕ . Coupled with the play-calling templates that are used to generate the specification, the resulting FSA is guaranteed to activate behaviors that pursue the commanded goal, whenever its observed internal and external state allow it to do so.

This framework relates to Trusted Autonomy. While the SDP can adjust its goals autonomously, the pursuit of those goals is constrained to abide by specific guarantees. Such guarantees will affect how such an agent or group of agents is viewed and trusted within a larger system.

3.3 A Distributed Robotics Application

One interesting question concerning Goal Reasoning and Trusted Autonomy is how to design a Goal Reasoning system that clearly communicates how and why it chooses to change its goals. We have addressed this in Goal Reasoning with Information Measures (GRIM), a Goal Reasoning system that instantiates the Goal Lifecycle, described in Section 3.1. GRIM, which is a derivation of the SDP (Section 3.2), employs a single measure for assessing goal performance and communicates this to an operator. We briefly describe an application of GRIM here; a more complete description can be found in [23].

We applied GRIM to a simulated disaster relief scenario (Figure 5) where a team of two autonomous vehicles must survey a set of regions to locate a local official and establish communications. Each of these three regions (labeled as an Airport and two Office Buildings) corresponds to an individual survey goal within the Goal Lifecycle, and is surveyed by following a series of waypoints. The Goal Reasoning process in GRIM is then framed with respect to the uncertainty left in the area survey, which is defined as the length of the search pattern that has yet to be traversed by the vehicles.

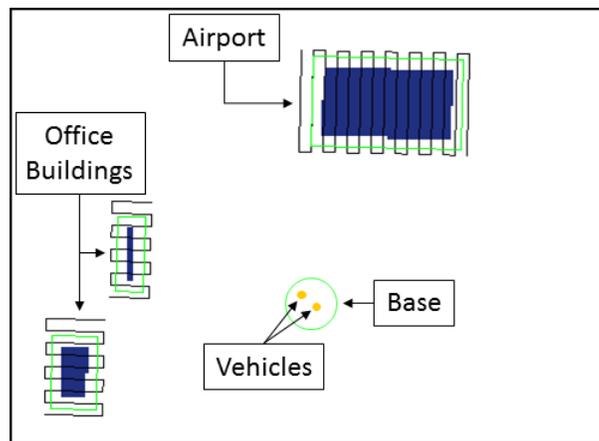


Fig. 5: Map of the survey regions for the disaster relief scenario used to demonstrate GRIM. Each of the three survey regions is covered by a waypoint pattern that the vehicles follow to search for a local official.

Figure 6 displays a graphical representation of four of the Goal Lifecycle strategies. In Figure 6a, each of the three survey goals is *formulated* by generating constraints on the maximum allowable uncertainty over time. After each of the goals is *formulated*, GRIM *selects* a single goal (the Airport survey goal) to pursue, based on the constraints of each goal. The *selected* goal is then *expanded* by generating a set of plans to achieve it. The expectations of these plans (depicted as a change in the uncertainty over time) are shown in Figure 6b. GRIM then *commits* to a single expansion, and *dispatches* that expansion to the vehicles. The expectations for the expansion, and a set of performance bounds that are generated as part of the *dispatch* strategy, are shown in Figure 6c. Finally, Figure 6d displays the execution performance over time, as obtained by the *monitor* strategy.

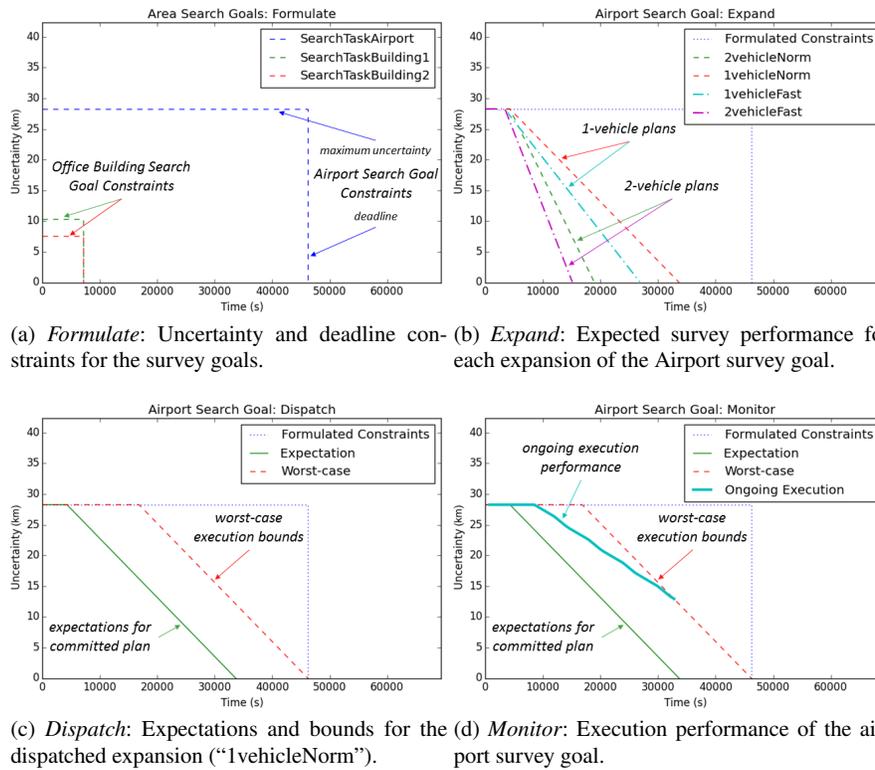


Fig. 6: Plots of selected strategies from the execution of GRIM for the disaster relief scenario.

During execution, when the vehicle’s performance is predicted to violate a goal constraint (as occurs in Figure 6d when its ongoing execution reaches the worst-case execution 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 constraints on the goal, the goal is marked as failed and *dropped*. If neither of these has occurred, but the performance violates the execution bounds, a *resolve* strategy is activated in an attempt to adjust the goal (or its expansion) before continuing execution. The selected *resolve* strategy can transition the goal back to an earlier mode in the Goal Lifecycle. 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 completely *re-expand* the goal to obtain a new set of expansions, before proceeding to *commit* to and *dispatch* one of the new expansions.

For further details on the operation of GRIM, please see [23], which describes an ablation study with the *resolve* strategies. We showed that they allow GRIM to perform Goal Reasoning during execution, improve its performance, and enable it to successfully complete more goals under uncertain and changing conditions.

Associating the Goal Lifecycle strategies with a single metric, as is done in GRIM, can be useful for multiple reasons. For example, it can be used to define clear decision points that increase the transparency of the decision process used by the Goal Reasoning system. For a system that has as much autonomy as GRIM, which can change not only its plans but also its goals, transparency in how those decisions are made may help to promote operator trust.

4 Future Topics

Sections 2 and 3 described models of Goal Reasoning and their relation to Trusted Autonomy. In this section, we describe future extensions to these models. We begin with a discussion on inverse trust and its support of adaptive autonomy in Section 4.1, and then describe the concept of rebel agents and their relation to Trusted Autonomy in Section 4.2.

4.1 Adaptive Autonomy and Inverse Trust

The ASM and GRIM agents have certain properties that, arguably, have the potential to engender trust. For example, since the ASM agent continuously monitors the behavior of teammates, it can rapidly respond to any changes in their plans or goals. Similarly, since the GRIM agent monitors the progress of controlled vehicles with respect to goal constraints, it can automatically apply a *resolve* strategy when it recognizes that a constraint is projected to be violated. However, neither ASM nor GRIM agents use specific mechanisms to build or maintain operator trust.

Traditional computational trust metrics [35] are used to measure how much trust an agent has in another agent using information from past interactions or third-party feedback. These metrics allow an agent embedded in a human team to measure its

trust in other teammates but do not allow it to measure how trustworthy it is from its teammates' perspective. For this reason, we developed an *inverse trust metric* [15] that allows an agent to estimate its own trustworthiness. While many factors that influence human-robot trust are not directly observable to an agent (e.g., a teammate's experience with other agents, or a teammate's internal evaluation of an agent), factors that are observable, such as the agent's performance, have been found to have the greatest influence on trust [19].

The inverse trust estimate allows an agent to evaluate its own performance and use that information to estimate the corresponding influence on its trustworthiness (i.e., increasing, decreasing, constant). Based on this estimate, an agent can reason about whether its current behavior is trustworthy or untrustworthy. In situations where the agent believes its behavior is untrustworthy, it can modify its behavior in an attempt to learn (and apply) a more trustworthy behavior, thus implementing a form of adaptive autonomy. Preliminary studies in limited simulations have shown that an agent using an inverse trust method can successfully adapt its behavior given implicit feedback [15], and can benefit further from explicit feedback [16] as well as the ability to generate explanations when it modifies its behaviors [14].

The primary benefit of our approach is that it gives the agent control over maintaining trust and does not require an exhaustive engineering effort to develop behaviors that will be trustworthy for all teammates, in all environments, and in all contexts. However, we have not yet integrated it with a Goal Reasoning agent, and to date it has only been examined in the context of an agent whose goals are static. Our plans for future work include testing variants of inverse trust in a Goal Reasoning agent in environments in which the operator can specify a variety of goals to achieve, and where unexpected situations can arise (thus motivating the need for self-selection of goals or recommendation of goal changes to the operator). We expect that our studies will demonstrate the utility of adaptive autonomy in interactive Goal Reasoning agents.

4.2 Rebel Agents

Rebel agents [10] represent a relatively novel research direction in the context of Goal Reasoning. Rebel agents can object to, or even completely reject, goals or associated courses of action that are assigned to them by external agents (human or artificial), and they challenge the general attitudes or behaviors of those other agents. For example, an operator may command a rebel agent to pursue a specified goal without knowledge of the agent's context or access to its information sources, in which case the agent may respond with a recommendation for an alternative goal (along with an explanation).

Several situations exist in which modeling a rebel agent that can adjust its goals (or plans) can be viewed as beneficial, including the following:

- *Divergent information sources*: The agent may have access to information currently not available to the operator, which requires immediate action that is incompatible with the assigned goal.
- *Moral conflict*: The agent may be endowed with a “moral conscience” model that conflicts with an assigned goal. This can be a factor for protest in human-robot interaction [8].
- *Diversity*: The agent may be intended to contribute to the diversity of its team so as to ensure that sufficiently varied points of view and alternative goals are considered. This direction is inspired by studies claiming that diverse teams tend to outperform non-diverse teams [21, 37] under certain circumstances.
- *Self-assessment*: The agent may assess its assigned task as not being a good match for its capabilities. This relates to studies in personality psychology where the strengths-based leadership approach [32] argues that every person (i.e., leaders or other team members) should be offered the opportunity to routinely conduct activities in line with their strengths.
- *Believability*: An agent playing a character in an interactive narrative or training simulation may be given (and refuse) a goal that undermines its believability [7].

Several of these situations assume that the agent has an internal motivation model that conflicts with an assigned goal (or plan). This model can be based on many factors, such as simulated memory, emotion, or social relationships. The agent’s attitude towards a goal can change over time (due, for example, to changes in the environment or in the agent’s knowledge of an operator’s motivation), leading to incremental increase or decrease in the agent’s inclination to rebel.

In one definition of Trusted Autonomy, Abbass *et al.* [1] express trust in terms of vulnerability. A moment of rebellion is inherently one of vulnerability. By rebelling, an agent (1) makes itself vulnerable, and (2) creates vulnerability in the system it is (or was originally) part of. This anticipates some ways in which trust can be a factor for rebellion, such as the following:

- *Self-trust*: The agent’s amount of trust in itself (e.g., its trust that it can accurately assess the current situation as warranting opposition).
- *Perceived trust*: The agent’s model of how much other agents trust its judgment (e.g., based on its perceived expertise).
- *Risk*: The degree to which the agent trusts other agents to handle the vulnerabilities that rebellion creates for the rebel agent and the entire system.
- *Distrust*: The agent’s distrust of other agents (i.e., its belief that it cannot entrust its vulnerabilities to them if the goal is to be achieved).

The relation between rebellion and trust is multifaceted. Some ways in which rebellion can impact trust include:

1. Rebellion can diminish the trust of other agents in the rebel agent (e.g., an operator may lose trust in an agent that refuses to pursue an assigned goal/objective).
2. Rebellion can increase the trust of other agents in the rebel agent (e.g., a rebel agent that displays expertise when rejecting a goal, by raising objections when appropriate, may be more trusted to act autonomously).

3. The way in which other agents behave following a situation of rebellion can impact the trust of the rebel agent in those other agents.

In our future work on Goal Reasoning agents that can rebel, we plan to first test them in scenarios in which agents may rebel because their operator does not have complete access to their current state (including sensor data). We wish to model methods through which the agent learns how to explain its rebellion and negotiate with its operator so as to maximize its confidence that it is pursuing a well-justified goal. For example, this may involve soliciting an explanation from the operator as to why a goal should be pursued, or providing an argument for rejecting it.

5 Conclusion

The topic of Goal Reasoning is an important one in Robotics and Artificial Intelligence. Creating autonomous systems that can deliberate on and change their own goals allows those systems more freedom to intelligently adapt their behaviors to unexpected events and changing conditions. This chapter presented two different models of Goal Reasoning. First, we presented the GDA model and a related method for goal selection using motivators, as well as an application of GDA in a human-robot teaming task. We also presented a model based on goal refinement, instantiated in the Goal Lifecycle, and discussed an architecture for placing guarantees on the behavior of the agents as well as an application in multi-agent robotics. Finally, we discussed two ongoing extensions to our Goal Reasoning work: adaptive autonomy using inverse trust and the study of rebel agents.

Goal Reasoning also has a number of close connections to the topic of Trusted Autonomy. Goal Reasoning agents can modify and change their goals autonomously, in addition to adjusting how they achieve those goals. As such, any trust of such systems must inherently extend to their capability to reason at the goal level. Goal Reasoning also provides opportunities to design systems that cultivate trust through transparency or through goals that actively account for the trust of other agents.

Acknowledgements We would like to thank Mark Roberts and Thomas Apker for their assistance with this chapter, as well as much of the underlying research.

References

1. Hussein A. Abbass, Eleni Petraki, Kathryn Merrick, John Harvey, and Michael Barlow. Trusted Autonomy and Cognitive Cyber Symbiosis: Open Challenges. *Cognitive Computation*, 8(3):385–408, 2016.
2. David W. Aha, Michael T Cox, Héctor Muñoz-Avila, and (Eds.). Goal Reasoning: Papers from the ACS Workshop (Technical Report CS-TR-5029). University of Maryland, Department of Computer Science, College Park, MD, USA, 2013.

3. David W. Aha, Matthew Klenk, Héctor Muñoz-Avila, Ashwin Ram, Daniel Shapiro, and (Eds.). Goal-Directed Autonomy: Notes from the AAAI Workshop. AAAI Press, Atlanta, GA, USA, 2010.
4. David W. (Ed.) Aha. Goal Reasoning: Papers from the ACS Workshop (Technical Report GT-IRIM-CR-2015-001). Georgia Institute of Technology, Institute for Robotics and Intelligent Machines, Atlanta, GA, USA, 2015.
5. Thomas B. Apker, Benjamin Johnson, and Laura Humphrey. LTL Templates for Play-Calling Supervisory Control. In *Proceedings of AIAA Science and Technology Exposition*, 2016.
6. Gianluca Baldassare and Marco Mirolli, editors. *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 2013.
7. Joseph Bates. The Role of Emotions in Believable Agents. *Communications of the ACM*, 37(7):122–125, 1994.
8. Gordon Briggs and Matthias Scheutz. “Sorry, I can’t Do That”: Developing Mechanisms to Appropriately Reject Directives in Human-Robot Interactions. In *2015 AAAI Fall Symposium Series*, 2015.
9. Jessie Y. C. Chen, Michael J. Barnes, Anthony R. Selkowitz, Kimberly Stowers, Shan G. Lakhmani, and Nicholas Kasdaglis. Human-Autonomy Teaming and Agent Transparency. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 28–31, Sonoma, USA. ACM.
10. Alexandra Coman, Kellen Gillespie, and Héctor Muñoz-Avila. Case-Based Local and Global Percept Processing for Rebel Agents. In Joseph Kendall-Morwick, editor, *Proceedings of the 23rd Annual International Conference on Case-Based Reasoning, CEUR Workshop*, pages 23–32, 2015.
11. Michael T Cox. Goal-driven autonomy and question-based problem recognition. In *Proceedings of 2nd Annual Conference on Advances in Cognitive Systems*, pages 29–45, Baltimore, MD, USA, 2013.
12. Michael T Cox, Zohreh Alavi, Dustin Dannenhauer, Vahid Eyorokon, Héctor Muñoz-Avila, and Don Perlis. MIDCA: A Metacognitive, Integrated Dual-Cycle Architecture for Self-Regulated Autonomy. pages 3712–3718, 2016.
13. Dustin Dannenhauer and Héctor Muñoz-Avila. Raising Expectations in GDA Agents Acting in Dynamic Environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2241–2247, Buenos Aires, Argentina, 2015. AAAI Press.
14. Michael W. Floyd and David W. Aha. Incorporating Transparency During Trust-Guided Behavior Adaptation. In *Proceedings of the 24th International Conference on Case-Based Reasoning (To Appear)*, Atlanta, GA, USA, 2016. Springer.
15. Michael W. Floyd, Michael Drinkwater, and David W. Aha. How Much Do You Trust Me? Learning a Case-Based Model of Inverse Trust. In *Proceedings of the 22nd International Conference on Case-Based Reasoning*, pages 125–139. Springer, 2014.
16. Michael W. Floyd, Michael Drinkwater, and David W. Aha. Improving Trust-Guided Behavior Adaptation Using Operator Feedback. In *Proceedings of the 23rd International Conference on Case-Based Reasoning*, pages 134–148, Frankfurt, Germany, 2015. Springer.
17. Michael W. Floyd, Michael Drinkwater, and David W. Aha. Trust-Guided Behavior Adaptation using Case-Based Reasoning. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. AAAI Press, 2015.
18. Kellen Gillespie, Matthew Molineaux, Michael W. Floyd, Swaroop S. Vattam, and David W. Aha. Goal Reasoning for an Autonomous Squad Member. In David W. Aha, editor, *Proceedings of the Workshop on Goal Reasoning (held at the 3rd Conference on Advances in Cognitive Systems)*, pages 52–67. Georgia Institute of Technology, Institute for Robotics and Intelligent Machines, 2015.
19. Peter A. Hancock, Deborah R. Billings, Kristin E. Schaefer, Jessie Y.C. Chen, Ewart J. De Visser, and Raja Parasuraman. A Meta-analysis of Factors Affecting Trust in Human-Robot Interaction. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(5):517–527, 2011.
20. Nick Hawes. A Survey of Motivation Frameworks for Intelligent Systems. *Artificial Intelligence*, 175(5):1020–1036, 2011.

21. Lu Hong and Scott E Page. Groups of Diverse Problem Solvers can Outperform Groups of High-Ability Problem Solvers. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46):16385–16389, 2004.
22. Ulit Jaidee, Héctor Muñoz-Avila, and David W. Aha. Case-Based Goal-Driven Coordination of Multiple Learning Agents. In *Proceedings of the 21st International Conference on Case-Based Reasoning*, pages 164–178, Saratoga Springs, NY, USA, 2013.
23. Benjamin Johnson, Mark Roberts, Thomas Apker, and David W. Aha. Goal Reasoning with Information Measures. In *Proceedings of the 4th Annual Conference on Advances in Cognitive Systems*, Evanston, IL, USA, 2016.
24. Subbarao Kambhampati, Craig A Knoblock, and Qiang Yang. Planning as Refinement Search: A Unified Framework for Evaluating Design Tradeoffs in Partial-Order Planning. *Artificial Intelligence*, 76(1):167–238, 1995.
25. Matthew Klenk, Matt Molineaux, and David W Aha. Goal-Driven Autonomy for Responding to Unexpected Events in Strategy Simulations. *Computational Intelligence*, 29(2):187–206, 2013.
26. Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. Temporal-Logic-Based Reactive Mission and Motion Planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, December 2009.
27. Kathryn E. Merrick and Kamran Shafi. Achievement, affiliation, and power: Motive profiles for artificial agents. *Adaptive Behavior*, 19(1):40–62, 2011.
28. Matthew Molineaux, Matthew Klenk, and David W. Aha. Goal-Driven Autonomy in a Navy Strategy Simulation. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, Atlanta, GA, USA, 2010. AAAI Press.
29. Héctor Muñoz-Avila, David W. Aha, Ulit Jaidee, Matthew Klenk, and Matthew Molineaux. Applying Goal Driven Autonomy to a Team Shooter Game. In *Proceedings of the 23rd International FLAIRS Conference*, pages 465–470, Daytona Beach, FL, USA, 2010.
30. Dana Nau. Current Trends in Automated Planning. *AI Magazine*, 28(4):43–58, 2007.
31. Jay Powell, Matthew Molineaux, and David W. Aha. Active and Interactive Learning of Goal Selection Knowledge. In *Proceedings of the Twenty-Fourth Florida Artificial Intelligence Research Society Conference*. AAAI Press, 2011.
32. Tom Rath and Barry Conchie. *Strengths Based Leadership: Great Leaders, Teams, and Why People Follow*. Simon and Schuster, 2008.
33. Mark Roberts, Thomas Apker, Benjamin Johnson, Bryan Auslander, Briana Wellman, and David W. Aha. Coordinating Robot Teams for Disaster Relief. In *Proceedings of the 28th International FLAIRS Conference*, Hollywood, Florida, USA, 2015.
34. Mark Roberts, Daniel Borrajo, Michael T. Cox, Neil Yorke-Smith, and (Eds.). Goal Reasoning: Papers from the IJCAI Workshop. New York, NY, USA, 2016.
35. Jordi Sabater and Carles Sierra. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24(1):33–60, 2005.
36. Ron Sun. Motivational representations within a computational cognitive architecture. *Cognitive Computation*, 1(1):91–103, 2009.
37. James Surowiecki. *The Wisdom of Crowds*. Doubleday, 2004.
38. Swaroop Vattam, Matthew Klenk, Matthew Molineaux, and David W Aha. Breadth of Approaches to Goal Reasoning : A Research Survey. In David W. Aha, Michael T. Cox, and Hector Muñoz-Avila, editors, *Goal Reasoning: Papers from the ACS Workshop (Technical Report CS-TR-5029)*, page 111, College Park, MD, 2013. University of Maryland.
39. Ben G. Weber, Michael Mateas, and Arnav Jhala. Applying Goal-Driven Autonomy to StarCraft. In G. Michael Youngblood and Vadim Bulitko, editors, *Proceedings of the 6th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 101–106, Stanford, CA, USA, 2010. AAAI Press.
40. Ben G. Weber, Michael Mateas, and Arnav Jhala. Learning from Demonstration for Goal-Driven Autonomy. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1176–1182, Toronto, Canada, 2012.
41. Mark Wilson, James McMahon, Artur Wolek, David W. Aha, and Brian H. Houston. Toward Goal Reasoning for Autonomous Underwater Vehicles: Responding to Unexpected Agents.

42. Mark Wilson, Matthew Molineaux, and David W Aha. Domain-Independent Heuristics for Goal Formulation. In *Proceedings of 26th International FLAIRS Conference*, pages 160–165, St. Pete Beach, USA, 2013. AAAI Press.