Toward Goal Reasoning for Autonomous Underwater Vehicles: Responding to Unexpected Agents

Mark A. Wilson[1], James McMahon[2], Artur Wolek[3], David W. Aha[1], and Brian H. Houston[2]
[1]Adaptive Systems Section, Code 5514
[2,3]Physical Acoustics Branch, Code 7130
[3]ASEE Postdoctoral Fellow
Naval Research Laboratory, Washington, D.C.
[1,2]*first.last*@nrl.navy.mil | [3]*first.last*.ctr@nrl.navy.mil

Abstract

We describe preliminary work toward applying a goal reasoning agent for controlling an underwater vehicle in a partially observable, dynamic environment. In preparation for upcoming at-sea tests, our investigation focuses on a notional scenario wherein a autonomous underwater vehicle pursuing a survey goal unexpectedly detects the presence of a potentially hostile surface vessel. Simulations suggest that Goal Driven Autonomy can successfully reason about this scenario using only the limited computational resources typically available on underwater robotic platforms.

1 Introduction

Goal Reasoning (GR) is a form of planning and acting in complex (e.g., uncertain, dynamic, partially observable) domains. A GR agent can deliberate about and alter its own goals at appropriate times, such as when the environment behaves unexpectedly. The field of GR has received increased focus in recent years but most GR agents are not embodied in a robotic platform. Conversely, task-driven planning agents have been investigated for use in underwater and other unmanned vehicles (e.g., [Cashmore *et al*., 2015]). However, these agents do not employ GR techniques and therefore cannot adequately adapt to unexpected changes in their environment.

The *Goal Driven Autonomy* (GDA) model [Klenk *et al.*, 2013] of GR monitors plan execution for discrepancies between expected states and observations (i.e., processed sensor readings from the environment). When a discrepancy is detected, the agent constructs a consistent explanation, encompassing its observation history and the discrepancy, which may improve its knowledge of the world by inferring features of the state that cannot be directly observed.

We report on simulated and preliminary results for an application of GR to a robotic vehicle. We describe initial at-sea trials with a GDA agent controlling an Iver2 autonomous underwater vehicle (AUV) [OceanServer 2012]. We are testing the decision-making capabilities of this agent *in situ*, providing the AUV with challenges in the form of a simulated unmanned surface vehicle (USV) unexpectedly traversing the AUV's area of operations while the AUV performs its mission.

2 Goal Driven Autonomy

GDA (Figure 1) is a model for online planning with reasoning about goal formulation and management [Molineaux *et al.*, 2010]. It extends Nau's [2007] model of online planning, using the Controller to create and pursue new goals when unexpected events occur in complex domains.

The GDA Controller uses the Planner to create a plan to achieve the current goal g from the current state s_0. The Planner outputs to the Controller a sequence of actions $\langle a_1, \dots, a_n \rangle$ to execute, and a corresponding sequence of expected states $\langle x_1, \dots, x_n \rangle$, where x_n is a goal state for g.

As the Controller executes the plan in the state transition environment, it performs a four-step cycle to manage goals in response to unexpected events:

1. ***Discrepancy detection:*** After the Controller executes action a_i, the Discrepancy Detector compares the new observed state s_i to the

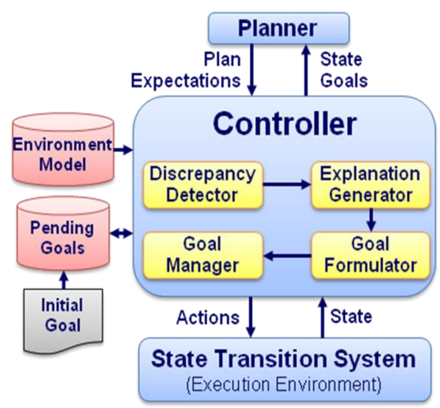

Figure 1: The Goal-Driven Autonomy Conceptual Model

corresponding expectation x_i. If they differ, a discrepancy has occurred and the GDA model attempts to explain and resolve it.

2. ***Explanation:*** If discrepancies between the new state and the expectation are detected, the Explanation Generator attempts to create an explanation of the discrepancies.
3. ***Goal formulation:*** The Goal Formulator creates new goals that are appropriate given the explanation.
4. ***Goal management:*** Finally, the Goal Manager prioritizes and selects among the Pending Goals, including new goals from the Goal Formulator. The selected goal is then given to the Planner to generate a new plan and expectations.

3 AUV Autonomy Model

In our control architecture, the GDA Controller monitors the AUV's state and directs it to perform sensing and navigation tasks, delegating them to lower-level control components. To address the challenges of motion control in dynamic environments that may be only partially known *a priori*, we employ the reactive MOOS-IvP autonomy architecture [Benjamin *et al*. 2010], a widely used, open source robotic control framework. MOOS is a message-passing suite with a centralized publish-subscribe model. The MOOS application IvP Helm is a behavior-based controller that sets navigation parameters to generate collision-free trajectories, using an interval programming technique that optimizes over the selected behaviors' objective functions.

The GDA Controller executes plans by activating, deactivating, and changing parameters of IvP Helm behaviors. (While IvP Helm can alter behaviors reactively, it cannot deliberate about what goal the vehicle should pursue, which is the focus of GDA.) Figure 2 depicts our agent architecture, which includes the Front Seat vehicle control system provided by the manufacturer (OceanServer).

For these trials, we employ: the PHOBOS planner [Wilson et al., 2014]; state comparison for discrepancy detection; a C++ implementation of DiscoverHistory [Molineaux and Aha, 2015] for explanation; rule-based scripts for goal formulation; and the goal manager selects the most recently-formulated goal.

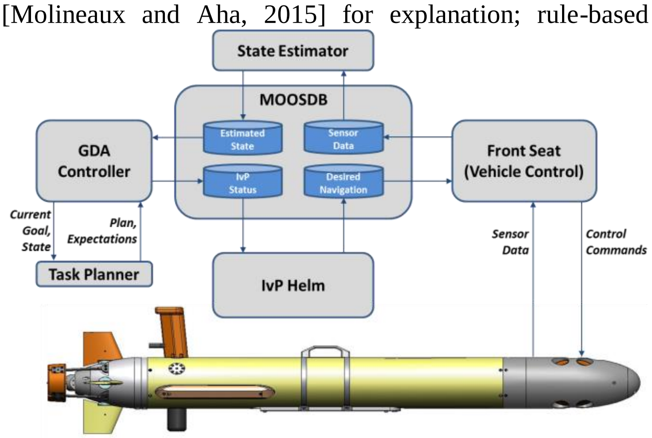

Figure 2: Our Iver2 AUV Autonomy System

4 Related Work

Much prior work in AUV control focuses on task-level planning that can be carried out by motion controllers. For instance, McMahon and Plaku [2016] generate dynamically-feasible collision-free motion plans that satisfying multiple goals specified using Regular Languages. Cashmore et al. [2014] describe a PDDL planner for underwater inspection tasks that can update its world model and replan when new information is discovered. Karpas et al. [2015] describe an extension to the Pike online executive that, while not focused on the AUV domain, permits a vehicle to relax temporal bounds on plan execution by teaming with a human operator. Unlike our approach, these do not incorporate a model of goal reasoning.

BDI architectures may provide reactive plan management facilities. For instance, the Jason interpreter for AgentSpeak [Bordini and Hübner 2006] permits an agent to select new plans in response to external events. Jason provides a focus on multiagent systems, whereas our system is focused on control of a single agent. Further, while Jason provides events that describe the most recent changes in the agent's beliefs or goals, our architecture provides explanatory reasoning, over the agent's history, to identify root causes of exogenous events and consequent changes to the agent's beliefs. Although we employ rule-based goal selection in this work, our GDA architecture provides the capability to select goals by deliberating using *motivators* [Wilson *et al.*, 2013], whereas Jason only provides rule-based nomination from a plan library.

Goal reasoning has been applied in a number of domains, including Tactical Action Officer decision making [Molineaux *et al*., 2010], the game of Starcraft [Weber *et al*., 2012], interactive storytelling [Coman *et al.*, 2015], and cyber defense [Goldman *et al.*, 2015]. Studies of specific techniques in goal reasoning have focused on, among others, expectation generation [Dannenhauer and Muñoz-Avila, 2015], goal formulation [Wilson *et al.*, 2013, Jaidee *et al*., 2011], explanation generation [Molineaux and Aha, 2015], goal prioritization [Young and Hawes, 2012], plan recognition incorporating goal knowledge [Vattam and Aha, 2015], and formal models of goal reasoning as a process [Roberts *et al.*, 2014, Cox 2015]. Our focus in this paper is on AUV control.

Goal reasoning has had some limited application for robotic control. Roberts et al. [2015] examine the use of goal reasoning for coordinating teams of robots in disaster recovery scenarios. Wilson *et al.* [2014] propose a planner and expectation model for AUV control with goal reasoning. Cox et al. [2016] incorporate a ROS interface into the MIDCA metacognitive architecture for communication with a Baxter robot. In contrast, we plan to conduct at-sea trials with the Iver2 robot under GR control.

5 Demonstration

The purpose of this exercise is to demonstrate our GDA agent's ability to control an AUV in a basic scenario involving unexpected observations during execution.

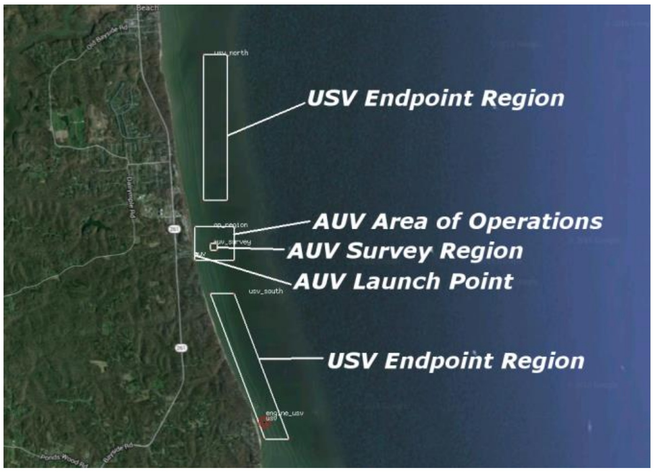

Figure 3: An Example Problem for the AUV Test Domain

5.1 Vehicle

The target vehicle for our initial demonstrations is an OceanServer Iver2 AUV. The Iver2 vehicle is a low-cost lightweight torpedo shaped AUV. It has a diameter of 14:7 cm, a length of 127 cm, weighs approximately 19kg (depending on the sensor configuration), and has a maximum speed of about 2.0 m/s. To match the MOOS-IvP simulator's dynamics with this vehicle, we use the default MOOS-IvP values for buoyancy rate, max acceleration and deceleration, depth rate, and turn rate. These values have been shown to accurately capture the dynamics of the AUV in field trials [McMahon and Plaku, 2016].

5.2 Domain

The agent's PDDL domain description includes: the vehicle's location, depth, speed, and heading; notional processed input from passive sonar sensors (classified as "engine noise" or "active pings"); actions for traversing to a waypoint and surveying a region (causes the vehicle to execute surveying motions, but does not presently engage an active sonar or receive sonar responses); and exogenous events for various changes in state owing to the reactive control layer or the environment. For clarity we do not present an exhaustive list of actions or events and describe only those that are relevant to the goal reasoning process.

We are conducting a series of trials, wherein the vehicle is tasked with surveying a pre-defined region (the mission goal) and then returning to a "home point" once the goal is complete. A "safe point" is defined as a location to which the vehicle can retreat if threatened. A simulated second vehicle with randomized start and end points may threaten the AUV and cause it to retreat if the AUV detects its presence. These elements are illustrated by the example problem shown in Figure 3. This example depicts the AUV's area of operations, the survey region, the AUV launch point, and the endpoint regions for the simulated USV.

Goal Name	Parameters	Purpose
go-home		Drives AUV to launch point
go-to-point	?x (x-coord) ?y (y-coord)	Drives AUV to specified coords
sweep-area	?s (predefined area) ?l (lane width)	Performs a survey in specified region
wait		Do nothing

Table 1: Available agent goals for the test mission

5.3 Mission Description

The AUV starts at its launch point near the shoreline at the Naval Research Laboratory's Chesapeake Bay Detachment (Figure 3). Its initial goal is to perform a simulated survey of the ocean floor in a target location. The AUV's speed is limited to 1.5 m/s in this scenario. Available goals for the AUV are shown in Table 1, and its initial state s_0 is described in Table 2. The "ranged" values in Table 2 are provided via the PHOBOS planner's ability to produce acceptable ranges of expected values (for each action), which decreases false discrepancy rates.

The simulated USV starts at a randomly-chosen point in one of two endpoint regions, located along the shoreline north and south of the AUV's start position (Figure 3). The USV's start and end points are chosen from a uniform distribution over these areas. The USV's maximum speed is 5 m/s.

The USV is either in a "hostile" mode, using active sonar pings which would correspond to a vessel seeking the AUV, or in "neutral" mode, not emitting active sonar. Simulated

Fluent Name	Parameters	Value
underway		false
mission-started		false
area-surveyed	area1	false
home-location	5, 0	true
x-pos		(range -10 20)
y-pos		(range -15 15)
speed		(range 0 1)
heading		(range 0 360)
depth		(range 0 0.5)
safe-point-location	50, -50	true
engine-noise		false
ping-noise		false
contact-in-area		false
hostile-contact-in-area		false

Table 2: Initial state for the test mission (x-y coordinates are aligned with the cardinal directions and measured from the launch point).

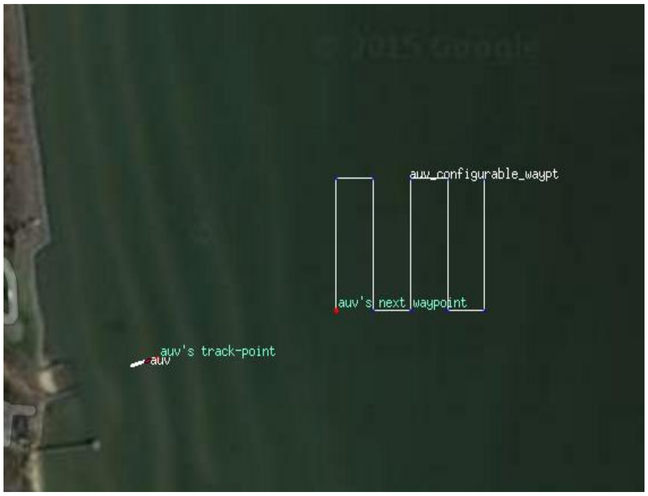

Figure 4: AUV transiting to survey area

pings are emitted once per second and have a detectable radius of 60m. Whether the USV is hostile or neutral, it emits engine noise with a detectable radius of 80m. This choice of endpoint region and USV speed results in the USV being (potentially) detected at a different time and relative position in each trial. The aim of introducing such variability is to demonstrate the robustness of the GDA Controller to different discrepancies.

The AUV transits from its launch point to a 100x100m area approximately 245m east and 100m north of the launch point, and perform a survey over this area (Figure 4).

The simulated surface vessel transits through the center of the AUV's survey area, then to the randomly-chosen point in the endpoint region it did not start in (Figure 5). When it is within engine-noise range of the AUV, the simulator reports the engine noise to the GDA agent, causing the fact

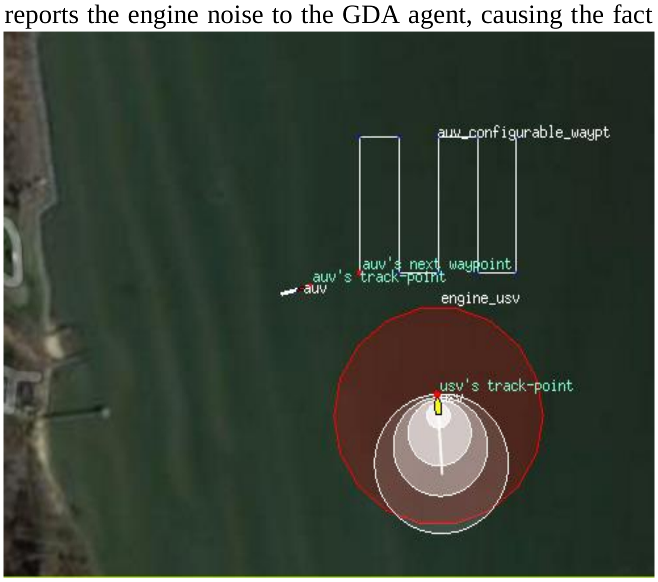

Figure 5: Surface vessel transiting through AUV search area. The red circle indicates the range of the vessel's engine noise, and the white circles represent active pings from the vessel.

Figure 6: The AUV returning to a safe point after detecting the surface vessel's pings

engine-noise to be true in the agent's next observation. The unexpected fact *engine-noise* will trigger the GDA process, causing the following sequence:

- The Explainer resolves the discrepancy by adding a *contact-entered-area* event to the agent's history, which also sets *contact-in-area*. For a detailed discussion of explanation generation refer to [Molineaux and Aha, 2015].
- The Goal Formulator (rule-based for this test) evaluates the current state with explanation revisions. The fact *contact-in-area* is now true, but has no bearing on goal selection in this mission. The goal *sweep-area* remains active.
- The Goal Manager, in this mission, simply accepts the goal output by the Formulator as the new or continuing active goal.

A similar sequence occurs if the USV is in "hostile" mode, and its simulated pings are detected by the AUV. The agent's new observation will unexpectedly include the *ping-noise* fact, causing the agent to reason about the discrepancy.

- The Explainer resolves the discrepancy by inserting a *hostile-contact-entered-area* event in the agent's history, which also sets *hostile-contact-in-area*.
- The Goal Formualtor evaluates the revised current state, including the *hostile-contact-in-area* fact, which is associated with the *go-to-point* goal with the coordinates from the *safe-point-location* fact.
- The Goal Manager accepts the new goal and the agent constructs a new plan to execute it.

For this demonstration, the safe point is set near the launch point to prevent the AUV from driving any large distance from the launch point, which could complicate recovery in the event of an error (Figure 6).

When the USV is no longer within ping detection range, the simulator stops reporting the noise to the AUV, causing the *ping-noise* fact to be unexpectedly false. The GDA agent resolves this discrepancy by adding a *hostile-contact-exited-area* event to the history, removing the *hostile-contact-in-area* fact from the current state, and re-formulating the *sweep-area* goal.. After the *sweep-area* goal is complete, the AUV formulates a new goal, *go-home*, to return to the launch point.

6 Simulated Results

To demonstrate the robustness of the GDA Controller to varying discrepencies we conducted 25 simulated trials of the mission described in Section 5.3 The trials were executed at 3x real time using the MOOS-IvP uSimMarine and pMarinePID utilities to simulate the mission in place of the vehicle's Front Seat controller (Figure 3). The USV's hostility was selected at random for each trial. We observed that the AUV correctly responded to the unexpected USV in 100% of cases, retreating to the "safe point" when the USV's pings were detected in "hostile" mode (10 trials), or continuing its mission despite the unexpected engine noise when the USV was in "neutral" mode (15 trials). Additionally, the AUV correctly responded by resuming its survey mission after the USV was out of detection range in 100% of "hostile" trials. No additional discrepancies were generated. Simulations were run in a virtual machine using an Intel Core i7-4800MQ CPU with 8 GB of RAM available. Although this hardware is significantly more powerful than that available on the Iver2 (see Section 7), we did not observe the GDA process consuming more than 10% of CPU cycles even at 3x real time simulation. These results suggest that the GDA Controller can successfully reason about the scenario considered herein using only the limited computational resources typically available on underwater robotic platforms.

7 Discussion

We presented preliminary results demonstrating the ability of our agent to control a simulated autonomous underwater vehicle using Goal Driven Autonomy. Currently, we are using an Iver2 vehicle for at-sea tests that replicate the simulation. We have installed and tested our agent software on the vehicle's processor (1.4 GHz Pentium-M, 1 GB RAM), and we have conducted initial at-sea exercises at the same Chesapeake Bay location used in the simulated tests.

In our initial at-sea tests, we have found that the goal reasoner can formulate goals and execute plans based on user input (start mission) and recognized completion of prior goals (return home after surveying a region). Figure 7 depicts a visualization of data collected during an at-sea test, as the vehicle returns to its starting point after completing a survey. We anticipate that, after we resolve certain robotics challenges, our goal reasoner will correctly formulate goals in response to simulated changes in its environment. Successful tests will be used to inform our efforts as we proceed toward more ambitious at-sea trials.

Although we chose a relatively simple mission for our initial demonstration of at-sea capabilities, we look forward to incorporating more realistic and complex challenges into our missions. For example, these will include noise in sensor models, maritime sources of interference, the use of real (vice simulated) sensors, the need to avoid collisions with non-simulated neutral or friendly vehicles, and the application of more advanced behaviors such as classification or following of other vehicles.

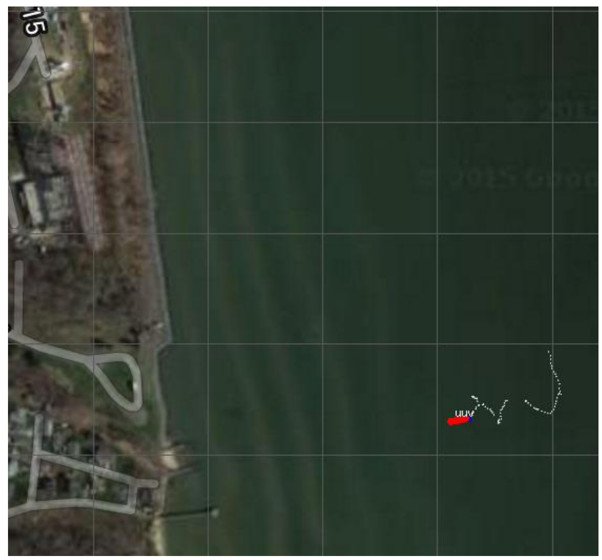

Figure 7: The AUV returns to its launch point after completing a survey (at-sea data).

References

[Benjamin *et al.*, 2010] Michael R. Benjamin., Henrik Schmidt, Paul M. Newman, and John J. Leonard. Nested autonomy for unmanned marine vehicles with MOOS-IvP. *Journal of Field Robotics* 27, 834-875, 2010.

[Bordini and Hübner, 2016] Rafael H. Bordini and Jomi F. Hübner. BDI agent programming in AgentSpeak using Jason. In *Proceedings of the Sixth International Workshop on Computation Logic in Multi-Agent Systems*. London, UK: Springer-Verlag, 2006.

[Cashmore *et al.*, 2015] Michael Cashmore, Maria Fox, Derek Long, Daniele Magazzeni, Bram Ridder, Anau Carrera, Narcis Palomeras, Natalia Hurtos, and Marc Carreras. ROSPlan: Planning in the Robot Operating System. In *Proceedings of the Twenty-fifth International Conference on Automated Planning and Scheduling.* Jerusalem, Israel: AAAI Press, 2015.

[Cashmore *et al*., 2014] Michael Cashmore, Maria Fox, Tom Larkworthy, Derek Long, and Daniele Magazzeni. AUV mission control via temporal planning. In *Proceedings of the International Conference on Robotics and Automation*. Hong Kong, China: IEEE Press, 2014.

[Coman *et al*., 2015] Alexandra Coman, Kellen Gillespie, and Héctor Muñoz-Avila. Believable emotion-influenced perception: The path to motivated rebel agents. In *Goal Reasoning: Papers from the ACS Workshop*. Atlanta, GA: Georgia Institute of Technology, Institute for Robotics and Intelligent Machines, 2015.

[Cox *et al*. 2016] Michael T. Cox, Zohreh Alavi, Dustin Dannenhauer, Vahid Eyorokon, Hector Muñoz-Avila, and Don Perlis. MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In *Proceedings of the Thirthieth AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press, 2016.

[Cox 2015] Michael T. Cox. Toward a formal model of planning, action, and interpretation with goal reasoning. In *Goal Reasoning: Papers from the ACS Workshop*. Atlanta, GA: Georgia Institute of Technology, Institute for Robotics and Intelligent Machines, 2015.

[Dannenhauer and Muñoz-Avila, 2015] Dustin Dannenhauer and Héctor Muñoz-Avila. Raising expectations in GDA agents acting in dynamic environments. In *Proceedings of the Twenty-fourth International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina: AAAI Press, 2015.

[Goldman *et al*., 2015] Robert P. Goldman, Mark Burstein, Ugur Kuter, Paul Robertson, and Dan Cerys. Knowledge representation for dynamic formulation of sensing goals in cyber defense. In *Goal Reasoning: Papers from the ACS Workshop*. Atlanta, GA: Georgia Institute of Technology, Institute for Robotics and Intelligent Machines, 2015.

[Jaidee *et al*., 2011] Ulit Jaidee, Héctor Muñoz-Avila, and David W. Aha. Integrated learning for goal-driven autonomy. In *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence*. Barcelona, Spain: AAAI Press, 2011.

[Karpas *et al*., 2015] Erez Karpas, Steven J. Levine, Peng Yu, and Brian Williams. Robust execution of plans for human-robot teams. In *Proceedings of the Twenty-fifth International Conference on Automated Planning and Scheduling*. Jerusalem, Israel: AAAI Press, 2015.

[Klenk *et al*., 2013] Matthew Klenk, Matt Molineaux, and David W. Aha. Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence*, 26(2), 187-206, May 2013.

[McMahon and Plaku, 2016] James McMahon and Erion Plaku. Mission and motion planning for autonomous underwater vehicles operating in spatially and temporally complex environments. *Journal of Oceanic Engineering*, PP(99), 1-20, 2016.

[Molineaux *et al*., 2010] Matt Molineaux, Matthew Klenk, and David W. Aha. Goal-driven autonomy in a Navy strategy simulator. In *Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press, 2010.

[Molineaux and Aha, 2015] Matthew Molineaux and David W. Aha. Continuous explanation generation in a multi-agent domain. In *Proceedings of the Third Conference on Advances in Cognitive Systems*. Atlanta, GA: Cognitive Systems Foundation, 2015.

[Nau, 2007] Dana S. Nau. Current trends in automated planning. *AI Magazine*, 28(4), 43–58, December 2007.

[OceanServer, 2012] OceanServer Technology Inc., http://www.iver-auv.com: Iver2 Autonomous Underwater Vehicle

[Roberts *et al*., 2014] Mark Roberts, Swaroop Vattam, Ronald Alford, Bryan Auslander, Justin Karneeb, Matthew Molineaux, Tom Apker, Mark Wilson, James McMahon, and David W. Aha. Iterative goal refinement for robotics. In *Planning and Robotics: Papers from the ICAPS Workshop*. Portsmouth, NH: AAAI Press, 2014.

[Vattam and Aha, 2015] Swaroop S. Vattam and David W. Aha. Case-based plan recognition under imperfect observability. In *Proceedings of the Twenty-third International Conference on Case-Based Reasoning*. Frankfurt, Germany: Springer, 2015.

[Weber *et al*., 2012] Ben G. Weber, Michael Mateas, and Arnav Jhala. Learning from demonstration for goal-driven autonomy. In *Proceedings of the Twenty-sixth AAAI Conference on Artificial Intelligence*. Toronto, Ontario, Canada: AAAI Press, 2012.

[Wilson *et al*., 2013] Mark Wilson, Matthew Molineaux, and David W. Aha. Domain-independent heuristics for goal formulation. In *Proceedings of the Twenty-sixth Florida Artificial Intelligence Research Society Conference*. St. Pete Beach, FL: AAAI Press, 2013.

[Wilson *et al*., 2014] Mark A. Wilson, James McMahon, and David W. Aha. Bounded expectations for discrepancy detection in goal-driven autonomy. In *AI and Robotics: Papers from the AAAI Workshop*. Quebec City, Quebec, Canada: AAAI Press, 2014.

[Young and Hawes, 2012] Jay Young and Nick Hawes. Evolutionary learning of goal priorities in a real-time strategy game. In *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Entertainment*. Palo Alto, CA: AAAI Press, 2012.