Towards Applying Goal Autonomy for Vehicle Control

Mark Wilson MARK.WILSON@NRL.NAVY.MIL
Naval Research Laboratory, Navy Center for Applied Research in AI, Washington, DC 20375

Bryan Auslander BRYAN.AUSLANDER@KNEXUSRESEARCH.COM
Knexus Research Corporation, 9120 Beachway Lane, Springfield, VA 22153

Benjamin Johnson BLJ39@CORNELL.EDU
Cornell University, School of Mechanical and Aerospace Engineering, Ithaca, NY 14853

Thomas Apker THOMAS.APKER@EXELISINC.COM
Exelis Inc., 2560 Huntington Ave, Alexandria, VA 22303

James McMahon JAMES.MCMAHON@NRL.NAVY.MIL
Naval Research Laboratory, Physical Acoustics, Code 7130, Washington, DC 20375

David W. Aha DAVID.AHA@NRL.NAVY.MIL
Naval Research Laboratory, Navy Center for Applied Research in AI, Washington, DC 20375

Abstract

Unmanned vehicles have been the focus of active research on autonomous motion planning, both deliberative and reactive. However, they are fundamentally limited in their autonomy by an inability to independently reason about, prioritize, and change the goals they pursue. We describe two new projects in which we are incorporating *goal autonomy* on unmanned vehicle platforms. We will apply the Goal-Driven Autonomy (GDA) model to permit our vehicles to reason about their objectives and discuss how properties of the domains affect the application of GDA.

1. Introduction

Unmanned vehicles are often used to explore and act in regions that are dangerous or otherwise undesirable for humans to visit. Many unmanned vehicles are remotely operated: Rather than acting autonomously using onboard control systems, they act directly on control commands from human operators to execute their missions. Remote operation may be desirable in some circumstances (e.g., to maximize control over the safety of an unusually valuable vehicle, such as a Mars rover). However, in many instances we would prefer that unmanned vehicles operate without human input, which would reduce operator load, avoid human error in operating the vehicles, and allow the vehicles to continue pursuing their missions when out of contact with human operators.

Most efforts to provide greater autonomy for unmanned vehicles have focused on a problem we refer to as *motion autonomy*, the primary example of which is to navigate autonomously to a desired location or to follow a prescribed route (e.g., Tan, Sutton, & Chudley, 2004; Wooden et al., 2010). Although motion autonomy techniques are broadly adaptable and allow robotic vehicles to autonomously accomplish many desired tasks, they do not allow vehicles to dynamically self-select goals to pursue or to re-prioritize their existing goals. This limits motion autonomy to predictable environments, as changes in the environment or previously unobserved facts may require an agent to select new objectives or mission parameters to act correctly.

To address this, we describe two new efforts to enrich unmanned vehicles' reasoning with *goal autonomy*: the ability to dynamically formulate, prioritize, and assign goals[1]. Enabling the vehicle to decide what goal it should accomplish in any given situation, in addition to existing techniques for achieving those goals autonomously, allows the vehicle to act correctly in a broader range of situations without supervision. This is especially valuable in long-duration missions in dynamic environments, where the vehicle is likely to encounter a variety of situations too complex to enumerate *a priori*. For instance, a maritime vehicle on a long mission may encounter a broad range of underwater hazards and opportunities for investigation in unpredictable configurations. To provide the ability to select appropriate goals in a wide range of situations, we will apply *Goal-Driven Autonomy* (GDA), a model for responding to unexpected occurrences by formulating and reprioritizing goals (Molineaux, Klenk, & Aha, 2010a).

In one project, *Autonomous Behavior Technology for Unmanned Underwater Vehicles*, we will apply the GDA model to an underwater vehicle, providing it the decision-making ability necessary to conduct long duration, independent missions with varying objectives. In another project, *Autonomous Systems Integration*, we will apply the GDA model to the task of plume-tracking, in which ground and air vehicles must cooperate to discover the source of an airborne contaminant, while also collecting and transferring power to avoid disruption of activity from loss of battery reserves.

GDA has previously been applied in several simulated test domains inspired by real-world scenarios (Molineaux et al., 2010a) as well as game environments (Weber, Mateas, & Jhala, 2012; Jaidee, Muñoz-Avila, & Aha, 2013). However, the projects presented here, although currently in simulation, will be our first application of GDA on real-world robots or vehicles.

In this paper, we present an overview of GDA, discuss the parameters of the application domains, present initial architectures for both projects, and discuss aspects of applying goal autonomy to situated agents and integrating goal autonomy with motion autonomy in two very different problem domains.

2. An Overview of Goal-Driven Autonomy

Goal-Driven Autonomy (GDA) (Figure 1) is a model for online planning with reasoning about goal formulation and management (Molineaux et al., 2010a). It extends Nau's (2007) model of online planning, using the Controller to create and pursue new goals when unexpected events occur in complex environments (e.g., stochastic, partially-observable).

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 $< a_1, \ldots, a_n >$ to execute, and a corresponding sequence of expected states $< x_1, \ldots, x_n >$, where x_n is a goal state for g.

[1] We use "goal autonomy" rather than "goal reasoning" throughout, to distinguish from "motion autonomy."

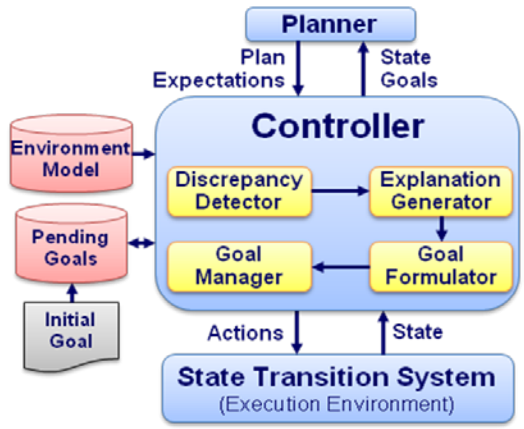

Figure 1: The Goal-Driven Autonomy (GDA) conceptual model.

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

Related work on autonomy focuses on the areas of goal autonomy, which addresses management of the agent's objectives, and motion autonomy, which addresses tasks such as safely moving a vehicle from one position to another.

Although the projects presented here represent our first efforts to use the GDA model on situated vehicles, GDA has been used in the past to control simulated agents. The ARTUE agent has been used to guide simulated vehicles inspired by Mars rovers (Wilson, Molineaux, & Aha 2013) as well as teams of simulated naval vessels (Molineaux et al., 2010a), but has never been integrated with dynamic motion controllers for real robots. EISBot (Weber et al., 2012), GRL (Jaidee, Muñoz-Avila, & Aha, 2012), and GDA-C (Jaidee et al., 2013) have all been used to successfully control all or part of a player's forces in real-time strategy games, a form of centralized direction for multi-agent systems. We present an architecture for centralized direction, but our system must interface with group control algorithms designed to prevent collisions while allowing several agents to work toward a common goal.

Other types of goal autonomy have also been used to control simulated agents. The ICARUS cognitive architecture (Choi, 2011) has been applied to simulated car-driving domains with a reactive goal-management component that introduces new goals taken from a long-term goal memory, given general and domain-specific conditions. Coddington's (2006) MADBot architecture, which can introduce new goals when domain-specific motivational thresholds are exceeded, has been used to control simulated ground-vehicle robots.

Goal autonomy systems have also received attention on robotic platforms. Dora the Explorer (Hawes et al., 2011) is a robot with goal autonomy capabilities, but is limited to goals focused on exploring and categorizing its environment. The SapaReplan planner has been used in the DIARC robotic-control architecture (Schermerhorn et al., 2009) to allow a robotic agent to optionally pursue *soft goals* by taking advantage of ungrounded opportunities in the environment, which it models using simulated objects called *counterfactuals*. However, SapaReplan can pursue such soft goals only temporarily and must not allow them to interfere with its required hard goals. This contrasts with our use of GDA, which permits the indefinite suspension of goals.

An alternative means of encoding multiple objectives onto an autonomous platform is the use of correct-by-construction controller synthesis. Kress-Gazit, Fainekos, and Pappas (2009) present a technique for specifying multiple goals and the conditions required to achieve them as *Linear Temporal Logic* (LTL) formulas. These formulas are used to generate a *Finite-State Automaton* (FSA) controller that is guaranteed to eventually accomplish all specified goals, assuming the required conditions are met and the environment meets defined expectations. However, the computational cost of constructing the FSA grows exponentially with the number of goals and conditions, and requires pre-specification of goals for all situations in which the robot must act. Thus, for large problems this framework requires a goal manager to provide a receding horizon for the controller as in (Wongpiromsarn, Topcu, & Murray, 2009). Livingston, Murray, and Burdick (2012) and Sarid, Xu, and Kress-Gazit (2012) introduce limited forms of goal formulation that respond competently to unexpected states and surprising opportunities, respectively, for synthesized controllers. Using controllers generated from LTL formulas will allow a task planner to plan atomic actions that can be decomposed into multiple LTL-level goals, and ensure that agents that are assigned complex, multi-stage tasks will complete them or provide information about unexpected states in the environment.

Approaches to autonomous control for underwater vehicles can be broadly classed into deliberative and reactive motion planning. Deliberative approaches variously use, among others, genetic algorithms (Alvarez, Caiti, & Onken, 2004), rapidly-exploring random trees (Tan et al., 2004), A* search over discretized environments (Garau, Alvarez, & Oliver, 2005), and gradient-descent optimization over cost functions (Kruger, Stolkin, Blum, & Briganti, 2007). Plaku and McMahon (2013) address simultaneous task and motion planning for underwater vehicles using LTL task specifications with sampling-based deliberative methods to avoid the complexity of guaranteed correctness. Reactive, or local, planning approaches are particularly useful in regions that are large or not well-mapped. Virtual potential fields (Khatib, 1985) are a common reactive system. Antonelli et al. (2001) alleviate the risk of this approach "trapping" a vehicle in local minima by adding a supervisor module to modify the vehicle's behavior based on the environment's geometry. While most of these approaches assume holonomic vehicle models, Apker and Potter (2012) describe a means of encoding a vehicle's dynamic constraints to improve performance and reliability. However, unlike our work, these systems address motion autonomy rather than the problem of goal autonomy.

The IvP Helm (Benjamin et al., 2010) provides a reactive UUV controller based on multi-objective optimization rather than potential fields, and exhibits limited goal autonomy by changing *modes* based on the state. However, it does not reason about goals the vehicle should accomplish in the environment.

Research on autonomy for individual air and ground vehicles is more mature than for underwater vehicles, and recent work has focused on guiding groups of vehicles to accomplish given tasks. Several authors have explored combining potential fields with FSAs to allow their systems to react to state changes by changing agent objectives. Mather and Hsieh (2012) apply this approach to robots engaged in surveillance tasks. Worcester, Rogoff, and Hsiehm (2011) develop a finite state representation of a construction task, and use a centralized system to partition its components among a team of robots. Martinson and Apker (2012) describe a physics-inspired FSA that operates in the robots' behavior space, changing the way they generate motion commands from potential fields depending on their proximity to a target and navigation quality. In contrast to this body of work, we instead focus on goal autonomy, and discuss applications of these methods to teams of unmanned vehicles in Section 5.

4. Application Domains

4.1 Long-Duration Underwater Autonomy

Autonomously-controlled unmanned underwater vehicles (UUVs) have been used for underwater exploration (Antonelli et al., 2001), observation and inspection of underwater structures (Antonelli et al., 2001), scientific observation (Binney, Krause, & Sukhatme, 2010), and mine countermeasures (LePage & Schmidt, 2002). However, these missions typically are of short duration (at most eight to sixteen hours) and operate over a small region.

In our first project we will apply GDA to autonomously direct a UUV on unsupervised long-duration missions. These missions could eventually last weeks or months. Long-term missions may require the vehicle to pursue different goals at different times, such as goals related to transiting to a region, avoiding other vessels, surveying oceanic geography, detecting mines and other manufactured obstacles, and taking oceanographic measurements. The ocean environment is highly unpredictable, and a UUV on a long-duration mission must be able to react intelligently to unexpected events and objects. Throughout the course of a mission a UUV may need to change its objectives, or even abort its mission, due to unforeseen environmental hazards, underwater barriers, encounters with other vehicles, or failures of onboard systems.

These missions may motivate goal autonomy. Although motion autonomy could correctly guide the vehicle on any task selected in response to such anomalies, goal autonomy provides the ability to select goals generally and dynamically without reference to a human operator. Because an at-sea UUV has very limited communication with human operators, the vehicle must make goal decisions autonomously.

For example, consider a UUV taking oceanographic measurements (e.g., water salinity) over a region, when a surface vessel enters its area and stops. If the measurements are being taken near the ocean surface, attempting to take them at or near the new vessel's position may risk collision. While motion autonomy systems can likely minimize risk and maximize data quality, they cannot consider the broader implications of the vessel's arrival and how best to respond. If it is a friendly vessel, it may be appropriate for the UUV to surface, broadcast that scientific measurements are being taken, and request that the vessel vacate the area. If the UUV is a military vehicle operating in contested or unfriendly waters, and the vessel is not friendly, it may be appropriate to halt and silence the UUV to avoid detection. If in open waters, the UUV may be correct to abort the data-collection mission and notify its operator of the surface vessel's approach. Goal-driven autonomy is a general model for generating appropriate responses to unplanned situations, and is therefore well-suited to the control of unmanned vehicles at sea.

Key challenges in this domain include:

- ***Unpredictable environments:*** Existing deliberative motion autonomy techniques for UUVs require advance knowledge of the environment in which the path will be planned while existing reactive motion autonomy techniques respond to unknown environments unpredictably. Both present challenges in long duration missions where a UUV may venture into waters that are not well-charted or for which there are no reliable data on currents. Furthermore, deliberative techniques have difficulty planning for dynamic obstacles whose motion may not be well understood, while reactive techniques can complicate the task of detecting discrepancies that occur during motion plan execution.
- ***Computational constraints:*** The CPUs that our agent will use to control the UUV are not powerful, and necessitate an emphasis on computationally efficient solutions.
- ***Uncertain environment state:*** The lack of many sensors often found on ground vehicles and other robots (e.g., for localization, visual inspection, range-finding), combined with noisy readings from sensors that are available, presents unique challenges.

4.2 Airborne Contaminant Detection

Unmanned air vehicles (UAVs) are used in remote sensing, scientific research, and search-and-rescue applications. Unmanned ground vehicles (UGVs) can be used to explore and act in situations that are dangerous to humans, such as in contaminated waste cleanup and explosive ordnance disposal missions, and to provide logistics support, such as carrying equipment.

In our second project, we will apply GDA to direct a team of UAVs equipped with aerosol sensors and UGVs with support equipment that includes landing pads, UAV rechargers, and solar panels. We know that the environment is bounded and that autonomous navigation is possible, but make no assumptions about initial plume locations, availability of traversable paths for the UGVs, or locations of brightly lit areas for solar recharging. This problem combines motion planning, task scheduling, and resource allocation in an unknown environment.

Conventional motion autonomy methods require a complete output specification for each vehicle given possible sensor inputs. In our scenario this is computationally intractable given the potential number of vehicles, sensors, and actions. Using GDA to make goal and task level decisions permits the synthesis of controllers that encode a limited number of relevant responses given the current goal, thus making the motion autonomy problem tractable.

Unlike the UUV domain, in the UAV/UGV domain we must control several vehicles to cooperatively achieve goals. However, if goal decisions are decentralized among vehicles, each vehicle would need to model all its teammates' possible goals and plans, or risk interference with teammates pursuing different goals. By centralizing GDA to coordinate the vehicles, we can guarantee all vehicles will pursue the same goal at any given time, and that the goal will be achieved based on guarantees offered by lower-layer controllers. This leads to the key challenges for GDA implementation in this domain:

- ***Motion abstraction:*** The GDA Controller must direct multiple autonomous vehicles to accomplish tasks requiring solutions to continuous-motion problems. Multiple vehicles must autonomously carry out these tasks without interfering with each other, a problem too computationally intensive to solve at the GDA level. Hence, we require abstract representations of the continuous motion problems that are suitable for computation at the goal autonomy layer, while supporting goal decisions that can be used as a basis for planning and controller synthesis for individual vehicles.

- ***Individual discrepancies:*** Although vehicles are directed in coordinating teams to achieve goals, discrepancies can still occur on the individual level (e.g., one vehicle's battery may run low due to malfunction). Our solution must manage goals and vehicle task assignments to permit responses to each vehicle's discrepancies, while using abstracted representations of goals as team activities that can be continued in spite of individual discrepancies.

5. Applying Goal-Driven Autonomy

GDA is well-equipped for its usual role in providing goal autonomy in task-planning domains. However, applying GDA in robotic vehicle domains requires appropriate abstractions from motion guidance to task-level actions. In this section we describe different approaches to this multi-layered abstraction in our underwater autonomy and airborne contaminant domains.

Factors such as environment predictability and the need for cooperation affect how GDA should be implemented and applied in a given domain. For single vehicles operating in dynamic or poorly specified environments (e.g., Mars rovers or singleton UUVs), each sense-act cycle represents an opportunity to reevaluate and adjust the agent's goals with respect to the most recent state. Loosely coordinated teams, particularly those working closely with humans, benefit from a concurrent control and planning architecture in which the system's goals are drawn from a limited set of easily interrupted goals whose supporting tasks can be learned offline (Talamadupula et al., 2011). In contrast, tightly coordinated teams require team members to behave in a predictable manner so that their teammates can respond appropriately. In this context, each individual's behaviors for achieving goals should be guaranteed; hence, such systems can benefit from correct-by-construction controller synthesis (per team member). In this case, goal interruption must occur safely, which requires extra time to make sure that each team member can safely interrupt its current goal and start another. This delay decreases the reactivity of the goal autonomy layer.

The granularity of atomic actions available to the GDA Controller can vary from simple (e.g., "go to *x, y, z*") to complex (e.g., "supply landing sites for the UAVs and recharge their batteries"). This granularity depends on properties of the underlying control layers, which in turn depend on environment predictability and team coordination required. We present examples at opposite extremes of these domain properties, and note how these impact the granularity of the goals used.

5.1 Autonomous Behavior Technology for UUVs

While there is a large body of work on UUV motion autonomy, current approaches do not have the ability to reason about goals. In our planned approach, GDA will allow a UUV to respond with appropriate actions to unexpected situations whenever the vehicle's current set of goals is no longer satisfactory.

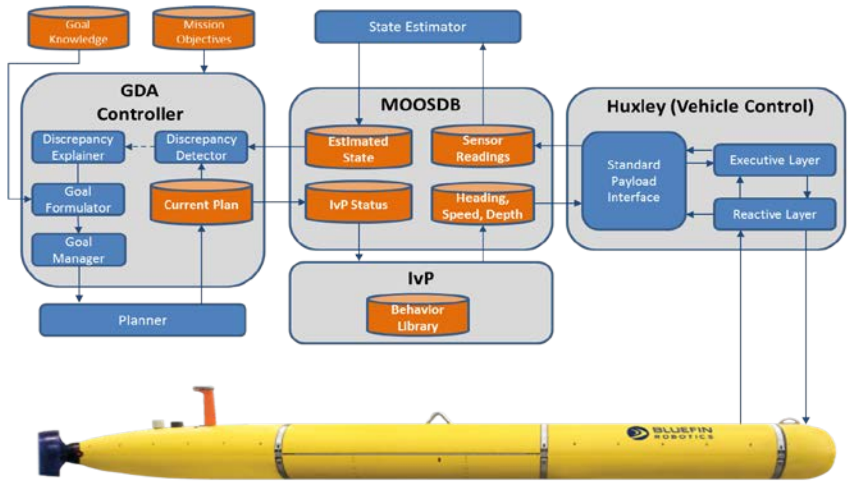

Figure 2: The GDA agent architecture for controlling a UUV with MOOS-IvP.

5.1.1 Integration with Motion Autonomy Systems

Deliberative motion autonomy techniques for UUVs require advance knowledge of the environment in which the path will be planned, any currents that must be taken into account, and the future motion of dynamic obstacles. In a long-duration mission, a UUV may venture into waters that are not well-charted or for which there are no reliable data on currents. Dynamic obstacles may include other vessels that are engaged in unpredictable maneuvering, or whose motion is not well-understood at the time of planning because sensor data are not conclusive. Without such useful constraints on the guidance problem, deliberative path planning alone may not be appropriate for a UUV on a long-duration mission.

We will apply the MOOS-IvP autonomy architecture (Benjamin et al., 2010) to provide suitable path guidance. MOOS is a message-passing middleware system with a centralized publish-subscribe model. IvP Helm is a behavior-based MOOS application that chooses a desired heading, speed, and depth for the vehicle in a reactive manner to generate collision-free trajectories. Unlike potential field methods, IvP Helm uses an interval programming technique that optimizes over an arbitrary number of objective functions to generate desired heading, speed, and depth values and activate or deactivate sensor payloads.

We developed a new GDA agent architecture based on ARTUE (Molineaux et al., 2010a), are using it to control a UUV in simulation, and will later apply it to control our UUV. The GDA Controller will direct the vehicle to perform various tasks (e.g., sensing, navigation) while preserving its ability to navigate partially unknown or poorly mapped environments. It will accomplish this by activating and deactivating specified IvP Helm behaviors and altering the parameters of active behaviors. While IvP Helm can make these decisions independently, it is a reactive mechanism and cannot deliberate about what goal the vehicle should pursue, which is the focus of GDA. Figure 2 depicts our agent architecture, where GDA will direct goal autonomy, IvP Helm will provide motion guidance, and Bluefin's Huxley control architecture will execute low-level control.

The UUV domain has few constraints on the environment, which distinguishes it from the contaminant detection domain, where we will use a constrained environment and abstractions to provide guarantees of motion controller correctness. The ocean is large, sparsely mapped, and dynamic. Therefore, it is not possible to provide guaranteed-correct motion control (Kress-Gazit et al., 2009). Furthermore, unlike the controllers we use on the UAVs, IvP Helm cannot independently recognize that a navigational failure has taken place.

To allow IvP Helm independent control over motion while preserving the GDA Controller's ability to recognize anomalous situations, we are developing an abstraction that replaces expected states in our Discrepancy Detector with semantically richer expectations. This will allow our agent to ignore certain values or expect values in some range between actions, and to resolve intervals between actions by checking conditions during execution rather than computing the expected duration of a process from a domain model. This would allow the goal reasoner to, for example, expect position values to fall within some range until a motion is completed or some other unexpected event (e.g., a barrier) triggers a discrepancy. Using this technique affords better separation of responsibilities between the goal autonomy layer and the motion autonomy layer. It also offers improved performance by eliminating discrepancies caused by allowing the motion autonomy layer to independently execute motion tasks and by obviating precise modeling of vehicle motion and other lower-level processes during planning.

5.1.2 Modeling Uncertainty

Our current model of discrepancies assumes that observations are not noisy. This assumption does not hold in real-world environments, where sensors are noisy and sometimes faulty, which can cause uncertainty in observations and the estimated state. The discrepancy model also assumes that observations occur at precise times relative to actions taken (i.e., either immediately after one action or immediately after the amount of time necessary for an event to occur as predicted by the domain model). This second assumption is also unrealistic: the sampling rate of the sensors may not correspond precisely to the timeline of the expected states, and the transmission and reception of the data by asynchronous processes that lack maximum-update-time guarantees may interfere with the timely delivery of the state observation. Hence, when detecting discrepancies, observations may not correspond exactly to expected states as generated by a planner, though they may be closely correlated.

To address these issues, we intend to improve our new expectations model by introducing a probabilistic model that assigns a distribution to each value or range in an expectation. This will allow for computing a likelihood value for each observed state, which can be used to detect discrepancies (i.e., under some conditions, a low likelihood for an observation may indicate a high probability that it is anomalous).

5.2 Autonomous Systems Integration

In this project, we will apply GDA to the problem of controlling a team of UAVs and UGVs to locate the source for a plume of airborne particles. While the maneuvering of sensors for plume source location has been previously studied (Spears, Thayer, & Zarzhitsky, 2009), little work has been done on providing autonomous support for such a team. We will apply goal autonomy to simultaneously coordinate search operations and logistics support, including safe landing zones and recharging stations.

5.2.1 Integration with Motion Autonomy Systems

We use a hierarchical approach for implementing team motion autonomy that involves three decision layers. The highest layer uses GDA to select mission goals. The GDA Controller uses a $\text{SHOP2}_{\text{PDDL+}}$ planner (Molineaux, Klenk, & Aha, 2010b) to produce a sequence of actions and associated safety conditions. The bounded nature of the UAVs' flight envelope guarantees that this planner will generate achievable plans, which are executed by an FSA on each vehicle to allow local trajectory planning, execution, and discrepancy detection. To increase robustness to agent failure and reduce the size of the FSA, we are employing the Physicomimetics swarm control algorithm (Apker & Potter, 2012) to reactively generate vehicle trajectories.

To bridge between high-level goals and low-level tasks in the GDA Controller, we will use LTL as a translation mechanism between decision layers. LTL controller synthesis has been used to automatically produce verifiable FSA controllers to accomplish complex tasks on autonomous robots (Kress-Gazit et al., 2009). In this approach, the GDA Controller will generate a set of complex actions and constraints for each agent's motion autonomy system, and the LTL Controller will generate simpler actions (e.g., "go to $\langle x, y, z\rangle$") for the agent's guidance system. This contrasts with previous approaches, which required LTL tasks to be pre-specified, or required pre-specified templates that can assign newly-discovered areas of interest as new destination goals (Sarid et al., 2012).

For a group of collaborating robots, the LTL controller synthesis problem quickly becomes infeasible. We are addressing this by using goal autonomy to alleviate this state-space explosion problem by supplementing the mission goal with smaller, short term goals with mission constraints. That is, we will use it to decompose the complete task specification into smaller, local specifications for individual or small teams of UxVs, thus limiting the goals that are within the scope of the task. This could reduce an infeasible task into smaller, more computationally efficient tasks for the LTL synthesizer.

The FSA that LTL synthesis creates can be used by the GDA Controller to detect unexpected events during operation. Discrepancies can be detected by comparing the FSA's expected state with the agent's observed state.

Finally, the FSA is guaranteed to satisfy its underlying task specification, which provides a valuable check to ensure that the goals selected by the GDA Controller do not conflict with each other or with the mission's safety constraints. This guarantee on the FSA's behavior assumes that the environment acts as expected, and that the robot's sensors and actuators operate without error. We can relax these assumptions by using Johnson and Kress-Gazit's (2012; 2013) method for analyzing the behavior of an LTL-synthesized controller, which tolerates errors in the sensing and actuation of the robot. After creating a probabilistic model of the robot's interaction with the environment, their method uses model checking to find the probability that the robot exhibits a particular behavior (defined by an LTL formula). This will be used by the Discrepancy Explainer to diagnose the perceived discrepancy.

5.2.2 Controlling a Team of Vehicles

In the contaminant detection domain, several UAVs and UGVs must coordinate to locate the contaminant's source. While the vehicles are expected to execute maneuvers independently, their efforts should be centrally coordinated to complete the mission quickly and with minimal mutual interference. Therefore, the GDA Controller must coordinate the vehicles' efforts. Our strategy

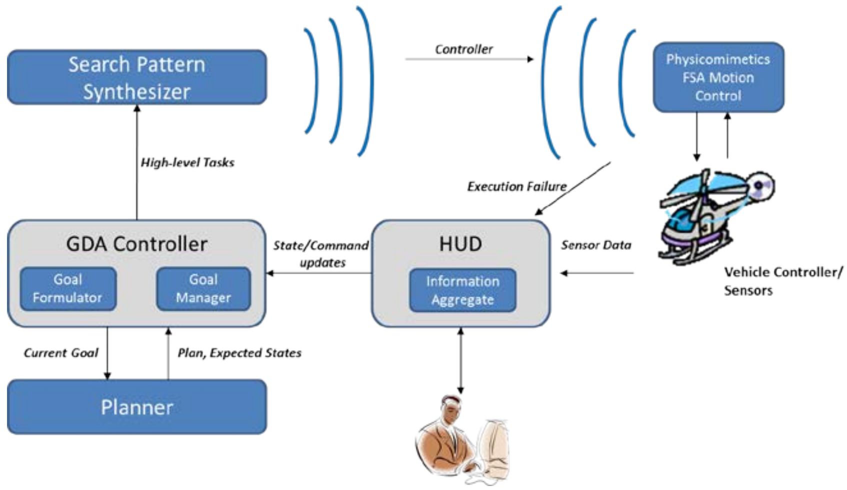

Figure 3. The GDA architecture for controlling the UAVs.

for solving this problem assigns the UAVs to follow plumes of contaminants to their source and uses UGVs in a support role.

Figure 3 depicts our prototype architecture, which uses the MASON simulation toolkit (Luke, 2005) to simulate vehicle motion and chemical-plume dynamics. The mission goal is to detect possible plume locations. Initially, the planner assigns all UAVs to small groups and directs each group to investigate a possible plume, or remain in reserve. Each group's plume assignment is passed to a separate intermediate level planner, which creates a lawnmower search pattern to follow. (In our future work, we will replace this with LTL-synthesized controllers.) All of the UAVs use Physicomimetics motion planning to jointly investigate each location in the pattern for evidence of a plume.

The discrepancies that we currently model concern unexpectedly low UAV battery states, suspected plume locations, and task completion signals from groups or individual agents. When a discrepancy is encountered, the GDA Controller reassesses its goals and forms new plans. For instance, if an agent's battery charge becomes crucially low, then the GDA Controller will assign a new goal for the agent to recharge its battery, and will change the group's composition by tasking other vehicles to continue searching for plumes. Later, we will model anomalies such as opportunities to deploy solar panels, which may interfere with UAV transport or landing operations, and winds that interfere with UAV flight and aerosol sensor performance.

We will integrate UGVs in this domain. They will transport UAVs to contaminated regions, harvest energy for battery power, and recharge the UAVs' batteries during operations. Launch, landing, search patterns, and battery charging involve precise, coordinated motion control that can be achieved only in favorable conditions. This requires guarantees on the agents' behavior throughout a maneuver, which is an ideal application of LTL control. The GDA Controller complements this by managing higher level goals, scheduling these operations, and determining their locations.

6. Discussion

We based our implementation decisions on the degree of predictability in each environment and the need for agent cooperation. These vary substantially between our two projects.

6.1 Environment predictability

Ocean currents, ship traffic, and underwater features are generally unknown in advance of deployment. As a result, any motion autonomy algorithm that makes specific guarantees is bound to fail in the UUV domain. There is little benefit in the UUV domain to synthesizing a guidance system more complex than a MOOS-IvP behavior, as the GDA Controller may frequently select new goals when more accurate states become available.

In contrast, the plume detection environment can be observed and accurately predicted over short time scales, allowing synthesis of controllers that are guaranteed to perform well in those conditions. At longer time scales, much of the environment is static or repetitive (e.g., areas of sun vs. shade), allowing a planner to schedule complex tasks with a high probability of success. The GDA Controller will detect fewer discrepancies in this environment and will be more focused on managing the team's resources.

The plume detection mission benefits from abstractions of the environment and agent behavior that are possible in predictable environments. These abstractions allow goal autonomy to largely ignore issues of motion autonomy.

6.2 Need for cooperation

The UUV domain involves a single vehicle that has little or no interaction with other agents, and reasons about only a few constraints (e.g., to avoid goal oscillations). This frees GDA to make highly independent decisions about the vehicle's activity by selecting the best available goal for its current state. This level of independence permits a direct connection between GDA and the guidance systems, with no need for a controller-synthesis step.

Cooperation is the defining feature of the plume detection domain. As a result, no individual agent can be allowed to replan its actions in a way that interferes with its peers. This forces goal autonomy to a central node whose role is restricted to issuing clearly defined instructions that will be used to synthesize low-level controllers (FSAs) for each team member. These extra layers of abstraction will allow goal autonomy to coordinate the team's behaviors to ensure that no hardware will be lost unexpectedly, although it will introduce delays between selecting and implementing new goals.

Architecture decisions involving cooperative agents need to balance closeness of cooperation with the agents' ability to respond to new information quickly. A continuum of cooperation options exists, varying from agents that cluster closely (to form coherent arrays) to fully independent agents. With less cooperation, fewer abstractions are required between GDA and low-level control, while close cooperation requires more abstractions and, implicitly, a more predictable environment to allow those abstractions.

7. Conclusion

In this paper we described initial architectures and proposed models for projects in which goal autonomy (i.e., the GDA model) will be used to control unmanned vehicles. We identified different modeling requirements in the application of GDA to situated agents depending on certain domain properties, which affect the capabilities afforded to GDA by lower level layers in the autonomy architecture. In particular, the granularity of actions that are atomic for the GDA Controller varies widely according to the computational complexity of motion and the guarantees provided by lower level systems.

In the contaminant detection domain, the motion of a team of vehicles toward a location where sensing will take place must be carefully coordinated so as to avoid collisions or other interference. Solving this guidance problem (i.e., finding waypoints that each individual should follow) in the goal autonomy layer would be computationally infeasible. However, specialized guidance techniques combined with domain-specific controllers, can reduce computational complexity. Hence, in the contaminant detection domain, the abstraction level of the GDA Controller's actions must be at least as high as instructions for each team of vehicles to follow.

In contrast, we do not require coordination of many individual agents in the UUV domain. Therefore, the GDA Controller's plans can be more concrete (e.g., specify a sequence of waypoints for the vehicle to follow). Furthermore, the unpredictability of the ocean environment requires that GDA detect discrepancies without the aid of guarantees as provided by the LTL controllers in the contaminant detection domain. To support GDA discrepancy detection, behaviors implemented by lower level systems should be as predictable as possible. This reinforces our belief that the GDA Controller's actions should be simpler in this domain.

Thus, when designing a goal autonomy robotic controller, the required granularity of the actions will be dictated by the available reactive and abstraction layers. Highly granular actions improve predictability but impose a higher computational burden on the GDA Controller. More abstract actions reduce this computational burden, but generally require more time to safely coordinate goal changes, reducing system reactivity. They also require more predictable environments for low level controllers.

As we progress to more complex tasks and control of non-simulated vehicles, we will develop and implement new models for GDA that address the issues of real-world situated agents. We have argued these models are needed (e.g., probabilistic expectation models for discrepancy detection). We expect to create compelling demonstrations of goal autonomy for controlling unmanned robotic vehicles after these models are in place.

Acknowledgements

Thanks to NRL for supporting this research. Thanks also to NRL's Brian Houston and Ben Dzikowicz for their insightful comments and corrections. The views and opinions contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of NRL or the DoD.

References

Alvarez, A., Caiti, A., & Onken, R. (2004). Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering, 29*(2), 418-429.

Antonelli, G., Chiaverini, S., Finotello, R., & Schiavon, R. (2001). Real-time path planning and obstacle avoidance for RAIS: an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering, 26*(2), 216-227.

Apker, T., & Potter, M. (2012). Physicomimetic motion control of physically constrained agents. In W. Spears & D. Spears (Eds.), *Physicomimetics: Physics-based swarm intelligence*. New York: Springer.

Benjamin, M., Schmidt, H., Newman, P., & Leonard, J. (2010). Nested autonomy for unmanned marine vehicles with MOOS-IvP. *Journal of Field Robotics, 27*(6), 834-875.

Binney, J., Krause, A., & Sukhatme, G. S. (2010). Informative path planning for an autonomous underwater vehicle. *Proceedings of the 2010 IEEE International Conference on Robotics and Automation* (pp. 4791-4796). Anchorage, AK: IEEE Press.

Choi, D. (2011). Reactive goal management in a cognitive architecture. *Cognitive Systems Research, 12*(3), 293-308.

Coddington, A. (2006). Motivations for MADBot: A motivated and goal directed robot. *Proceedings of the Twenty-Fifth Workshop of the UK Planning and Scheduling Special Interest Group* (pp. 39-46). Nottingham, UK: University of Nottingham.

Garau, B., Alvarez, A., & Oliver, G. (2005). Path planning of autonomous underwater vehicles in current fields with complex spatial variability: An A* approach. *Proceedings of the International Conference on Robotics and Automation* (pp. 194-198). Barcelona, Spain: IEEE Press.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Francisco, CA: Morgan Kaufmann.

Hawes, N., Hanheide, M., Hargreaves, J., Page, B., Zender, H., & Jensfelt, P. (2011). Home alone: Autonomous extension and correction of spatial representations. *Proceedings of the International Conference on Robotics and Automation* (pp. 3907-3914). Shanghai, China: IEEE Press.

Ho, C., Mora, A., & Saripalli, S. (2012). An evaluation of sampling path strategies for an autonomous underwater vehicle. *Proceedings of the International Conference on Robotics and Automation* (pp. 5328-5333). St. Paul, MN: IEEE Press.

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

Johnson, B., & Kress-Gazit, H. (2012). Probabilistic guarantees for high-level robot behavior in the presence of sensor error. *Autonomous Robots*, *33*(3), 309-321.

Johnson, B., & Kress-Gazit, H. (2013). Analyzing and revising high-level robot behaviors under actuator uncertainty. To appear in *Proceedings of the International Conference on Intelligent Robots and Systems*. Tokyo, Japan: IEEE Press.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings of the International Conference on Robotics and Automation* (pp. 500-505). St. Louis, MO: IEEE Press.

Kress-Gazit, H., Fainekos, G.E., & Pappas, G.J. (2009). Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, *25*(6), 1370-1381.

LePage, K.D., & Schmidt, H. (2002). Bistatic synthetic aperture imaging of proud and buried targets from an AUV. *Journal of Ocean Engineering, 27*(3), 471-483.

Livingston, S.C., Murray, R.M., & Burdick, J.W. (2012). Backtracking temporal logic synthesis for uncertain environments. *Proceedings of the International Conference on Robotics and Automation* (pp. 5163-5170). St Paul, MN: IEEE Press.

Mather, T., & Hsieh, M. (2012). Ensemble synthesis of distributed control and communication strategies. *International Conference on Robotics and Automation* (pp. 4248 –4253). St. Paul, MN: IEEE Press.

Molineaux, M., Klenk, M., & Aha, D.W. (2010a). Goal-driven autonomy in a Navy strategy simulator. In *Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Molineaux, M., Klenk, M., & Aha, D.W. (2010b). Planning in dynamic environments: extending HTNs with nonlinear continuous effects. In *Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Nau, D. (2007). Current trends in automated planning. *AI Magazine*, *28*(4), 43–58.

Nau, D., Cao, Y., Lotem, A., & Muñoz-Avila, H. (1999). SHOP: Simple hierarchical ordered planner. *Proceedings of the Sixteenth International Joint Conferences on Artificial Intelligence* (pp. 968-973). Stockholm, Sweden: Morgan Kaufmann.

Plaku, E., & McMahon, J. (2013). Motion planning and decision making for underwater vehicles operating in constrained environments in the littoral. To appear in *Towards Autonomous Robotic Systems.* Oxford, UK: Springer.

Powell, J., Molineaux, M., & Aha, D.W. (2011). Active and interactive learning for goal selection knowledge. In *Proceedings of the Twenty-Fourth Florida Artificial Intelligence Research Society Conference*. West Palm Beach, FL: AAAI Press.

Sarid, S., Xu, B., & Kress-Gazit, H. (2012). Guaranteeing high-level behaviors while exploring partially known maps. In *Proceedings of Robotics: Science and Systems*. Sydney, Australia: MIT Press.

Schermerhorn, P., Benton, J., Scheutz, M., Talamadupula, K., & Kambhampati, S. (2009). Finding and exploiting goal opportunities in real-time during plan execution. *Proceedings of the International Conference on Intelligent Robots and Systems* (pp. 3912-3917). St. Louis, MO: IEEE Press.

Spears, D., Thayer, D. and Zarzhitsky, D. (2009). Foundations of swarm robotic chemical plume tracing from a fluid dynamics perspective. *International Journal of Intelligent Computing and Cybernetics*, *2*(4), 745–785

Talamadupula, K., Kambhampati, S., Schermerhorn, P., Benton, J., & Scheutz, M. (2011). Planning for human-robot teaming. In G. Cortellessa, M. Do, R. Rasconi, & N. Yorke-Smith

(Eds.) *Scheduling and Planning Applications: Papers from the ICAPS Workshop*. Freiburg, Germany: AAAI Press.

Tan, C.S., Sutton, R., & Chudley, J. (2004). An incremental stochastic motion planning technique for autonomous underwater vehicles. *Proceedings of IFAC Control Applications in Marine Systems Conference* (pp. 483-488). Ancona, Italy: Elsevier.

Warren, C.W. (1990). A technique for autonomous underwater vehicle route planning. *IEEE Journal of Oceanic Engineering, 15*(3), 199-204.

Weber, B. G., Mateas, M., & Jhala, A. (2012). Learning from demonstration for goal-driven autonomy. In *Proceedings of the Twenty-sixth AAAI Conference on Artificial Intelligence*. Toronto, Ontario, Canada: AAAI Press.

Wilson, M., Molineaux, M., & Aha, D.W. (2013). Domain-independent heuristics for goal formulation. In *Proceedings of the Twenty-sixth International FLAIRS Conference.* St. Pete Beach, FL: AAAI Press.

Wongpiromsarn, T., Topcu, U., & Murray, R. M. (2012). Receding horizon temporal logic planning. *IEEE Transactions on Automatic Control*, *57*(11), 2817-2830.

Wooden, D., Malchano, M., Blankespoor, K., Howard, A., Rizzi, A., & Raibert, M. (2010). Autonomous navigation for BigDog. *Proceedings of the International Conference on Robotics and Automation* (pp. 4736-4741). Anchorage, AK: IEEE Press.

Worcester, J., Rogoff, J., & Hsiehm M. (2011). Constrained task partitioning for distributed assembly. *Proceedings of the International Conference on Intelligent Robots and Systems* (pp. 4790–4796). Vilamoura, Algarve, Portugal: IEEE Press.