Proceedings of the 18th International Symposium on Unmanned Untethered Submersible Technology (UUST),
Portsmouth New Hampshire, August 2013.

RUN-TIME ETHICS CHECKING FOR AUTONOMOUS UNMANNED VEHICLES: DEVELOPING A PRACTICAL APPROACH

Donald P. Brutzman, Duane T. Davis, George R. Lucas Jr., and Robert B. McGhee
Naval Postgraduate School (NPS)
Monterey California USA 93943
brutzman@nps.edu, dtdavi1@nps.edu, grlucas@nps.edu, robertbmcghee@gmail.com

Abstract

Recent experience has shown that the concept of robot ethics is important for establishing norms defining allowed behaviors for unmanned systems. However, approaches considered to date are either based on highly abstract artificial intelligence schemes or else uniquely "hard wired" into a given robotic architecture in an unrepeatable fashion. A more-general approach is needed for defining and deploying ethical constraints on robotic systems that are expected to operate autonomously with only occasional human control.

This paper explores development of a practical approach to ethical operation of unmanned maritime systems in maritime environments. This approach is based on the Rational Behavior Model (RBM), a three-layer robot control architecture modeled on the control hierarchy of Naval vessels. RBM utilizes a finite state machine as the basis for mission definition and high-level control. This approach provides for exhaustive pre-mission testing by human operators and predictable runtime decision-making that takes mission-specific ethical constraints into account.

The authors believe that these achievable approaches for ethically constrained mission planning and operations can be applied to a broad range of unmanned systems. The potential legal and ethical impacts of unmanned systems in the real world must be considered and addressed. Although our focus is on the most challenging circumstances of autonomous lethality in naval scenarios, the same approaches also appear to be relevant to scientific, commercial and civilian operations by unmanned maritime systems. This work opens a practical door leading into that necessary future.

Approach Overview

Ethical operation of unmanned systems is becoming a topic of significant importance not only for operators, but for decision-makers, developers, and policy-makers as well. At the same time, increasing onboard computational capabilities have enabled these systems to operate with more autonomy and to conduct increasingly complex missions with little or no human intervention. A number of approaches to the implementation of ethics for unmanned systems have been suggested [1]. Many of these are premised on the availability of robust artificial intelligence (AI) algorithms.

Legal culpability requires predictable, consistent, and repeatable robot activities that are conducted coherently with human operators. Decades of implementation experience have led us to believe that human-like intelligence and judgment are not required to achieve a useful operational capability in autonomous mobile robots. Furthermore, we are convinced that fundamental and useful types of robot ethical behavior can also be attained, even in hazardous or military environments, without invoking concepts of AI whose consistent implementation may prove speculative, futuristic, and operationally ambiguous.

Current research work at NPS has demonstrated practical approaches for ethical reasoning to govern robot behaviors. These consist of applying ethical constraints to existing patterns of robot planning in a manner that matches real-world operational constraints, rather than creating some new paradigm for philosophical contemplation. This approach is intentionally grounded in patterns of command decision-making used by military forces, which often must operate in a loosely coordinated fashion while observing highly consistent rules of engagement (ROE). Our focus is on maritime robotics supporting fleet and coalition naval forces. The same ethical patterns also appear to be applicable to scientific and commercial use of unmanned maritime systems.

Three key insights inform this work. First is that humans working in military units are able to deal with these challenges without ethical quandaries; unmanned systems need to compatibly follow their operations orders and ROEs under loose supervision, just as any other military unit might. Second is that abstract ethical behaviors don't define the mission plan; rather ethical constraints inform the mission plan. Ethical tasking is a modification of regular tasking, not an entirely new philosophical paradigm. Third is that the most-promising path forward lies in producing general mission orders that are first

understandable by (legally culpable) humans, then reliably and safely executable by robots.

Recent project results include defining, evaluating and visualizing tactical missions for unmanned systems using open-source NPS simulation software, the Autonomous and Unmanned Vehicle Workbench (AUVW). It is also feasible to meaningfully rehearse robot operations using simulation prior to field testing. An important aspect of our work is that mission orders can be tested exhaustively in human executable form before being translated into robot executable form. This provides the kind of transparency and accountability needed for after-action review of missions, and possible legal proceedings in case of loss of life or property resulting from errors in performing mission orders.

This paper describes a practical approach to the implementation of ethical behavior for unmanned underwater vehicles (UUVs) and other maritime vehicles which operate autonomously while following human-specified mission guidance. Specific topics include a discussion of operational ethics as they relate to unmanned vehicles, the Rational Behavior Model (RBM) unmanned system control architecture, and extensions to RBM that incorporate ethical constraints. Examples are presented along with a comparison to other proposed approaches to ethical unmanned system operation.

Ethics and Unmanned Systems in Maritime and Underwater Environments

The ethical and legal concerns attendant upon the development of unmanned underwater and surface vehicles (UUVs, USVs) are, at first glance, continuous with those already encountered in the aerial environment (which, owing to the use of Predators and Reapers in "targeted killing" operations, has garnered the most attention). In all environments, the concerns focus on the prospects for operational compliance with relevant international law (especially the Law of Armed Conflict, LOAC), and the dilemmas of moral and legal accountability for the consequences of noncompliance (for whatever reason) with these prevailing norms.

In the so-called "just war tradition," as well as in international law, the most important and relevant norms or principles guiding the conduct of armed conflict are usually termed "discrimination" (or distinction) and "proportionality." The former specifies that non-combatants ("civilians, and civilian objects") are never legitimate targets of attack. This is also known as the "Principle of Noncombatant Immunity."

Proportionality, by contrast, imposes an economy of scale on the use of military force in tactical operations that are otherwise deemed to be required by the overall military and political strategy being pursued through armed conflict. Any tactical military operation that is deemed to be required to attain a legitimate military objective, and which is legally and morally permissible (inasmuch as it does not aim at achieving the military objective through the deliberate targeting of the lives and property of noncombatants), is further constrained under international law to employ *only as much lethal force as is required* to obtain the legitimate and permissible objective. This second principle is sometimes termed colloquially "the economy of force," and demands that military personnel use only as much lethal force as is necessary to mission success, and not engage in reckless or wanton, unnecessary destruction or loss of life. In making that determination, moreover, the damage and loss of life caused by the operation (including inadvertent and unintended harm that might befall noncombatants) must be deemed *reasonable* in light of the importance of the objective attained. These vaguely-defined and imprecisely quantifiable concerns, taken together, comprise the Principle of Proportionality in combat.

The increased military use of robotics in armed conflict has been driven as much by these legal and moral considerations as by economics. Like precision-guided munitions that preceded them, most military uses of robotics are thought to be, or (as some advocates claim) have proven to be simultaneously more discriminate in avoiding direct targeting of noncombatants, and vastly more proportional in the destruction and harm inflicted on adversaries generally, than their conventional counterparts. In principle (and setting aside the general problems associated with precise identity recognition and targeting in unmanned systems), any UAV or UUV can be programmed straightforwardly to never target civilians ("noncombatants") or civilian objects. The precision with which they can otherwise direct uses of lethal force largely guarantees that their use in a given military operation will result in greater compliance with the demands of the Principle of Proportionality than will any conventional alternative. Those who defend, or advocate for the increased use of unmanned systems in armed conflict [1,2,3] do so largely in light of these considerations: namely, that responsible use promises even greater compliance with the ethical and legal norms that govern armed conflict and the use of force generally.

Critics of military robotics have, of course, challenged these findings and objected to these

promises as rosy and misleading [4-6]. They note, with some justification, that in concrete examples (such as the highly controversial use of unmanned or unoccupied aerial platforms in "targeted killings" of suspected terrorists), the technology makes feasible missions that would not likely otherwise be undertaken, and thus invariably inflicts casualties on noncombatants and their property that would otherwise not have occurred [2]. Were such systems enabled with increased autonomy, moreover, these critics [7] object that no meaningful accountability would attach under law or morality for such harm, whether inadvertent or somehow "deliberate" (in the sense of machine malfunction, or of mistaken character recognition in the targeting determination). In the case of full autonomy linked with lethal force, and absent any human oversight, neither the field commander, initial operator, nor manufacturer can be held criminally liable or otherwise morally accountable for the harm done, or the laws thereby violated, because the "machine itself" would have made and undertaken the relevant decisions and actions in question. Thus it seems (the critics conclude) absurd to hold a machine morally or legally accountable for its actions, which in turn leads to recommendations that robots with potentially lethal effects simply cannot be permitted.

One dramatic response [1] is to design an autonomous and lethally armed unmanned system that is simultaneously *morally responsible*, and so capable of making moral decisions, complying with international law, and being held meaningfully responsible for errors or infractions. This admittedly complicated, formidable, and as-yet unrealized engineering goal would employ strong conceptions of artificial intelligence, to include as-yet undeveloped computational models of moral reasoning, while equipping unmanned systems themselves with a robotic analogue of moral emotions like "guilt," which would aid the system in moderating and effectively controlling its use of force and its targeting decisions. An initiative to advance our current abilities in the area of artificial moral reasoning along such lines is currently being undertaken by interdisciplinary teams at Georgia Tech, and at Tufts and Brown Universities, for example.

Our approach to "runtime ethics" offers several practical advantages over these alternatives, as well as with respect to these substantive obstacles and criticisms. First, the underwater environment is relatively free of potential moral hazards when compared with land or air environments. Seabed infrastructure is usually well plotted and reported on nautical charts. Trawlers and dive operations are easily locatable. Apart from the occasional whale, marine sanctuary, drift net, or perhaps James Cameron diving in the Mariana Trench, the underwater environment is largely devoid of mobile civilian or commercial operations that might pose risk of inadvertent harm. Hence the scale of difficulties presented by the twin principles of noncombatant immunity and proportionality are greatly reduced in comparison with these other, more familiar operational environments for unmanned systems.

Secondly, our command-scenario approach to unmanned underwater systems does not entail, require, or aim at fully autonomous operation. Instead, the runtime environment is more correctly classified as "semi-autonomous" (in compliance with current OSD Guidance, 2012). UUVs within this environment are designed to operate autonomously under normal mission conditions, but also equipped to recognize when a maneuver or operation might encounter moral or legal restrictions (such as whether or not to follow an enemy submarine that undertakes evasive action by illegally entering a prohibited underwater marine preserve). In such instances, the system is programmed either to abort the mission and return to base, or else to "flag" the consequent decision for executive command oversight (very much as would a human submarine commander, engaged in such a mission and confronting such a moral or legal dilemma). This ensures either that mistakes and inadvertent violations of law are avoided, or that, when undertaken, ambiguous missions have been thoroughly vetted and approved by those capable of assuming command responsibility for the decisions. In this manner, we believe we have integrated ethical and legal concerns and constraints into the overall design and operation of our UUVs in a manner that is fully compliant, thoroughly responsive and responsible, and perhaps most importantly, feasible using existing technology, rather than relying on futuristic speculative technological advances.

The Rational Behavior Model (RBM)

Overview and History. The initial form of the tri-level RBM control architecture depicted graphically in Figure 1 was introduced in 1993 and extended over subsequent years to effectively link both high-level symbolic processing associated with mission planning and control to largely numerical computation required for real-time vehicle control and sensor processing [8, 9, 10]. In the vocabulary of the RBM formalism, the Execution Level is the lowest layer of vehicle control software and is concerned with carrying out the hard real-time tasks

typically associated with physical interaction of a vehicle with its surrounding medium. Operations at this level must execute in real time and are typically reactive in nature. In a manned submarine, these tasks are usually carried out by enlisted crewmembers and include responsibilities such as controlling diving planes, rudders, engine rpm, etc.

In the middle RBM layer, the Tactical Level, Execution Level functions are organized into behaviors. Such functions include maintaining course and depth, sonar obstacle avoidance, maneuvering while surfaced, etc. However, the Tactical Level also includes more complicated behaviors such as sonar mapping, waypoint navigation, mapping, search planning, response to emergency situations, etc. The requirements of this layer correlate to the officer watchstanders in a manned submarine. In this context, the Tactical Level itself can be viewed as analogous to the Officer of the Deck responsible for coordinating the actions of the other watch officers to carry out high-level mission tasks [8, 10].

In addition to implementing behaviors, the Tactical Level is responsible for maintaining much of the vehicle's state information in the form of a world model. In the context of ethical robot operation, this is an important responsibility because much of the robot's situational awareness upon which ethical decisions rely is maintained in this model. Thus, it will be at the Tactical Level that adherence to or violation of ethical constraints will first be noted. We have repeatedly found that modeling robot logic after common human patterns for complex action provides an excellent structural basis for achieving robot responses that are both logically consistent and operationally complementary to human activities.

The highest level of software in the RBM architecture is called the Strategic Level [8-10]. This level corresponds to the commanding officer of a manned submarine. Unlike the RBM Execution and Tactical Levels, the Strategic Level conducts only symbolic computation in a non-numeric mode. This level is responsible for considering alternative actions and making decisions without a sense of continuous time or space. Stated differently, the Strategic Level operates entirely in the domain of mathematical logic to determine which high-level mission goals to pursue based solely on the success or failure of previous high-level goals. Thus, Strategic Level reasoning is conducted without regard for vehicle state and is based solely on the Tactical Level's understanding of when the currently executing goal is successfully achieved, has failed to be achieved, or can no longer be pursued without violating ethical constraints [15].

The non-numerical nature of the Strategic Level allows for declarative mission definitions that are compatible with first-order logic. Early Strategic Level implementations utilized the logical programming language Prolog to define missions that were both human readable and machine executable. Recent implementations, including the AUVW mission simulation and rehearsal system, have relied on XML vocabularies that also support both human readability and machine processing [11-13, 15-19]. In addition, the ubiquity and standardization of XML-processing applications and programming interfaces makes XML vocabularies suitable for use on multiple vehicles and various computing platforms.

Complex missions to be carried out by human agents are often specified in terms of a series of phases with predetermined phase transition rules and defined mission end conditions. The RBM models this approach by representing missions as finite state machines (FSM) where individual Strategic Level goals correspond to states and transitions are executed upon the success of failure of the corresponding goal. Specifying missions in this way provides a level of determinism at the Strategic Level. That is, a specific sequence of Strategic Level goal successes and failures will result in a predictable goal execution sequence. Subsequent sections will demonstrate the use of this determinism to support exhaustive mission testing that directly supports the imposition of operator accountability.

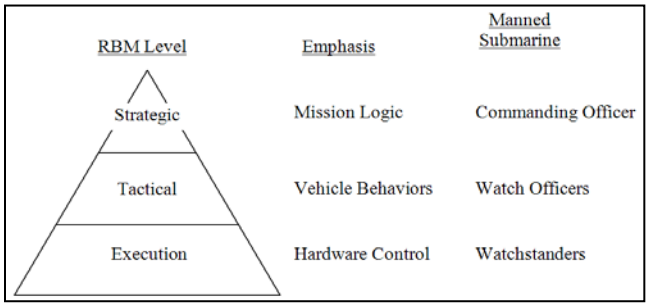

Figure 1. RBM three-level architecture compared to the control paradigm of a manned naval vessel. [8]

Design: Mission Execution Automata (MEA). One aspect of the original RBM that slowed progress was need for a mathematical model for Strategic Level vehicle control. Recent work addressed this issue through formal definition of RBM semantics in mathematical terms as a generalization of a Turing Machine (TM) into a broader class of automata, Mission Execution Automata (MEA) [12-13,15-18].

```
Goal 1.  Proceed to Area A and search the area.  If the search is successful execute goal 2.  If the
search is unsuccessful, execute goal 3.

Goal 2.  Obtain an environment sample from Area A.  If the sample is obtained, execute goal 3.  If
the sample cannot be obtained, proceed to recovery position to complete the mission.

Goal 3.  Proceed to Area B and search the area.  Upon search success or failure, execute goal 4.

Goal 4.  Proceed to Area C and rendezvous with UUV-2.  Upon rendezvous success or failure, proceed to
recovery position to complete the mission.
```

Figure 2. Example mission orders expressed in structured natural language using a standardized vocabulary. [13,15-18]

Thus general approaches for a wide repertoire of activities can remain computationally tractable and consistently implementable across a complete variety of computer architectures. Such generality is critical if common tasking paradigms are to be achieved for all unmanned systems.

Deeper theoretical investigation was also pursued to confirm tractability. A TM consists of an FSM augmented by an external agent in the form of a potentially infinite memory realized as the tape of an "incremental tape recorder". It is known that no digital machine can be more computationally powerful than a universal Turing machine, in which the logical behavior of a specific FSM is encoded on the tape of the machine in the form of a state table [14]. The crux of TM generalization to MEAs is allowing the external agent to be not only a tape recorder, but alternatively, either a human being or a sensor-based robot [11]. Specifically, an MEA consists of a mission-specific FSM provided with one or more external agents. Each of these agents must have the ability to sense the environment in some well-defined and limited way and to respond to commands issued by the FSM and answer queries from the FSM using a predetermined, finite vocabulary [13]. Although, TMs have been almost exclusively relegated to the status of a mathematical concept, their generalization to an MEA provides an attractive UUV control mechanism that allows TM semantics to transparently interact with an RBM Tactical Level implementation to provide a convenient, mathematically grounded basis for control of long-duration autonomous robot missions that allows for dealing with unforeseen runtime contingencies [13].

An individual MEA is (by definition) mission specific since it utilizes a mission-specific FSM. For general application it has proven useful to define a universal Mission Execution Engine (MEE) that is analogous to a universal TM [14] that can implement a FSM representing an arbitrary UUV mission [15,16]. In this context the MEE can be viewed as executing mission orders for specific UUV missions just as the command structure of a manned submarine executes specific tasking from higher authority. These mission orders are also analogous to, but more general than, the machine description part of the universal TM [14]. For UUVs operating in high-risk environments, it is the authors' contention that it is a moral imperative that human operators be legally accountable. The mathematical formalism provided by MEEs directly supports this level of accountability because its determinism allows mission orders to be subjected to exhaustive testing before actual robot execution. When such testing has been completed, the senior mission specialist can formally approve orders as an executable specification for the subsequent generation of robot mission orders by robot specialists. The mission specialist then can reasonably assume legal accountability for any errors in mission orders. Further, mission orders specified in an executable form such as appropriately structured Prolog or XML can be read declaratively by non-programmers (e.g., mission specialists) to provide the kind of transparency needed for after action review and possible legal proceedings relating to loss of life or property.

Example RBM Mission using Prolog. A simple five-phase unmanned vehicle reconnaissance mission might be phrased in specialized natural language as in Figure 2. This mission is used to illustrate the design of an MEA capable of carrying out any similar mission when expressed as a series of phases written as mission orders. It is assumed that the syntax and semantics of these specialized natural language mission orders are understood in the same way by both the person issuing the orders and the person receiving them. Orders written to achieve this objective are said to be syntactically well formed and semantically unambiguous.

Figure 3 presents a more abstract representation of the logic of the example mission in the form of a state graph that includes "Start", "Mission Abort" and "Mission Complete" terminal phases that are implicit in the natural language specification. Although not directly executable, this depiction provides an intuitive view of the mission-specific state machine. Evidently, this state graph is in agreement with the natural language mission orders of Figure 2.

Executable specifications of this mission have been developed for simulation and human testing in both Prolog and XML [13, 15]. Human testing of the mission flow was demonstrated in [13] as shown in Figure 3, exploring the mission of interest shown in Figure 4. The annotated user log depicts the results of three separate human mission tests of hypothetical mission progressions. These tests show Strategic Level invocations of various phases and queries concerning the status of the currently executing phase.

```
CG-USER(1): (?- execute_mission)) ; example 1
Search Area A!
Search successful? yes
Sample environment!
Sample obtained? yes
Search Area B!
Search successful? yes
Rendezvous UUV2!
Rendezvous successful? yes
Return to base!
At base? yes
Mission succeeded.


CG-USER(2): (?- execute_mission)) ; example 2
Search Area A!
Search successful? yes
Sample environment!
Sample obtained? no
Return to base!
At base? yes
Mission succeeded.


CG-USER(3): (?- execute_mission)) ; example 3
Search Area A!
Search successful? no
Search Area B!
Search successful? no
Rendezvous UUV2!
Rendezvous successful? yes
Return to base!
At base? no
Mission failed.
```

Figure 3. Excerpted, annotated test results: exhaustive interactive human testing of area search and sample mission orders. User responses are shown in bold face. [15]

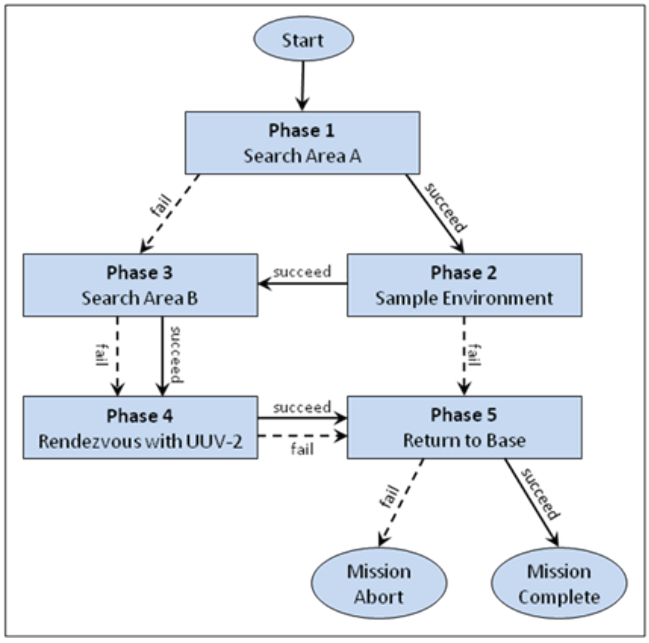

Figure 4. State Graph for example UUV area Search and Sample Mission, used in previous and current testing. [15]

For example, the first sequence tests the mission flow when the Tactical Level successfully completes all phases. In this case the order of execution is search area A, sample environment, search area B, rendezvous with UUV-2, return to base, and mission complete. The second and third scenarios test different sequences of phase success and failure. Because these tests are deterministic, the operator can definitively test mission flow for all possible sequences of phase success and failure. As long as the FSM is free of loops and of reasonable complexity, exhaustive testing is possible. Additionally, the FSM's mathematical rigor allows for automated structural testing of the mission for loops, state reachability, satisfiability, and acceptance of all possible phase success/failure sequences [14]. Mapping RBM to these fundamental constructs added confidence in the generality of this approach. Nevertheless, TMs and FSMs are not defined at an appropriate level of abstraction to handle real-time mission control in a practical fashion.

Human testability and automatic testability of mission logic are critical features that must be retained by any architectural design which hopes to inspire confidence in human supervisors that robot tasking is predictable, complete, and compliant with guiding regulatory doctrine.

In addition to mission-flow testing with a human operator providing Tactical Level responses, this mission has also been tested in simulation using the AUVW as documented in [11]. Because simulation testing within the AUVW incorporates fully implemented Tactical and Execution Level software operating in a physically-based virtual environment, these tests provide important insights into how the mission is likely to progress in real-world situations.

Autonomous Vehicle Command Language (AVCL)

Tactical orders. Over the past two decades, much work in this project has been dedicated to establishing a common declarative language that expresses task elements finding common employment in ships, submarines and aircraft. Typically such tasks and general orders are defined in a consistent manner for naval and commercial domains, as evidenced by the fact that human pilots and ship drivers can observe operations in the cockpit or control room of nearly any other vessel and immediately comprehend current activities. Example general orders include "all ahead full," "come right to course 090," "make your depth/altitude 100 meters," "all stop" etc. Low-level orders typically match open-loop control algorithms (e.g. "right full rudder") or closed-loop control algorithms (e.g. "steady course north"). In the RBM architecture, this level of command is handled at the Tactical level. The AVCL tactical command repertoire for USVs and UUVs are shown in Figures 5 and 6 respectively. Similar command sets have been derived for unmanned air vehicles (UAVs) and unmanned ground vehicles (UGVs).

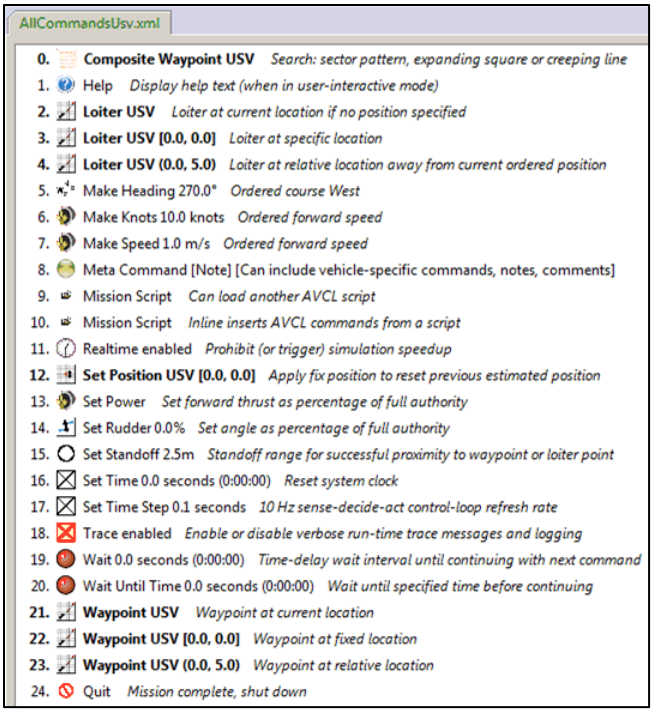

Figure 5. Task vocabulary of Autonomous Vehicle Command Language (AVCL) commands for USVs, corresponding to RBM Tactical and Execution levels.

Numerous missions have been simulated using these command sets. AVCL missions can also be easily translated into a variety of command dialects or source-code sequencers that are unique to a variety of types of robot architectures. Thus the current approach serves as an excellent basis for exploring mission-tasking issues for a wide range of robots.

Some of the more-advanced tactical orders ("Hover at point X-ray," "Rendezvous in Zone Charlie," "Surface and return to base") can match combinations of lower-level tactical orders. An important characteristic of such tasks is that they either succeed or fail. For tactical mission orders, more complicated mission orders are avoided. This matches real-world conditions for human operators where various tasks and activities are either pursued or not. Thus a set of tactical-level mission commands can be used as a single operational task block, running to completion in a linear fashion with no embedded decision-tree branching.

Many of the simpler AVCL tactical commands directly invoke Execution Level functionality for open-loop or closed-loop control of basic vehicle maneuvering. All tactical orders, which operate sequentially and require varying time intervals, depend upon hard-real-time operation of a sense-decide-act RBM Execution Level to handle fundamental control issues.

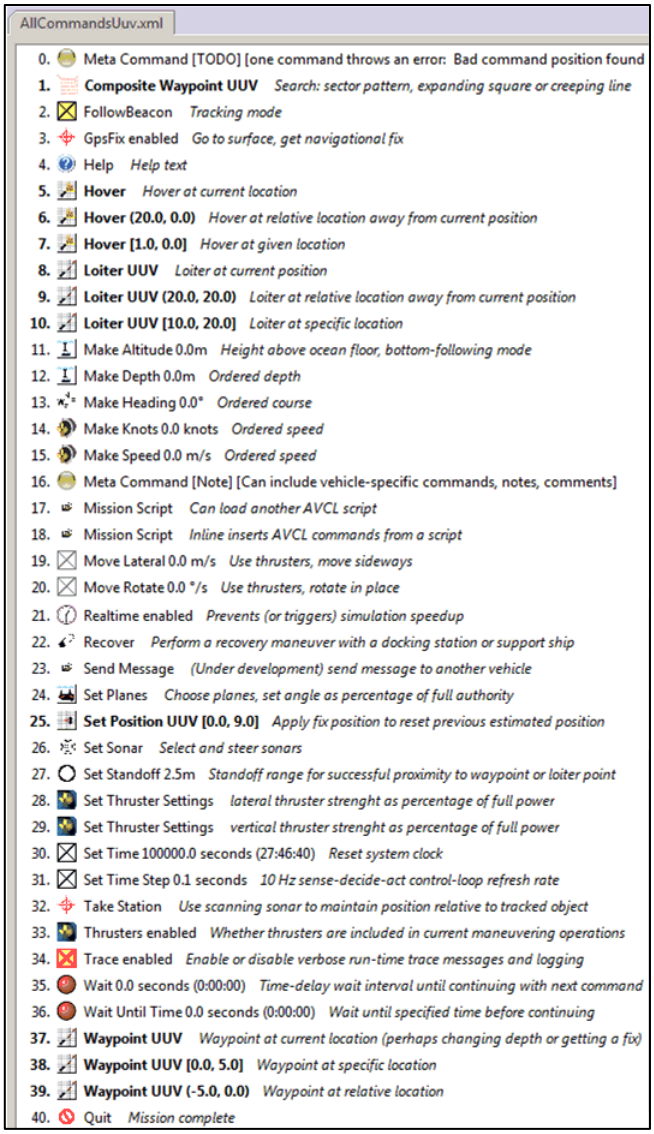

Figure 6. The vocabulary of AVCL task commands for UUVs is closely similar, adding depth information (or altitude over bottom) to the repertoire shown for AVCL USV commands [19].

Strategic orders. Mission logic, meaning decision-making choices and determination of courses of action, occurs at the Strategic level of the RBM architecture. A different set of logical constructs is used for mission control at this level of abstraction. Tactical-level tasks are considered, commanded and executed, as deemed appropriate, continuing through a series of task successes or failures, until a strategic-level mission is complete. This separation of concerns between strategic-level decision making and tactical-level task execution provides a general approach to mission conduct that can unambiguously map to a wide variety of high-level control paradigms used by human and robotic systems.

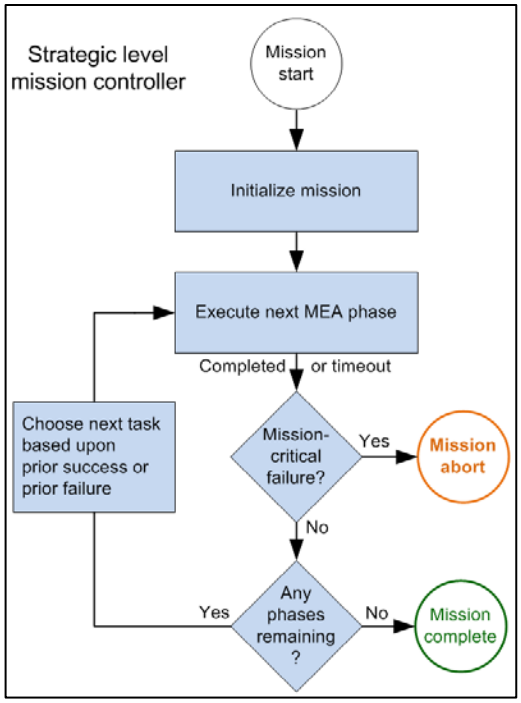

Figure 7. The strategic-level mission controller sequences through MEAs task phases until mission is complete. Note that success/failure is an essential design characteristic for each phase of mission conduct.

The following AVCL vocabulary of strategic-level mission MEAs is designed and implemented (to varying degrees) in the AUV Workbench for simulated mission testing.

- *Attack*
- *Decontaminate*
- *Demolish*
- *IlluminateArea*
- *Jam*
- *MarkTarget*
- *MonitorTransmissions*
- *Patrol*
- *Rendezvous*
- *Reposition*
- *SampleEnvironment*
- *Search*

Several example mission controllers follow in Figures 8 and 9 that illustrate the basic algorithmic logic for specific operational vehicle tasks (e.g. searching an area, etc.) used by the Strategic Level. Similar algorithms are provided for each of the strategic-level behaviors in the AVCL vocabulary. Each tactical task can require an arbitrary or maximum time period to complete. Each task type in the strategic-level mission vocabulary also runs to completion, reporting either failure or success when done. An excerpted screenshot of the AUV Workbench graphical user interface (GUI) that supports goal editing follows in Figure 10.

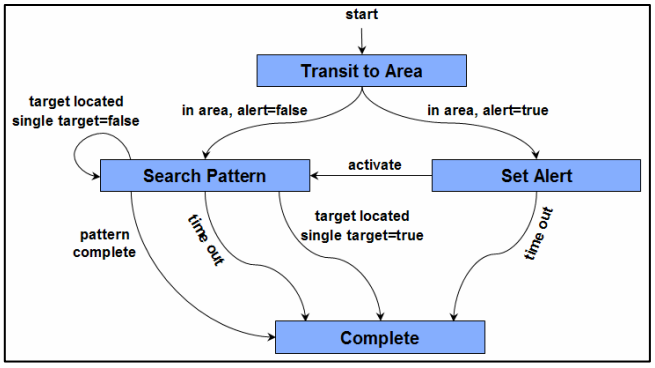

Figure 8. Strategic goal task algorithm for Search [17]. Completion status can only result in success or failure.

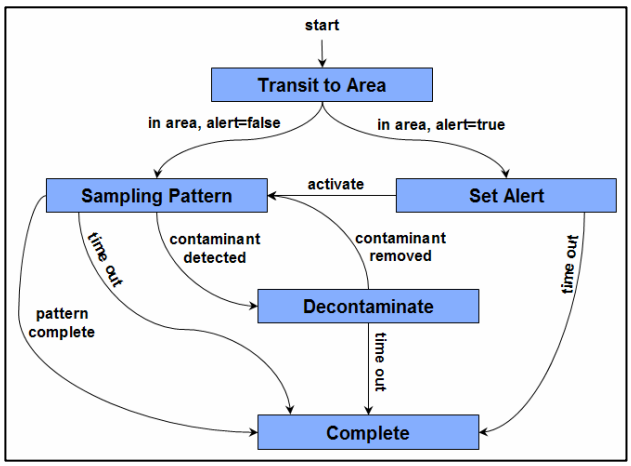

Figure 9. Strategic goal task algorithm for Decontaminate [17]. Completion status can only result in success or failure.

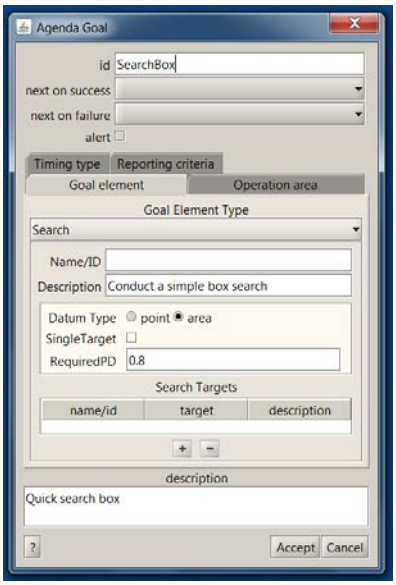

Figure 10. Example AUV Workbench GUI tabbed panel for editing an AVCL Search goal. [19]

Extensible Markup Language (XML)

Devising a readable format for robot-mission definition is important in order to achieve portability among multiple computer systems. AVCL is itself captured in XML, which not only provides plain-text readability but also full message validatability of both syntax and semantics. This feature is essential for mission reliability, helping to ensure the highest possible quality assurance.

Minor syntax or major semantic errors cannot be permitted to pass unchecked before deployment for at-sea execution. XML data inputs are portable and can be read and processed using libraries available for nearly any program language. Figure 11 shows example pseudo-code XML for the canonical mission. Of note is that such syntax is readable both by humans and systems. Further refinement produces AVCL-compliant mission specifications exactly matching the mission definitions found in Figures 2 and 4.

```
<?xml version="1.0" encoding="UTF-8"?>
<UUVMission>
  <GoalSet>
    <Goal area=”A” id=”goal1”>
      <Search nextOnSucceed=”goal2” nextOnFail=”goal3”/>
    </Goal>
    <Goal area=”A” id=”goal2”>
      <SampleEnvironment nextOnSucceed=”goal3” nextOnFail=”recover”/>
    </Goal>
    <Goal area=”B” id=”goal3”>
      <Search nextOnSucceed=”goal4” nextOnFail=”goal4”/>
    </Goal>
    <Goal area=”C” id=”goal4”>
      <Rendezvous nextOnSucceed=”recover” nextOnFail=”recover”/>
    </Goal>
    <Goal area=”recoveryPosition” id=”recover”>
      <Transit nextOnSucceed=”missionComplete” nextOnFail=”missionAbort”/>
    </Goal>
  </GoalSet>
</UUVMission>
```

```
<MissionPreparation>
  <UnitsOfMeasure time="seconds" mass="kilograms" angle="degrees" distance="mete
  <GeoOrigin longitude="-121.88500213623047" latitude="36.606998443603516" />
  <AgendaMission>
    <LaunchPositionAH description="Start point">
      <XYPosition timeStamp="0.0" y="6350.0" x="12300.0" />
    </LaunchPositionAH>
    <RecoveryPosition description="Finish point">
      <XYPosition y="6500.0" x="12300.0" />
    </RecoveryPosition>
    <GoalList>
      <Goals description="search operating area A" alert="false"
             nextOnFail="goalC" nextOnSucceed="goalB" id="goalA">
        <Search singleTarget="false" requiredPD="0.8"
              datumType="area" />
        <OperatingArea>
          <Rectangle>
            <NorthwestCorner>
              <XYPosition y="6425.0" x="12625.0" />
            </NorthwestCorner>
            <Width description="" value="50.0" />
            <Height description="" value="150.0" />
          </Rectangle>
        </OperatingArea>
        <Duration value="500.0" />
      </Goals>
```

Figure 11. At top is pseudo-code XML describing the canonical mission of Figure 2 and 4, showing that machine-readable and human-readable mission specifications are feasible. Below left is an excerpt from the fully validatable AVCL mission *MEAMission.xml* [16,19] showing precise AVCL syntax which is machine (or human)-readable. Lower right is matching XML tree view of same AVCL document.

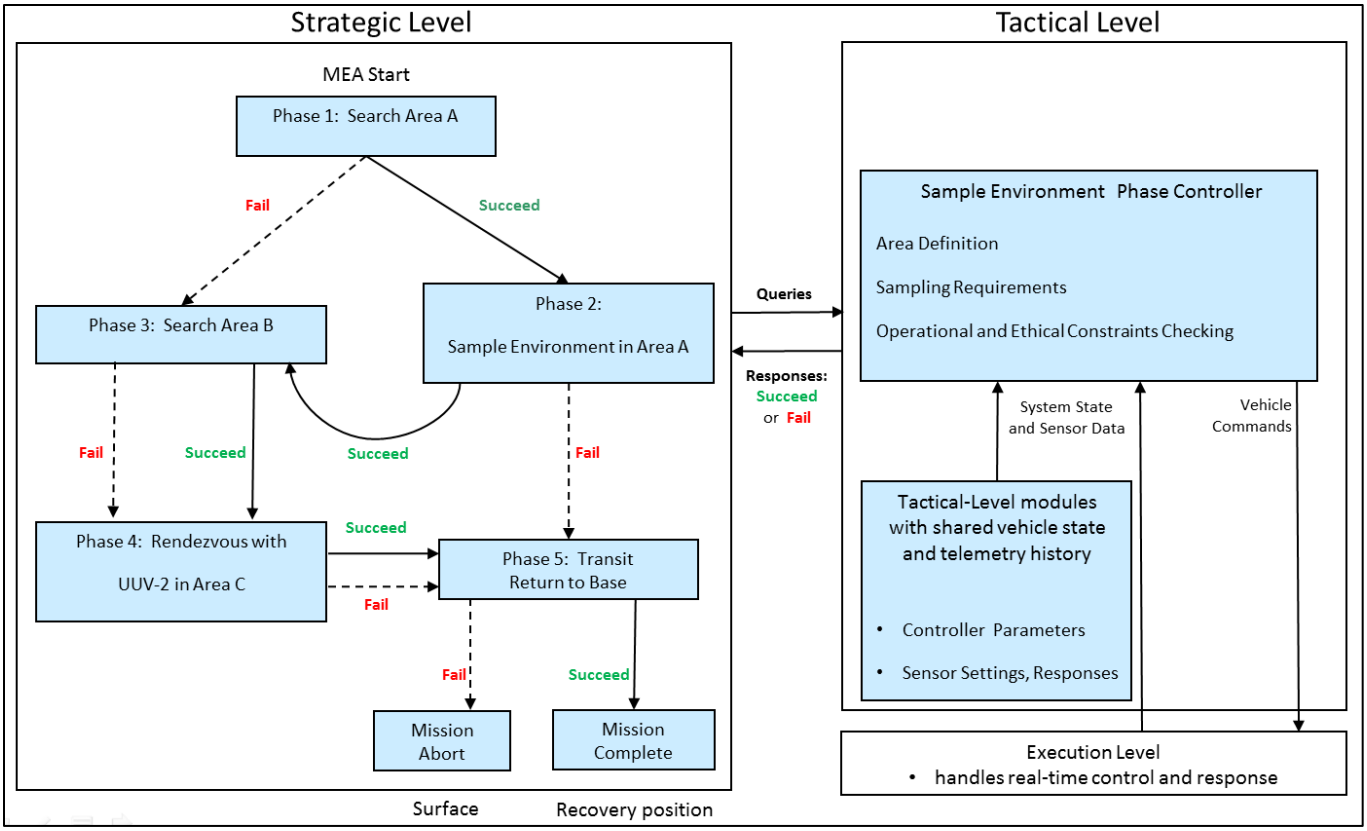

Figure 12. Canonical MEA mission on the left is sequenced by the Strategic Level controller, while the exemplar Tactical Level implementation on the right executes a corresponding series of ordered AVCL commands [12-13,15-19].

Figure 12 shows the original example MEA mission with emphasis added on logical outcome, along with the corresponding example RBM architectural implementation in the AUV Workbench. The left-hand side of Figure 12 shows mission logic, with conditional branching based on the success or failure of each individual tactical goal task. The AUV Workbench can successfully simulate the conduct of these examples.

The commonly shared interface design pattern for each goal type is illustrated in Figure 13. The essential design characteristic that each task type concludes in either success or failure is important, because:

- Goal modules are composable in any combination.
- Missions of arbitrary logical complexity can be designed simply by connecting task modules.
- Strategic mission-controller outputs consist of simple tactical-level commands that are each individually executable, either by unmanned or manned vehicles.
- A repertoire of missions can be collected that is experimentally testable, in simulation or in situ.
- Exported missions remain feasible for any variety of robot syntax or source code that can comply with common mission semantics used by humans.

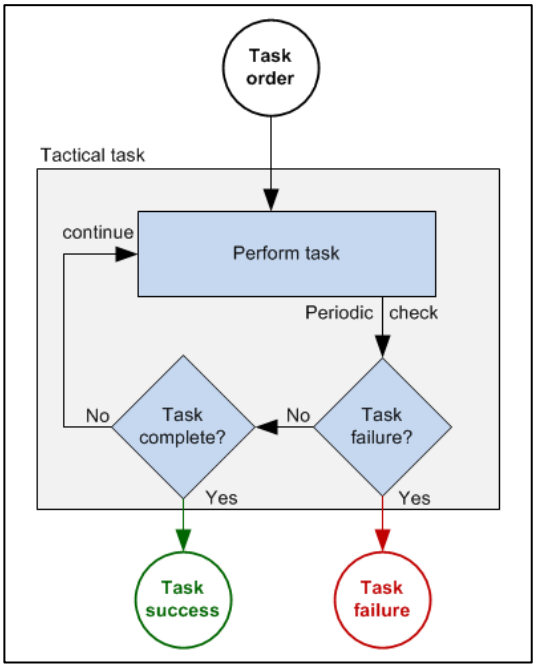

Figure 13. Internal logic and common input/output goal interfaces, enabling general task composability.

Mission Simulation using AUV Workbench

The NPS Autonomous Unmanned Vehicle (AUV) Workbench supports physics-based mission rehearsal, real-time task-level control of robot missions, and replay of recorded results in support of autonomous unmanned underwater, surface and air vehicles. Geographic information system (GIS) layers, a 2D geographic plot, X3D visualization and telemetry-plotting capabilities provide users with multiple in-depth mission portrayals. Extensible 3D (X3D) graphics and other open data standards are used throughout, implemented using open-source software for maximum repeatability and usefulness.

Four example missions are shown in Figure 14. In the upper left is a strategic-level mission that performs a series of logical decisions based on goal success. Across the bottom of the upper display are three UUV/USV missions. At bottom is a detailed view of the test mission.

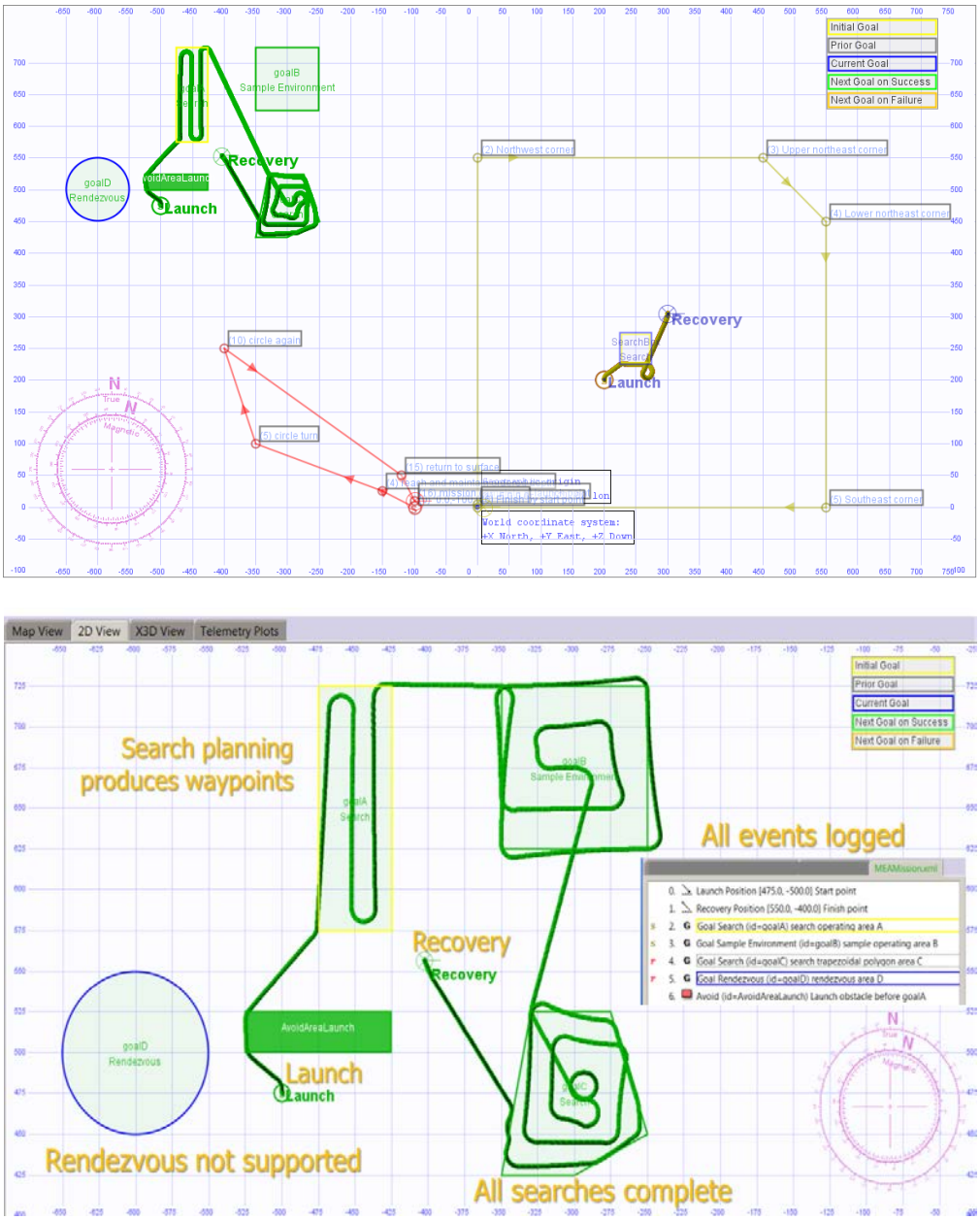

Figure 14. Multiple AVCL missions for UUVs and USVs displayed in the AUV Workbench. In upper left and at bottom, the canonical search/sample/search rendezvous mission is shown with annotations added at each major step [19].

Adding Ethics Constraints to Autonomous Missions

Taking Inventory. Examination of a wide variety of manned and unmanned vehicle missions has revealed that most ethical decisions are based on observing constraints on action. Figure 15 lists numerous requirements that must be met for a manned or unmanned vehicle to perform maritime operations in concert with other manned vehicles at sea. In some respects, military ethical constraints are a superset of civilian ethical constraints since safety of navigation and other requirements are common concerns. In every case, proper operation by unmanned systems remains a prerequisite for operation in concert with manned systems.

This effort led to a further key insight: ethical behaviors don't define a new class of missions or decision-making algorithm. Rather, ethical constraints inform regular mission plans with boundaries on action. Thus a practical new approach is feasible: integrating ethical constraints into mission definitions, rather than considering ethical operations as some new mode or paradigm of artificial intelligence. Ethical robot operations are thus similar to ethical manned operations: rules of conduct and rules of engagement must be observed throughout missions.

Civil ethical constraints	Define	Test	Notes
Mission tasking	√	partial	AVCL goals
Safe navigation and transit	√	√	AVCL avoidance areas
Follow pertinent rules of road			Requires rule-engine path planner, sensing model
Satisfactory navigational accuracy (GPS etc.)	√	√	Needed: sensor error models
Clearance to enter a specific geographic area	√	√	
Vertical clearance for underwater depth zone or airborne altitude zone	√	√	
Timing requirements using specific times or duration	√	√	
Sufficient vehicle health, power, safety status	partial	partial	
Meet communication requirements for tasking	partial	partial	Message-passing scheme
Identity beacon, transponder, AIS tracking, etc.			
Recording and reporting on situational data			

Military ethical constraints	Define	Test	Notes
Meet all relevant, international civil requirements	partial	partial	See above
Mission tasking	√	partial	AVCL goals
Contact identification, tracking signatures			Available in C2 systems
Identification friend foe (IFF), blue-force tracking (friendly/hostile/neutral/unknown/etc.)			Available in C2 data models
Robot option to warn without fear of self protection			Implementable via messaging
Determination of contact's hostile intent			Available in C2 data models, dissertation work in progress
Confirmation and permission requirements			Implementable via messaging
ROE use of deadly force, weapons releasability using brevity codes: weapons safe, hold, tight, free	partial	partial	Requires weapons model
Proportional weapons response			Requires weapons and threat models
After-action reporting	partial	partial	AVCL goals
Damage assessment			Requires models of interest

Figure 15. Common ethical constraints for civil and military operations, including AVCL definability and AUV Workbench support.

Implementation of Ethical Constraints in RBM

Prior work [15] described initial efforts to integrate ethical constraints within RBM missions. Figure 16 shows how the addition of simple constraint checking can be added to the text-based Prolog robot mission simulator. In this case, we took the combined controller/simulator examined in [15] and added a default ethical check (confirmed or denied by the human user) prior to performing each mission task.

- **Constraint 1:** If ethical search of Area A is not possible, go to Goal 4.
- **Constraint 2:** If ethical execution of Goal 3, 4, or 5 is not possible, abort the mission.

CG-USER(1): (tm) *; ethical test mission, run #1*
Search Area A!
Ethical execution possible? **n**o
Phase aborted. Unethical command.
Search Area B!
Ethical execution possible? **n**o
Mission aborted. Unethical command.

CG-USER(19): (tm) *; ethical test mission, run #2*
Search Area A!
Ethical execution possible? **y**es
Execute command!
Successful-1? **y**es
Sample environment!
Ethical execution possible? **y**es
Execute command!
Successful-2? ? **y**es
Search Area B!
Ethical execution possible? **y**es
Execute command!
Rendezvous UUV2!
Ethical execution possible? **y**es
Execute command!
Return to base!
Ethical execution possible? **y**es
At base? **y**es
Mission succeeded.

Figure 16. Modified RBM Prolog test program, originally based on mission definitions shown in Figure 3, integrating ethical checks into an RBM MEA mission controller. Task and ethics result responses (success or failure) are provided by a supervising human user. This approach enables manual testing of mission logic as well as corresponding ethical checks. Annotated, from [15].

Figure 17 shows the canonical mission revisited, showing how ethical constraints are both prerequisites and ongoing performance criteria for any task. The examination of multiple examples to date indicates that addition of constraints can superimpose ethical criteria on top of mission execution. This can be accomplished by straightforward modification of original mission. Each ethical-constraint violation equals a mission-task failure, thereby preserving the existing RBM control architecture for Strategic Level decision making.

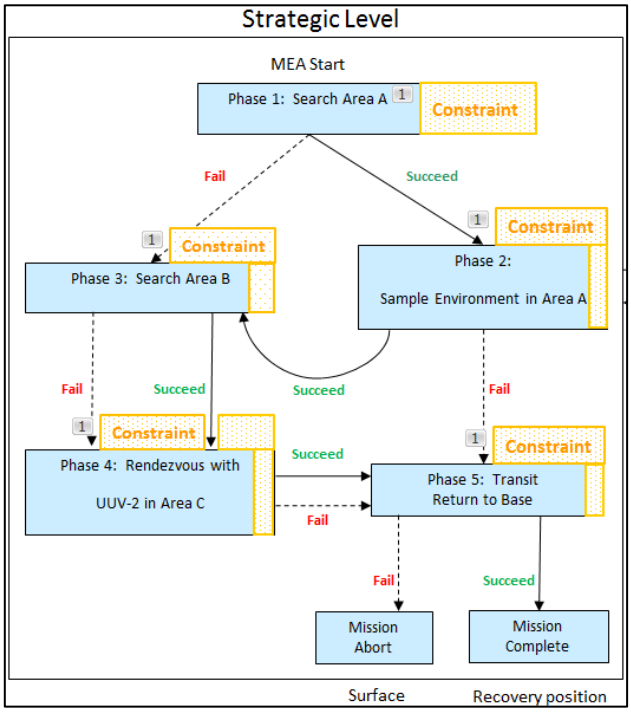

Figure 17. In-place addition of ethical constraints to the canonical search/sample mission already examined in Figures 2, 4, 11, 12 and 14. Constraints are prerequisites and operating conditions for each mission task. Here, each ethical-constraint violation equals a mission-task failure.

Note that this approach now shows how ethical questions can be posed as an integral part of mission task declarations. The answers to such constraint checks can be provided by (a) a supervising human critic, (b) a virtual environment conducting a simulation, or (c) on-board robot sensors. Therefore, in order to be meaningful, each ethical-constraint question must have a determinable answer. Responses to constraints (such as "Confirm outside of hazardous area") are then expressed as a single boolean result. Multiple constraints are possible for a single given task (e.g. adding "during the time window") but the overall result must be measurable as a boolean Success/Failure.

Based on this progress, even further evolution is possible in balancing the generality and composability of the basic RBM architectural control model. The current binary logic (*Succeed* or *Fail*) model might be compatibly extended, without loss of generality, to simplify these separate sets of recovery actions by calling out the decision logic for ethical-constraint failures as a first-class construct. Rephrased, MEAs might have 3 outputs (*Succeed*, optional *Ethics Fail*, *Task Fail*) that make it easier to distinguish between ethical-constraint failures (rather than task-performance failures). Figure 18 explores this possibility.

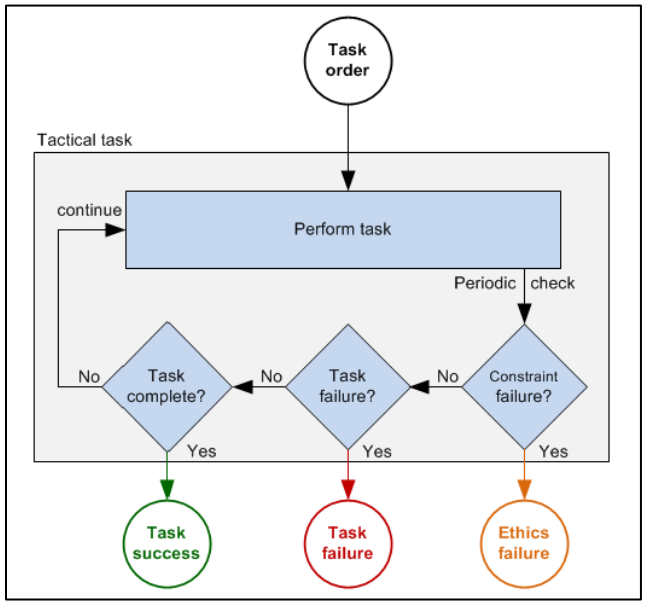

Figure 18. Potential modification to Figure 13, adding ethical constraint checking to internal logic and common input/output goal interfaces. General task composability is maintained without loss of generality. This modified task structure can directly expose and distinguish between operational-task failures and ethical-constraint failures.

Further mission design with corresponding simulation testing is needed to explore the tradeoffs associated with this enhanced construct. Conceivably this approach towards defining intertwined parallel paths for operational successes and ethical successes might even enable logical or legal after-action tracing of robot activities, thereby providing the potential to confirm compliance with legal, administrative and other ethical constraints.

Allowing human operators to analyze whether "the means taken justify the ends achieved" can directly expose the ethical constraints that were exercised in a given mission. These are promising directions deserving further exploration. Continued investigation is likely to close the gap between well-meant (but vaguely defined) philosophical concerns and implementable, strictly controlled, humanly accountable, legally justifiable deployment of autonomous systems.

Conclusions

Ethical control of maritime robot missions is feasible.

- Human mission tasking can be performed ethically, serving as a baseline pattern for robot mission design.
- Robot tasking must be consistent with human missions, both for compatible operations and also for human confidence that robot actions are permissible.
- Ethical constraints must be determinable.
- Common mission orders can be created that are parsable, executable and semantically consistent for any autonomous robot of interest.
- Ethical constraints can be applied to existing mission tasking to ensure that only authorized operations occur.
- No new reasoning paradigms or advance philosophical computational models need to be invented or embedded in existing robots. Ethical decision making lies in the definition of mission tasks and constraints.

Recommendations for Future Work

Multiple areas of work deserve further development.

- A library of missions and corresponding ethical constraints for unmanned systems needs elaboration and open publication for further testing by a variety of interested parties.
- Current work to catalog relevant ethical constraints need to be fully elaborated, tuned and aligned with primary references for civil and military operations including "Rules of the Road," safety guidelines, reporting criteria for situations requiring external human permissions, Rules of Engagement (ROE), etc.
- High-fidelity physically based simulation needs to complement and (where possible) precede at-sea testing to ensure that operational safety requirements for unmanned vehicles are met when mission tasks are constrained by ethical criteria.
- AVCL development needs to build and compare the effectiveness of mission task and source code exporters corresponding to a large variety of dissimilar robots.
- Naval and civil program plans for future unmanned systems need to include support for ethical constraints.

References

[1] Arkin, Ronald C. (2009), *Governing Lethal Behavior in Autonomous Robots*. Boca Raton, FL: Chapman & Hall/Taylor & Francis Group. Also Arkin, Ronald C. (2010). "The Case for Ethical Autonomy in Unmanned Systems," G.R. Lucas, Jr., ed. "New Warriors and New Weapons: Ethics & Emerging Military Technologies," *Journal of Military Ethics*, vol. 9, no. 4 (December 2010): 332-341. See also Ronald Craig Arkin, Patricik Ulam, and Alana R. Wagner, "Moral Decision Making in Autonomous Systems: Enforcement, Moral Emotions, Dignity, Trust, and Deception." *Proceedings of the IEEE*, 100/3 (March 2012): 571-589.

[2] Lucas, George R. Jr. (2013). "Ethics, Engineering, and Industry: the Moral Challenges of Lethal Autonomy," in Bradley J. Strawser, editor, *Killing by Remote Control* (Oxford: Oxford University Press, 2013) pp. 221-228.

[3] Strawser, Bradley J. "Moral Predators: the Duty to Employ Uninhabited Aerial Vehicles." in G.R. Lucas, Jr., ed. "New Warriors and New Weapons: Ethics & Emerging Military Technologies," *Journal of Military Ethics*, vol. 9, no. 4 (December 2010): 357-383.

[4] Sharkey, Noel (2008). "Grounds for Discrimination: Autonomous Robot Weapons", RUSI Defence Systems, 11:2, 86-89. Sharkey, Noel (2010). "Saying 'No!' to Autonomous Lethal Targeting," in G.R. Lucas, Jr., ed. "New Warriors and New Weapons: Ethics & Emerging Military Technologies," *Journal of Military Ethics*, vol. 9, no. 4 (December 2010): 342-368.

[5] Singer, P. W. (2009). *Wired for War*. New York: Penguin Press. Also Singer, P.W. (2010). "The Ethics of 'Killer Apps'" in G.R. Lucas, Jr., ed. "New Warriors and New Weapons: Ethics & Emerging Military Technologies," *Journal of Military Ethics*, vol. 9, no. 4 (December 2010): 299-312.

[6] Wallach, Wendell (2013). "Terminating the Terminator: What to do about Autonomous Weapons." *Science Progress*, vol. 29, January 2013: http://scienceprogress.org/2013/01/terminating-the-terminator-what-to-do-about-autonomous-weapons [accessed 31 July 2013]. Also Wallach, Wendell and Allen, Colin (2009). *Moral Machines: Teaching Robots Right from Wrong*. New York: Oxford University Press.

[7] Sparrow, Robert (2007). "Killer Robots", *Journal of Applied Philosophy*, vol. 24, no. 1, 62-77. Also Sparrow, Robert (2008). "Building a Better WarBot: Ethical Issues in the Design of Unmanned Systems for Military Applications," *Science and Engineering Ethics*, vol. 15, no. 2: 169-187. Also Sparrow, Robert (2011). "Robotic Weapons and the Future of War," *New Wars and New Soldiers: Military Ethics in the Contemporary World*. Eds. Paolo Tripodi and Jessica Wolfendale. London: Ashgate Publishing Ltd. Pp. 117-133.

[8] Byrnes, R., *The Rational Behavior Model: A Multi-Paradigm, Tri-level Software Architecture for the Control of Autonomous Vehicles*, Ph.D. Dissertation, Computer Science, Naval Postgraduate School, March 1993.

[9] Marco, D.B., Healey, A.J., and McGhee, R.B., "Autonomous Underwater Vehicles: Hybrid Control of Mission and Motion", *Autonomous Robots*, vol. 3, pp. 169-186, 1996.

[10] Brutzman, D., et al, "The Phoenix Autonomous Underwater Vehicle", *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, Ch. 13, pp. 323-360, ed. by Kortenkamp, D., et al., MIT Press, Cambridge, MA 02142, 1998.

[11] Davis, D., Becker, W., and Brutzman, D., "Facilitation of Autonomous Vehicle Coordination through an XML-Based Vehicle-Independent Control Architecture", *Proceedings of the 16th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, August, 2009.

[12] McGhee, R., Brutzman, D., and Davis, D., "A Universal Multiphase Execution Automaton (MEA) with Prolog Implementation for Unmanned Untethered Vehicles", *Proceedings of the 17th International Symposium on Unmanned Untethered Submersible Technology*, Portsmouth, NH, August 2011.

[13] McGhee, R.B., Brutzman, D.P., and Davis, D.T., *A Taxonomy of Turing Machines and Mission Execution Automata with Lisp/Prolog Implementation*, Technical Report NPS-MOVES-2011-2, Naval Postgraduate School, Monterey, CA 93943, April, 2011.

[14] Minsky, M.L., *Computation: Finite and Infinite Machines*, Prentice Hall, 1967.

[15] Brutzman, D., McGhee, R., and Davis, D., "An Implemented Universal Mission Controller with Run Time Ethics Checking for Autonomous Unmanned Vehicles—a UUV Example", *Proceedings of the OES-IEEE Autonomous Underwater Vehicles 2012*, September 2012, Southampton, England.

[16] McGhee, R.B., Brutzman, D.P., and Davis, D.T., *Recursive Goal Refinement and Iterative Task Abstraction for Top-Level Control of Autonomous Mobile Robots by Mission Execution Automata - A UUV Example*, Technical Report NPS-MV-12-001, Naval Postgraduate School, Monterey, CA 93943, March 2012.

[17] Davis, Duane T., Design, *Implementation and Testing of a Common Data Model Supporting Autonomous Vehicle Compatibility and Interoperability*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, September 2006.

[18] Brutzman, Donald P., *A Virtual World for an Autonomous Underwater Vehicle*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, December 1994.

[19] Brutzman, Donald P., NPS Autonomous Underwater Vehicle (AUV) Workbench, software application archive, Naval Postgraduate School, Monterey California, 1994 to present.

Published NPS references, software and mission examples are maintained online at https://savage.nps.edu/AuvWorkbench