Ethical Mission Definition and Execution for Maritime Robotic Vehicles: A Practical Approach

Duane Davis, Don Brutzman, Curtis Blais, and Robert McGhee
Naval Postgraduate School
Monterey California 93943-5000, USA
dtdavi1@nps.edu, brutzman@nps.edu, clblais@nps.edu, robertbmcghee@gmail.com

***Abstract*—Many types of robotic vehicles are increasingly utilized in both civilian and military maritime missions. Some amount of human supervision is typically present in such operations, thereby ensuring appropriate accountability in case of mission accidents or errors. However, there is growing interest in augmenting the degree of independence of such vehicles, up to and including full autonomy. A primary challenge in the face of reduced operator oversight is to maintain full human responsibility for ethical robot behavior.**

Informed by decades of direct involvement in both naval operations and unmanned systems research, this work proposes a new mathematical formalism that maintains human accountability at every level of robot mission planning and execution. This formalism is based on extending a fully general model for digital computation, known as a Turing machine. This extension, called a Mission Execution Automaton (MEA), allows communication with one or more "external agents" that interact with the physical world and respond to queries/commands from the MEA while observing human-defined ethical constraints.

An important MEA feature is that it is language independent and results in mission definitions equally well suited to human or robot execution (or any arbitrary combination). Formal description logics are used to enforce mission structure and semantics, provide operator assurance of correct mission definition, and ensure suitability of a mission definition for execution by a specific vehicle, all prior to mission parsing and execution. Computer simulation examples show the value of such a Mission Execution Ontology (MEO).

The flexibility of the MEA formalism is illustrated by application to a prototypical multiphase area search and sample mission. This paper presents an entirely new approach to achieving a practical and fully testable means for ethical mission definition and execution. This work demonstrates that ensuring ethical behavior during mission execution is achievable with current technologies and without requiring artificial intelligence abstractions for high-level mission definition or control.

Keywords—robot ethics; mission specification; unmanned systems; robotic systems

I. INTRODUCTION

Many experts and practitioners have worked long and hard towards achieving functionally capable robots. Unfortunately, progress in ethical control of unmanned systems has been elusive and problematic. Common paradigms that assume an always-amoral robot or that require undemonstrated morality-based artificial intelligence (AI) are equally untenable. For better or worse, actors around the world are rapidly designing and deploying mobile unmanned systems to augment human capabilities. Thus theory must meet practice.

This paper adapts policies and procedures for ethical responsibility and authority that have been proven to work effectively in collaborative military operations. Since ethical responsibility is not limited to military weapons but can also apply to even routine task completion, this approach appears to have broad usefulness for civil application of unmanned systems as well.

The authors' experience across four decades of robotic and military operations has demonstrated that robot mission tasks and goals can be clearly defined and refined with corresponding degrees of internal control supervision. Adding well-specified constraints can supplement mission orders, providing an ethical basis for unmanned system tasking that matches human understanding of similar responsibilities. Careful structuring of mission orders in the form of a mathematical construct called a Mission Execution Automaton (MEA) demonstrates a theoretically sound and scalable basis to this approach. Further, a Mission Execution Ontology based on principles of description logics provides mathematical assurances that mission definitions are semantically complete, including ethical constraints whenever appropriate.

The advent of digital computing, the emergence of AI, and the incorporation of both into a variety of robotic devices have brought ethical concerns to the forefront of debate. It has become quite apparent that legal and moral responsibility cannot be addressed by a set of fixed "laws" intended to constrain robot behavior. Reasoning about abstract situations from high-level principles [1] is beyond current capabilities to describe (much less arbitrate), and such essential imperatives cannot be regulated into irrelevance. Nevertheless, a number of important observations inform this work:

- Robots do what programmers and operators tell them to do—not what programmers and operators mean to tell them to do.
- Apparent intelligence notwithstanding, a robot is an inanimate object and cannot assume moral or legal responsibility for an action's consequences.
- For robot ethics to bear any tangible meaning, ultimate accountability must reside with the human programmers, manufacturers, and operators.
- Legal and moral liability requires that involved parties are in a position to reasonably foresee the outcomes for which they are being held responsible.

These observations are fairly widely accepted, but nevertheless can still lead ethicists to different conclusions. In debating military use of autonomous systems, for instance, Rob Sparrow of the International Committee for Robot Arms Control applies Jus en Bello requirements to argue that the military use of lethal robots is inherently unethical because robots cannot be held accountable for their actions [2]. Ronald Arkin, on the other hand, accepts the premises of Sparrow's argument but comes to the opposite conclusion—that if an autonomous system is capable of making a lethal decision more reliably than a human, then it is inherently unethical to not use that system [3]. This work demonstrates that such observations can inform a broader framework for ethical operation of intelligent robots, one that is realizable with current technologies and guided by human ethical decision making, without any requirement for black-box artificial intelligence that inevitably leads to second-guessing.

II. MILITARY OPERATIONS AS AN ANALOGY

Authority and responsibility in military operations provide a useful analogy. Military commanders are provided forces over which they exercise control, are assigned missions that they are expected to accomplish, and are held responsible for the proper employment of all assigned assets. More recently, militaries have relied on increasingly automated systems, however, automation does not obviate the commander's responsibility. Ultimately, it does not matter whether a military leader is employing a system of people or a system of machines: authority implies responsibility [4].

Accountability in military operations requires a level of trust that is based on a number of important factors. First, subordinate units must be qualified for the designated task requirements. Second, mission orders must be unambiguous and fully understood by tasked units. Finally, subordinate units must accurately assess task progress during execution. This trust relationship provides assurance that properly employed subordinates do not impose undue risk. Improperly employed subordinates, on the other hand, do impose undue risk for which the commander is rightfully held responsible.

A variation of this ethical mechanism can be applied to robots in both military and civilian applications as a corollary to the well-established legal principle of vicarious liability [5]. That is, operators of autonomous and unmanned systems can be held responsible for undesirable outcomes that they are in a position to prevent.

All robots possess a finite set of operational and sensory capabilities, the complexity of which varies from robot to robot. It is reasonable, then, to trust a robot to execute those atomic capabilities. It follows that an operator can assume moral and legal liability for missions comprised of these trusted capabilities [6]. For this framework to be practically feasible, robot operators must be provided adequate mission assurance and understanding to assume responsibility. This can be achieved by meeting three requirements:

- Robot missions must be defined in a mathematically sound manner that ensures that the mission will progress as intended in all circumstances.
- There must be no means by which an approved mission can be semantically modified between approval and execution by the target vehicle.
- Mission tasking and associated constraints must be comprised entirely of trusted atomic vehicle-specific behaviors, and the vehicle must be able to continually evaluate both behavior and constraint status at run time.

III. MISSION DEFINITION AS GOALS WITH RUN-TIME CONSTRAINTS

In describing complex tasks, humans often divide them into series of subordinate tasks to be executed in order. For instance, a complex task during which a manned vehicle is to conduct searches and collect environmental samples before rendezvousing with another vehicle might be specified as depicted in Fig. 1. Providing the vehicle's operator understands the individual subtasks, the mission can be reliably executed.

The vehicle operator implicitly relies on a discrete decision process to periodically take stock of the situation, determine the current task's status, and proceed to the next task when appropriate. This decision process is commonly referred to as an Observe-Orient-Decide-Act (OODA) loop when referring to military and other human operations, or as a Sense-Decide-Act (SDA) loop when referring to autonomous agent activities [7] [8].

One aspect of the above mission must be accounted for, however, to support execution by an autonomous agent: the implicit assumption of success for each task. When facing task failure, a human is able to use best judgement or request guidance from higher authority. This is not necessarily an option for autonomous agents. Rather, the appropriate course of action must be included in the mission description. This can be achieved by specifying subsequent alternative tasks that execute in the event of any particular mission task's success or failure.

Branching based on task success or failure results in missions whose execution is characterized by a sequence of successful and unsuccessful task executions. It is appropriate, then, to refer to the individual tasks as goals to be achieved rather than simply as tasks. A possible modification of the mission from Fig. 1 is provided in Fig. 2. Interestingly, the

Task 1: Proceed to Area A and search the area.

Task 2: Obtain an environmental sample from Area A.

Task 3: Proceed to Area B and search the area.

Task 4: Proceed to Area C and rendezvous with vehicle 2.

Task 5: Proceed to recovery position (mission complete).

Fig. 1. Example mission orders expressed in structured natural language for human execution.

SDA/OODA decision loop is still suitable for execution control of this revised mission.

Fig. 2 gives rise to a graphical representation of the natural-language mission definition. The flow graph of Fig. 3 is one among many potential representational forms for this and many other missions. It is of particular interest because it provides an intuitive depiction of a potentially complex mission. In fact, this mission specification can be used to mentally "rehearse" the mission by intentionally traversing the graph from start to finish while testing success and failure branches at every step. While not yet providing mathematical rigor, this ability to informally traverse task sequences is an important step towards providing assurance to the responsible operator that the mission will progress according to human intent.

As presented so far, this mission definition paradigm does not explicitly address the issue of ethical mission execution. Specifically, no mechanism has been suggested at this stage to define ethical constraints affecting the overall mission or individual tasks. For instance, it is apparent that an unmanned underwater vehicle (UUV) with an appropriate search behavior can achieve goals 1 and 3 of the example mission. Unfortunately, it may or may not be able to do so while avoiding detection, remaining clear of other vehicles, or

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, execute Goal 5.

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

Goal 4: Proceed to Area C and rendezvous with vehicle 2. Upon rendezvous success or failure, execute Goal 5.

Goal 5: Proceed to recovery position (mission complete). Upon successful arrival, mission complete. If unable to return to base, abort the mission.

Fig. 2. Modified search and sample mission providing success-failure branching and human or autonomous agent execution [17].

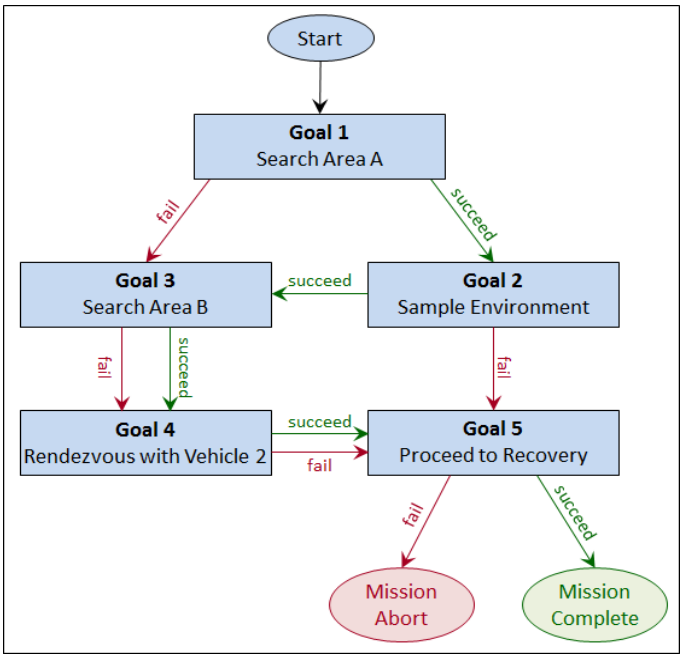

Fig. 3. Mission-flow graph for search and sample mission, human or autonomous agent execution [19].

maintaining a specific navigational accuracy. If any of these or any other ethical constraints are to be applied to the mission, then they must be incorporated into the mission specification. Further, from the standpoint of operator accountability, constraints must be specified in a manner that preserves the ability to trace high-level mission flow and in a way that is enforceable by the target vehicle.

An operator may determine that certain constraints must be enforced from launch until recovery (e.g., all safety systems must remain operational), while others only need to be enforced during the execution of specific goals (e.g., maintaining safety depth in the search area). As an example, mission- and goal-level constraints might be applied to the example UUV mission as depicted in Fig. 4. This construct still supports the prior rehearsal of missions and also allows for the in situ consideration of whole-mission and goal-specific constraints [9].

Under the binary branching model of Fig. 3, an impending ethical constraint violation implicitly equates to goal failure. This might be acceptable, but it might be desirable to treat impending constraint violations differently than simple failure. A third potential goal-execution outcome and a corresponding branching option in the mission flow structure might be useful. That is, execution of an individual goal terminates upon goal success, goal failure, or impending violation of a constraint applied to that goal. A modification of the example mission to implement this ternary branching model is graphically depicted in Fig. 5.

The flow graph mission specification described here is declarative. At this level of abstraction, individual goals execute sequentially according to the mission graph irrespective of time, and each goal predictably terminates in one of three possible states. This approach eliminates any need to make assumptions or guesses concerning intended vehicle

Constraint 1: The vehicle must maintain navigational accuracy within acceptable limits. Applies to entire mission.

Constraint 2: All safety equipment must be fully functional. Applies to entire mission.

Constraint 3: All mission systems must be operational. Applies to Goal 1, Goal 2, and Goal 3.

Constraint 4: Acceptable distance from shipping lanes in the form of 1000 meter lateral standoff or minimum depth of 20 meters must be maintained. Applies to Goal 1, Goal 2, Goal 3, and Goal 4.

Constraint 5: Must be able to detect surface contacts within 5000 meters. Applies to entire mission.

Constraint 6: Detected surface contacts are to be avoided by a minimum of 1000 meters. Applies to Goal 1, Goal 2, Goal 3, and Goal 4.

Constraint 7: Minimum depth of 20 meters is to be maintained. Applies to Goal 5.

Fig. 4. Constraints suitable for application to the example search and sample mission.

conduct during goal execution. Rather, the onus is placed on human operators to create well-defined and thorough missions. Further, if the size of the mission-flow diagram is reasonably managed, then exhaustive testing of all possible mission execution sequences is achievable and tractable.

IV. MISSION EXECUTION AUTOMATA (MEA): EXECUTABLE MISSION SPECIFICATIONS

A. *The Rational Behavior Model (RBM) Robot Control Architecture*

Flow-diagram expression of autonomous vehicle missions is particularly compatible with the highest level of abstraction for a number of proposed hierarchical robot control architectures [10] [11] [12] and has been utilized extensively with the Rational Behavior Model (RBM) [13]. RBM organizes robot control requirements into strategic, tactical, and execution levels. The strategic level controls high-level mission flow, accomplishing high-level tasks by issuing commands to the tactical level. The tactical level executes self-contained behaviors in response to strategic-level commands, notifying the strategic level of command outcomes (success, failure, or constraint violation). The execution level provides real-time control of actuators and sensors as directed by the tactical level. Evidently, the strategic level is well suited to execute mission-flow diagrams, and tactical-level behaviors can be comprised of atomic capabilities of a particular target vehicle.

If properly encoded, the strategic-level mission-flow diagram forms an executable mission specification. This human-and-machine compatible form is in line with ethical framework requirements and provides for human-based testing of mission code prior to robot execution. Further, from the perspective of the strategic level, it does not matter whether the tactical level behavior is executed by an actual robot, a computational model, or a human being, making full-fidelity mission-flow-diagram rehearsal possible.

B. *Mission Execution Automaton (MEA) Definition*

Mathematical rigor of strategic-level execution steps is obtained by observing that the run-time traversal of the mission-flow diagram is similar to the operation of a mathematical formalism called a Turing Machine (TM) [14]. TMs have a number of fundamental properties that are particularly important in the field of computer science [14] [15] but are generally considered impractical for real-world utilization [16]. They do, however, provide a strong theoretical foundation upon which to build a mathematically sound strategic-level mission-flow diagram execution mechanism suitable for both robots and human operators. With some modification, TM execution mechanics can provide an execution engine for mission-flow diagrams expressed as finite state machines (FSM) [14] [15]. More specifically, the mission-flow diagrams can be viewed as robot programs that are executable using TM semantics. Relaxation of some additional TM formalisms yields a mathematical construct that subsumes the more constrained TM [16]. This TM generalization is referred to here as a Mission Execution Automaton (MEA) which consists of a Mission Execution Engine (MEE) and an arbitrary set of mission orders in FSM form, and a communication link to at least one human or robot external agent capable of carrying out orders from a given finite set (as issued by the MEE) and returning results from another finite set [17] [18].

C. *Strategic-Level Mission Rehearsal and Testing*

MEA implementations were initially developed in Lisp and Prolog [19] [16] [18]. The declarative nature of Prolog in

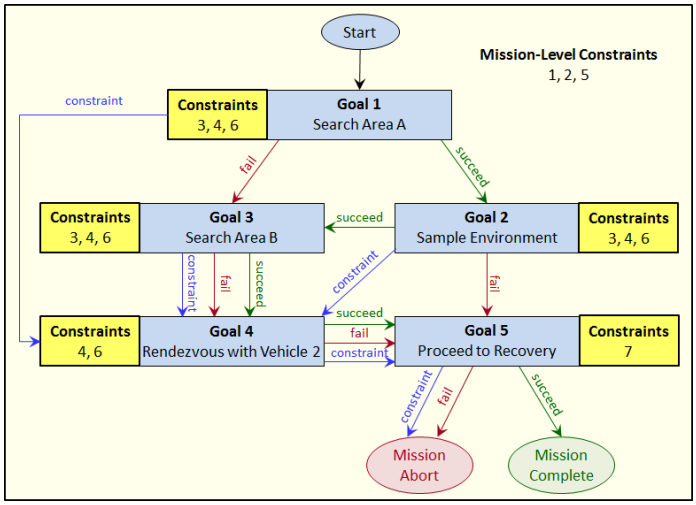

Fig. 5. Mission-flow graph for a search and sample mission with ternary branching for handling imminent ethical-constraint violations.

particular aligns well with the mission-flow diagram and makes Prolog mission orders intuitively understandable by non-programmers. Strengths notwithstanding, it is important to note that there is nothing inherently unique about Prolog, and the MEE and mission orders can be accurately created with any Turing-complete computer language [20] [9] [21].

A similarly capable, independent implementation uses the Hierarchical Task Network (HTN) behavior model and Python programming language in the Combined Arms Analysis Tool for the 21st Century (COMBATXXI), a simulation tool developed and used by the U.S. Army and U.S. Marine Corps within various analytic studies [22] [23]. Like the earlier Prolog simulation, the COMBATXXI simulation allows testing of the Strategic Level mission flow by a human operator, with typical results depicted in Fig. 6. In the simulation, the Strategic Level orders commencement of individual goals and the human operator reports success, failure, or constraint-based termination of each goal. The depicted traces in Fig. 6 correspond to instances where each goal terminates due to potential constraint violation and where each goal completes successfully.

This ability to formally test a mission definition by allowing a human operator to assume the role of the tactical level provides assurances that the branching upon individual goal success and failure matches the operator's intent. The previously discussed premise that ethical operation requires formal assurance that the mission will proceed as intended in all circumstances is thus satisfied so long as mission-flow graph size and structure is constrained so that it is exhaustively testable.

Based on this experience it is reasonable to conclude that flow graphs represent a higher level of abstraction for mission specification than any unconstrained text-based coding language. Moreover, advances in graphical coding [24] may eventually allow non-programmers to completely specify robot missions by constructing a flow graph such as Fig. 5 directly on a computer screen. Such advances can further enhance the comprehension, supervision and accountability of mission experts for producing legally valid mission definitions.

D. Progressive Refinement of Complex Mission Tasks

Referring to Fig. 5, it is apparent that Goal 1 is achievable only if the person or software at the tactical level has considerable knowledge about Area A and how to search it. Stated differently, it can be completed only if a tactical-level behavior can be invoked that will execute the search appropriately. To make this concrete, suppose that Area A contains hazards that are potentially harmful to (or impassible by) the search vehicle and that no current map of the area of interest is available. A classic algorithm for exploring for such circumstances is depth first search [25]. Such a search first discretizes the search area into a finite set of cells and proceeds by systematically moving the vehicle between cells. At each step, the search tests the currently occupied cell to see if the search object is there (success). If it is not, the search proceeds by moving the vehicle into a previously unexplored, adjacent cell. If no such cell exists, then the vehicle retreats to the previous cell and the process continues. If the vehicle finds itself at the starting location with no adjacent unexplored cells, then the search is complete (failure).

Fig. 7 depicts a flow graph for the above-described process, and further represents a refinement of Goal 1 of Fig. 5. If trusted vehicle behaviors corresponding to each of the figure's goals exist, they might be commanded by a human operator acting on behalf of the tactical level (example available in [18]). Alternatively, this depiction may be understood as a tactical level implementation capable of autonomous execution of Goal 1. Note that all applicable constraints must be continuously observed throughout execution.

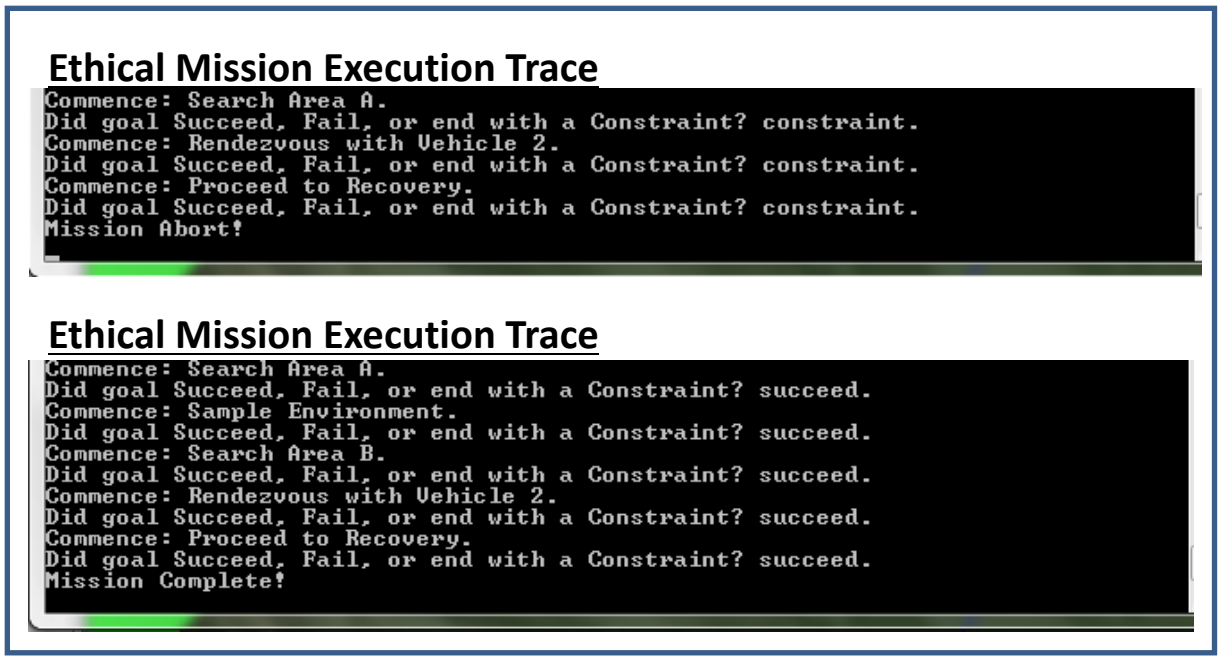

Fig. 6. Human interaction with a Strategic Level mission results obtained from the COMBAT XXI simulation.

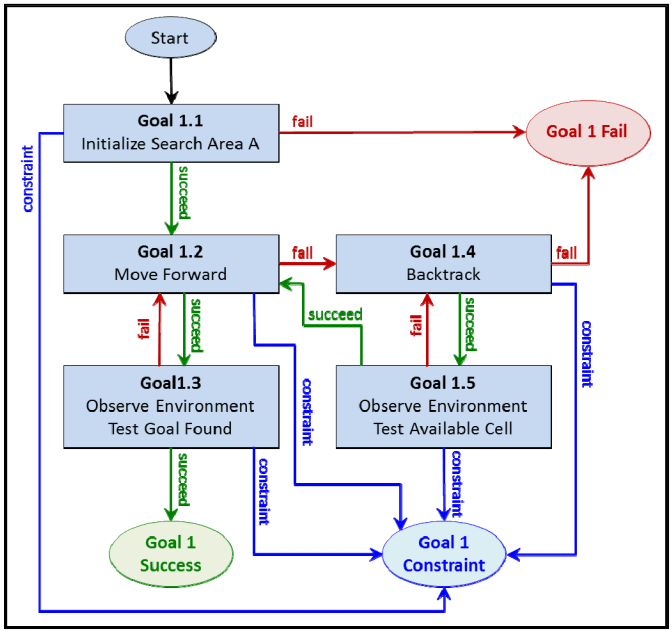

Fig. 7. Flow graph for a grid-based depth-first search of Area A developed through progressive refinement, adapted from [18].

This task decomposition raises an important implementation question: does Fig. 7 accurately represent the semantics of the desired behavior? Interestingly, since the flow diagram contains several decision loops, answering that particular question cannot be confirmed by exhaustive testing as was possible for the strategic-level mission orders. Such test limitations occur because, in general, there is no guarantee that such a search will always terminate for arbitrary terrain, targets and obstacles. Fortunately it is known that for a finite search area, depth first search will eventually terminate with either success or failure in searching for a specified goal [25]. Nevertheless, exhaustive testing for all possible terrain samples is not possible. Instead, the most that can be asked for is to show that all phase transitions in the given flow graph are correct [18].

It turns out that the inability to exhaustively prove correctness of a tactical-level flow graph is not as serious as it might at first appear. This is because tactical-level algorithm failure is no different than other outcomes that might cause a strategic-level goal to fail. Since strategic-level goal failure is accounted for in the overall mission-flow graph, the mission can still continue as planned. To make absolutely sure that such dependability occurs, a time-out condition resulting in goal failure must be incorporated at the tactical level [13].

Regardless, manual execution of progressively refined behaviors defined as flow graphs provides a mechanism for extending existing tactical-level behaviors. That is, once a flow graph accomplishing a specific purpose (e.g., depth-first search) has been suitably vetted, it effectively becomes a trusted tactical-level behavior itself. This means reinterpreting tactical-level flow graphs as code specifications or templates rather than actual code to be executed [18].

V. VALIDATION OF RBM SOFTWARE ARCHITECTURE THROUGH OPEN-OCEAN EXPERIMENTS

A. *Phoenix Autonomous Underwater Vehicle (AUV)*

Up to this point, presented results have related to the strategic level and the tactical level of RBM software for a single example of a "search and sample" mission for a notional UUV with results obtained through simulation. However, beginning in 1993, in parallel with formalization and publication of details of RBM [13], the value and practicality of this approach was demonstrated through a series of open-ocean experiments involving two small unmanned submarines.

The Phoenix AUV was an unmanned submarine designed and built at the Naval Postgraduate School (NPS) beginning in 1990. Weighing approximately 500 pounds, Phoenix included four cross-body thrusters, enabling active control of five degrees of motion (x, y, z, pitch and yaw) [26].

Large numbers of high-fidelity physics-based simulations were needed to correctly develop and test what were then considered AI approaches to replace human supervision. While this simulation involved distinct mission phases in the form of a command "script", similar to Fig. 1, no binary flow graph with phase failure contingencies was abstracted from these phases making exhaustive testing (and thus proof of correctness) of a mission impossible. Moreover, no concept of ethically constrained behavior was attempted in any of this work.

B. *Aries AUV*

Lessons learned from the Phoenix UUV were incorporated into the second-generation NPS UUV, the Aries. Specifically designed for open-ocean surveys, Aries was a somewhat larger vehicle that utilized more efficient forward thrusters but lacked cross-body thrusters [20]. An extensive and accurate physically based model of the vehicle and its environment, with three-dimensional (3D) graphical display, the Autonomous Unmanned Vehicle (AUV) Workbench, was developed and used for real-time testing of robot mission software [9] [21] [27].

Aries AUV missions were defined with the NPS-developed Autonomous Vehicle Command Language (AVCL), a schema-constrained XML data model supporting autonomous vehicle mission implementation, execution, and management [20]. While the mathematical concept of an MEA had not been developed at the time of AVCL's development, a fixed set of goal types is provided and branching upon goal completion is supported as well. Thus, AVCL is suitable for the realization of mission flow diagrams. Further, AVCL was intentionally designed to support implementation of RBM strategic and tactical levels and was utilized to define RBM-controlled Aries missions for AUV Workbench simulation and real-world open-ocean tests.

Simulation of a mission consisting of an AVCL specification for a search goal similar to Goal 1 of Fig. 5, while steering clear of avoidance areas specified as constraints in the AUV Workbench, is shown in Fig. 8. During the mission, the tactical level plans a path and maneuvers to the search area

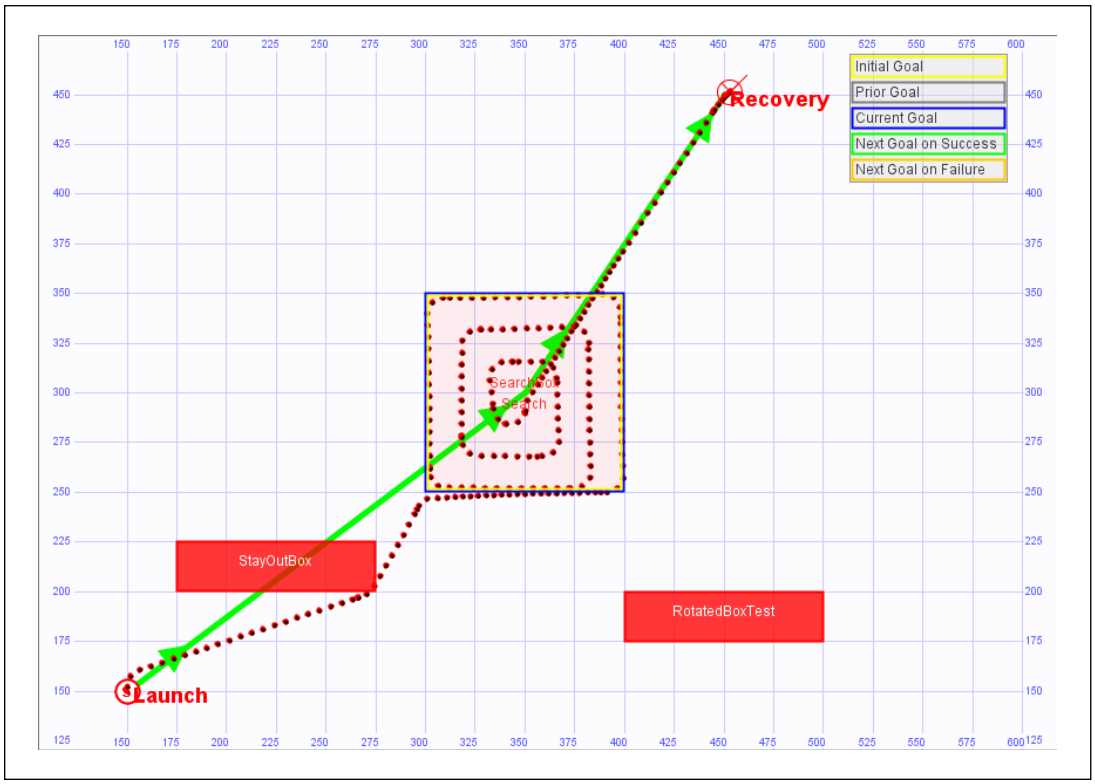

Fig. 8. A UUV mission defined using Autonomous Vehicle Command Language (AVCL) depicting simulated conduct of a goal-oriented mission with constraints [9].

while remaining clear of the avoid areas and then develops and executes a suitable pattern for the required area search. More complicated missions demonstrating the binary branching model were conducted in AUV Workbench simulations and also in open-ocean experiments in Monterey Bay [20].

VI. MISSION SPECIFICATION VALIDATION

A. *Description Logics (DL) and a Robot Mission Ontology*

Thus far, this discussion has focused on providing robot operators the ability to rigorously define and test strategic-level missions. The premises upon which this proposed framework for ethical robot tasking rests also require that an actual target vehicle can execute these missions without further translation. Fortunately, mathematical logic provides a mechanism for bridging strategic level missions and vehicle-specific code for specifying and ordering tactical-level behaviors.

Description Logics (DL) are a mathematical family of logic-based knowledge representation systems for describing concepts and roles within a system. DL ontologies can define the relationships in a system, how they operate, how they are used, and to what specific entities they apply. DLs are carefully and strictly defined to enable computationally efficient reasoning that can identify hidden relationships and errors, such as rule violations or contradictions [28].

DLs provide the foundation of the Semantic Web, a set of standards-based extensions to the World Wide Web that leverage DLs' expressiveness and rigor to provide extensive knowledge representation, discovery, and utilization capabilities [29]. Notably, the Web Ontology Language (OWL) [30] encodes a particularly powerful DL in a validatable plain-text computer-readable form [31]. OWL is used here to define a robot mission description and execution ontology that applies and enforces MEA semantics.

B. *Mission Execution Ontology (MEO)*

A Mission Execution Ontology (MEO) serves a number of purposes. First, it enables a formal and semantically rich description of the characteristics of a MEA mission description. For instance, OWL expressions are used to declare the existence of concepts such as "Mission", "Goal", and "Constraint" and also to define possible relationships between concepts. As an example, a "Mission" entity must have an "includes" relationship with at least one "Goal" entity and must have a "startsWith" relationship with exactly one of those entities. A partial graphical depiction of the concepts and relationships defined in the MEO is provided in Fig. 9.

In addition to the concepts abstracted directly from mission-flow diagram semantics, the MEO introduces the "Vehicle" concept providing for the inclusion of specific target vehicles in the mission-planning process. The "canExecute" and "canIdentify" relationships allow mission planners to explicitly assert that the intended target vehicle has a tactical-level behavior capable of completing a goal and recognizing potential violation of a constraint, respectively. Using this concept and relationship semantics enforced by MEO rules, it is impossible to define a valid mission that cannot be executed by the intended vehicle.

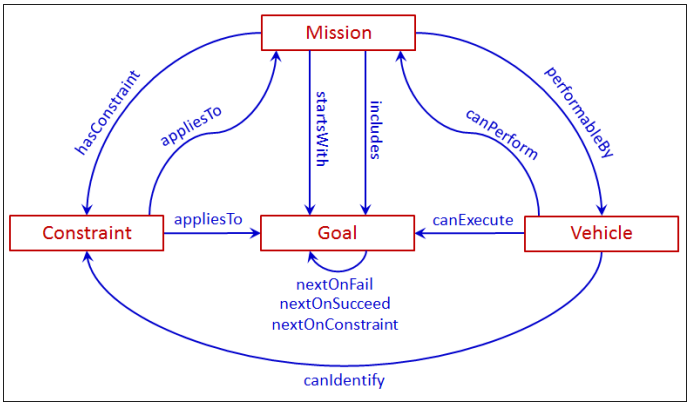

Fig. 9. A Mission Execution Ontology (MEO) is formally implemented using RDF/OWL describing the concepts, roles, and relationships associated with flow-diagram mission descriptions.

In addition to describing the "Mission", "Goal", "Constraint", and "Vehicle" concepts and how they relate to one another, the MEO allows the application of those concepts to real-world entities and the establishment of relationships among these entities. This means that the atomic entities to which the "Goal" and "Constraint" concepts are applied will be executable by specific target vehicles. The MEO can be applied to a snippet of vehicle-specific executable code by defining an OWL statement declaring its existence. Additional OWL statements can then declaratively apply concepts and establish relationships. Fig. 10 shows the mission of Fig. 5 expressed using OWL according to the MEO of Fig. 9, produced from RDF/OWL source using the Protégé Ontology Editor [32]. Detailed comparison of Fig. 5 with Fig. 10 (which was automatically generated) shows that the intended mission has been correctly encoded, is validated, and is executable by the target vehicle. Based on inspection of each figure, the human mission commander can confirm that all necessary ethical constraints have been applied in the correct contexts.

The ability to provide a full description of all goals and constraints using vehicle-executable code further strengthens the MEA construct. Specifically, not only is it impossible to define a mission for a particular vehicle without explicit "canExecute" relationships between the vehicle and all mission goals and "canIdentify" relationships between the vehicle and all mission constraints, but it is also impossible to assert these relationships without an appropriate vehicle-specific encoding of all mission goals and constraints.

Automated reasoning is an important tool for ensuring strategic-level mission validity before conducting exhaustive testing described in previous sections. If, for instance, a mission includes goals that are not executable by the target vehicle, a reasoner can quickly identify this shortcoming. Similarly a reasoner can detect mission flow-graph structural errors based on ontology rules, thereby precluding illogical loops, unreachable goals, and orphan goals without specified successors. A reasoner can also simplify mission definition by detecting implicit relationships that are not explicitly specified.

As described, a DL-based mission execution ontology defined in OWL ties MEA semantics discussed in previous sections to actual target vehicles. The ontology ensures not only structural validity of a mission-flow graph, but also its executability on the target vehicle. Thus, all of the requirements originally posed for assignment of human responsibility — that the mission is defined in a mathematically rigorous manner, that the mission is understandable by both the human operator and the target vehicle, and that the mission is comprised entirely of trusted vehicle behaviors — are captured in the human-approved mission orders and enforced by the strict semantics of the MEO. The ability to validate ethical correctness and completeness is an important new addition to cooperative mission definition and execution between humans and robots.

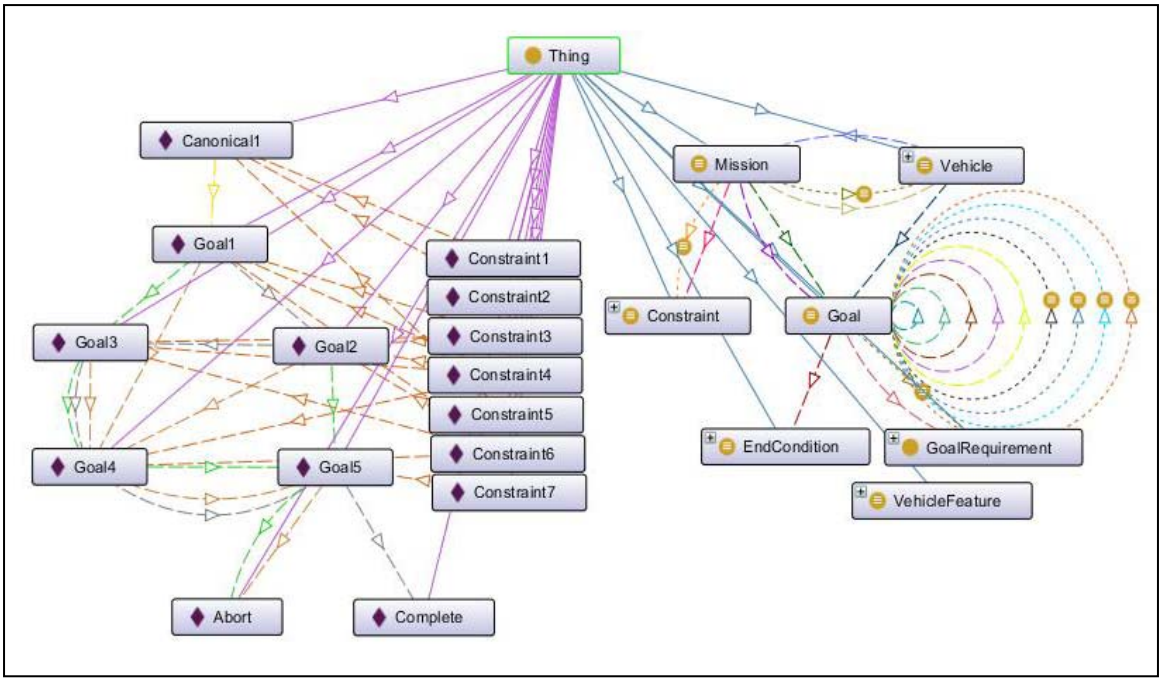

Fig. 10. OWL diagram of goals, relationships, and ethical constraints for Figure 5 mission as shown in Protégé Ontology Editor [32].

VII. SUMMARY AND CONCLUSIONS

Ethical operation of robotic systems requires human accountability. In both the legal and moral sense, this implies that human operators be in a position to understand, and therefore control, robot mission outcomes. This level of understanding can be achieved through the satisfaction of three requirements: operator understanding of high-level mission flow, mission descriptions understandable to both human operators and target vehicles, and mission descriptions consisting entirely of trusted behaviors and constraints.

Early NPS UUV missions were executed as a result of an inferencing process over a mission definition comprised of a set of rules and facts. This approach provided no means of proving the correctness of strategic-level missions comparable to the exhaustive testing of MEA flow graphs. For this type of system, errors in the mission axiom set can lead to unpredictable and potentially hazardous or self-defeating system execution behavior. Ultimately, this unpredictability precludes the formal assumption of responsibility or liability for robot missions.

Further, AI approaches in general almost invariably make use of easily confounded inferential reasoning or statistical pattern recognition. Applying such broad abstractions to the innumerable situations that can arise in the real world is inherently unpredictable, and also makes unrealistic any assumption of responsibility by human operators. It is therefore apparent that the abstract reasoning of general AI approaches is inappropriate, at least at the present time, for the highest level of robot mission definition and control.

Algorithms cannot replace human responsibility. Even so, a fully testable technology (such as that provided by the MEA and MEO formalisms) allows for the assignment of human accountability. Specifically, the MEA provides a mathematically rigorous mechanism for mission definition and execution as an exhaustively testable flow diagram. This approach ensures that accountable operators can fully understand all high-level task sequences before authorizing robot operations. The MEO employs DLs and Semantic Web technologies to provide strong assurances that MEA mission definitions are semantically correct and fully executable by specific target vehicles.

By applying the best strengths of human ethical responsibility, repeatable formal logic and directable unmanned systems together, these capabilities provide a practical framework for ethically grounded human supervision of unmanned systems.

ACKNOWLEDGMENT

The authors gratefully acknowledge the critical contributions of many dozens of colleagues and students, together with support from the NPS Consortium for Robotics and Unmanned Systems Education and Research (CRUSER; https://my.nps.edu/web/cruser) and the Office of the Secretary of Defense Joint Ground Robotics Enterprise (JGRE). The content of this paper reflects the opinions of the authors and not necessarily the position of the Naval Postgraduate School nor the sponsoring organizations.

REFERENCES

[1] I. Azimov, *I, Robot*, 2004 ed. New York, NY: Bantam Dell, 1950.

[2] R. Sparrow, "Just say "No" to drones," *Technology and Society Magazine,* vol. 31, no. 1, 2012, pp. 56-63.

[3] R. Arkin, *Governing Lethal Behavior in Autonomous Robots*. Boca Raton, FL: Taylor & Francis Group, 2009.

[4] L. A. McComas, *The Naval Officer's Guide*, 12th ed. Annapolis, MD: Naval Institute Press, 2011.

[5] "Vicarious Liability," in Merriam-Webster's *Dictionary of Law*, Springfield: Merriam-Webster, 2011.

[6] G.-J. Lokhorst and J. van den Hoven, "Responsibility for military robots," *Robot Ethics: The Ethical and Social Implications of Robotics*, P. Lin, K. Abney and G. A. Bekey, Eds. Cambridge, MA: MIT Press, 2012, pp. 145-156.

[7] G. T. Hammond, *The Mind of War: John Boyd and American Security*, Washington, DC: Smithsonian Institution Press, 2001.

[8] C. Dannegger, "Real-time autonomic automation," *Springer Handbook of Automation*, S. Y. Nof, Ed. New York, NY: Springer, 2009, pp. 381-404.

[9] D. P. Brutzman, D. T. Davis, G. R. Lucas, Jr., and R. B. McGhee, "Run-time ethics checking for autonomous unmanned vehicles: Developing a practical approach," *Proceedings of the 18th International Symposium on Unmanned Untethered Submersible Technology*, Portsmouth, NH, 2013.

[10] R. B. Byrnes, A. J. Healey, R. B. McGhee, M. L. Nelson, S. Kwak, and D. P. Brutzman, "The rational behavior software architecture for intelligent ships," *Naval Engineers Journal,* pp. 43-56, 1996.

[11] M. Ricard and S. Kolitz, "The ADEPT framework for intelligent autonomy," *Intelligent Systems for Aeronautics Workshop*, Brussels, Belgium, 2002.

[12] J. Albus, "Engineering intelligent systems," *Proceedings of the IEEE ISIC/CIRA/ISAS Joint Conference*, Gaithersburg, MD, 1998.

[13] R. Byrnes, *The Rational Behavior Model: A Multi-Paradigm, Tri-Level Software Architecture for the Control of Autonomous Vehicles*, Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA, 1993.

[14] M. Minsky, Computation: *Finite and Infinite Machine,* Englewood Cliffs, NJ: Prentice Hall, 1967.

[15] C. Petzold, *The Annotated Turing: A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine*, Indianapolis, IN: Wiley Publishing, 2008.

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

[17] R. B. McGhee, D. T. Davis, and D. P. Brutzman, "A universal multiphase mission execution automaton (MEA) with Prolog implementation for unmanned untethered vehicles," *Proceedings of the 17th International Symposium on Unmanned Untethered Submersible Technology*, Portsmouth, NY, 2011.

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

[19] D. P. Brutzman, R. B. McGhee, and D. T. Davis, "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*, Southampton, UK, 2012.

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

[21] D. T. Davis and D. Brutzman, "The autonomous unmanned vehicle workbench: mission planning, mission rehearsal, and mission replay tool for physics-based X3D visualization," *Proceedings of the 14th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, 2005.

[22] U.S. Army Training and Doctrine Command Analysis Center, COMBATXXI, September 2015. Available at http://www.trac.army.mil/COMBATXXI.pdf

[23] S. Posadas, *Stochastic Simulation of a Commander's Decision Cycle (SSIM CODE)*, Master's Thesis, Naval Postgraduate School, Monterey, 2001.

[24] C. Langley and C. Spenser, "The visual development of rule-based systems," *PC AI Magazine,* vol. 18, no. 3, pp. 29-36, 2005.

[25] S. S. Skiena, *The Algorithm Design Manual*, 2nd ed., London: Springer-Verlag, 1998, pp. 169-178.

[26] D. Brutzman, T. Healey, D. Marco, and B. McGhee, "The Phoenix autonomous underwater vehicle," in *AI-Based Mobile Robots*, Cambridge:, MIT/AAAI Press, 1998.

[27] D. P. Brutzman, *A Virtual World for an Autonomous Underwater Vehicle*, Naval Postgraduate School, Ph.D. Dissertation, Monterey, CA, 1994.

[28] M. Ortiz and M. Šimkus, "Reasoning and query answering in description logics," *Reasoning Web 2012*, Volume 7487 of Lecture Notes in Computer Science, Berlin Heidelberg, 2012.

[29] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American,* vol. 284, no. 5, pp. 34-43, May 2001.

[30] OWL Working Group, *OWL 2 Web Ontology Language Document Overview,* 2nd Ed., World Wide Web Consortium, 2012.

[31] I. Horrocks, "Ontologies and the Semantic Web," *Communications of the ACM,* vol. 51, no. 12,. 58-67, 2008.

[32] Stanford Center for Biomedical Informatics Research, Protégé Ontology Editor and Application Framework, Stanford University, 5 June 2016. [Online]. Available: http://protege.stanford.edu