Run-Time Ethics Checking for Autonomous Unmanned Vehicles
A New yet Classical Approach

CRUSER Project Report
review draft, 29 December 2012

Don Brutzman
Modeling, Virtual Environments, Simulation (MOVES)
Naval Postgraduate School (NPS), Monterey California USA
Topics

Motivation: human-taskable underwater robots

- Expressive power, semantics, syntax for a full solution
- Rational Behavior Model (RBM) 3-layer architecture: Strategic plans, Tactical operations, Execution control

Mission Planning, Execution with Ethical Constraints

- Declarative mission exemplar is helpful, general
- What does it mean to define, apply ethical constraints?
- Autonomous Vehicle Command Language (AVCL)
- AUV Workbench simulation results
- Topics for future student thesis work
Acknowledgements

Bob McGhee and Duane Davis contributed substantially to functionality of this approach.

Mike Bailey & Terry Norbraten made numerous software improvements to AUV Workbench.

George Lucas, Mark Dankel & Bradley Strawser gave useful feedback on ethics considerations.

CRUSER provided support for NPS faculty, staff labor.
Prior work: wargames and workshops include robot tasking, ethics issues

Future Unmanned Naval Systems (FUNS) wargame explored long-term autonomy issues for robot groups sharing reconnaissance tasks

CETMONS workshop explored robot ethics
  • Sponsored by USNA Stockdale Leadership Center
  • Can robots ever be given lethal weapons?
  • Both humans, robotic systems make errors in war
  • If robots make fewer errors than humans, then likely it becomes immoral to avoid using robots!
  • Slippery slope but includes potential benefits also
Motivating problem: ethics

How can robotic systems behave ethically?

- Important for military operations abroad
- Civil and scientific operations also coexist daily

Not necessarily about religion or morality

- Law of armed conflict internationally recognized; often captured as Rules of Engagement (ROE) guidance for specific national or coalition forces
- Military readiness must be prepared for combat

Legal basis for civil activity is often well defined

- Also measurable: are you vulnerable to lawsuit?!
Motivation: human-taskable maritime robots

We’re looking to near-term future when UUVs can offload work and augment human activity.

Various excellent robot solutions must remain compatible with human concepts and tasking.

- Necessary for mission planning and justification.
- Otherwise the robots are simply not autonomous.

Extensive/exhaustive prelaunch testing is critical.

- For mission confidence and in-water reliability.

This work is building an extendable architecture...

... for continued efforts bridging human, robot logic.
Our approach

Define missions that include ethical constraints without relying on artificial intelligence (AI) or obscure abstractions for appropriate behavior

- No embedded homunculus or ethicist engine

Design robot missions in way that can be adapted to a variety of disparate robot paradigms

- Generally adaptable to tasking of diverse systems

Build on patterns that work well for human groups cooperating on difficult, dangerous tasks

Otherwise still need human at end of comms tether
Key insight #1

Humans in military units are able to deal with these challenges without ethical quandaries.

Careful definitions are provided for:

- Tasks, missions, rules of engagement
- Ethical constraints

These allow both measured and rapid response, independently and cooperatively:

- Commanders do not deploy illegal, immoral weapons
- Unmanned systems must also pass similar scrutiny, otherwise commanders cannot utilize them
Rational Behavior Model (RBM)
3-layer robot architecture

Strategic
• Declarative planning of goals while avoiding obstacles and observing constraints

Tactical
• Operational control of navigation, tactical and mission tasks. Sensor employment.

Execution
• Low-level control tasks. Open-loop, closed-loop commands for propulsors and effectors.

Twenty+ years of well-documented, progressive work
Turing machine

Turing machine (TM)

- consists of a *finite state machine* (FSM)
- augmented by an *external agent* in the form of a *potentially infinite* memory
- realized as *tape* of an “incremental tape recorder”

- Often referenced but seldom used

  - Extensive theoretical development has shown that Universal TM has greatest computational power
  - Clumsy to program, infrequently used
  - Nevertheless, appealing basis for theoretical design since it maps to *general theory of any computation*
Prior work: technical reports and papers on Mission Execution Automaton (MEA)

We adapted Universal Turing Machine (UTM) architecture to create a universal multiphase human-interactive MEA

- Express fundamental logic of mission planning
- Corresponds to RBM Strategic level implementation
- Goal: demonstrate computational generality

Implemented using Prolog language for easy backtracking during planning

- Prolog dialect: Allegro Common Lisp
- Interactive, text based
- Human provides external-world sensing responses
Goal-based Mission Example

- Simple yet general mission goals, decision logic
- Common approach, adaptable to other vehicles
- Extendable and refinable mission tasking
Mission execution rule set: simply loop through goal tasks
Example Goal-based Mission Definition

**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 **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, **rendezvous** with UUV-2. Upon rendezvous success or failure, **transit** to recovery position to complete the mission.
Goal-based Mission Example

Mission execution is independent of robot software implementation
Executable mission rule set: loop through phases

(<-- (execute_mission)
  (initialize_mission)
  (repeat) (execute_current_phase) (done) !)
(<-- (initialize_mission)
  (abolish current_phase 1)
  (asserta ((current_phase 1)))
(<-- (execute_current_phase)
  (current_phase ?x) (execute_phase ?x) !)
(<-- (done) (current_phase 'mission_complete))
(<-- (done) (current_phase 'mission_abort))
Executable mission phases for this example added to Prolog rule set

; Mission specification

(\-\- (execute phase 1) (command "Search Area A") (phase_completed 1))
(\-\- (phase_completed 1) (ask "Search successful" ?A) (affirmative ?A) (change_phase 1 2))
(\-\- (phase_completed 1) (change_phase 1 3))

(\-\- (execute phase 2) (command "Sample environment") (phase_completed 2))
(\-\- (phase_completed 2) (ask "Sample obtained" ?A) (affirmative ?A) (change_phase 2 3))
(\-\- (phase_completed 2) (change_phase 2 5))

(\-\- (execute phase 3) (command "Search Area B") (phase_completed 3))
(\-\- (phase_completed 3) (ask "Search successful" ?A) (change_phase 3 4))

(\-\- (execute phase 4) (command "Rendezvous UUV2") (phase_completed 4))
(\-\- (phase_completed 4) (ask "Rendezvous successful" ?A) (change_phase 4 5))

(\-\- (execute phase 5) (command "Return to base") (phase_completed 5))
(\-\- (phase_completed 5) (ask "At base" ?A) (affirmative ?A)
   (change_phase 5 'mission_complete) (report "Mission succeeded"))
(\-\- (phase_completed 5) (change_phase 5 'mission_abort) (report "Mission failed"))
Running mission example #1

CG-USER(1): (?- (execute_mission))
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.

User enters responses to mission execution automaton (MEA) queries, thereby providing environmental responses and testing mission logic.

These correspond to ethics checks measured by robot, or supervised by a human.

Background
Running mission examples # 2, 3

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

CG-USER(3): (?- (execute_mission))
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.

Human operators can logically test various mission alternatives to gain confidence

Simulation can also run through every option exhaustively to further confirm correctness
Adding ethical constraints to mission requirements

Following the leader:
how do human teams accomplish tasks ethically?

The same rules need to apply to unmanned systems.
Example ethical constraints, civil

Safe navigation, follow pertinent rules of road
Satisfactory navigational accuracy (GPS etc.)
Have received timely clearance to enter a
specific geographic area for given time period
  • Also vertical clearance for underwater depth zone
    or airborne altitude zone
Sufficient vehicle health, power, safety status
Meet communication requirements for tasking
  • Identity beacon, transponder, AIS tracking, etc.
  • Recording and reporting on situational data, etc.
Example ethical constraints, military

Meet all relevant, international civil requirements
Identification friend foe (IFF), blue-force tracking
  • friendly/hostile/neutral/unknown
Prior determination of contact’s hostile intent
  • Robot option to warn without fear of self protection
ROE use of deadly force, weapons releasability
  • Brevity codes: weapons safe, hold, tight, free
  • Confirmation and permission requirements
After-action reporting, damage assessment

Et cetera, et cetera...
# Civil ethical constraint support in AVCL, AUV Workbench

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

<table>
<thead>
<tr>
<th>Military ethical constraints</th>
<th>Define</th>
<th>Test</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meet all relevant, international civil requirements</td>
<td>partial</td>
<td>partial</td>
<td>See above</td>
</tr>
<tr>
<td>Mission tasking</td>
<td>√</td>
<td>partial</td>
<td>AVCL goals</td>
</tr>
<tr>
<td>Contact identification, tracking signatures</td>
<td></td>
<td>partial</td>
<td>Available in C2 systems</td>
</tr>
<tr>
<td>Identification friend foe (IFF), blue-force tracking (friendly/hostile/neutral/unknown/etc.)</td>
<td></td>
<td></td>
<td>Available in C2 data models</td>
</tr>
<tr>
<td>Robot option to warn without fear of self protection</td>
<td></td>
<td></td>
<td>Implementable via messaging</td>
</tr>
<tr>
<td>Determination of contact’s hostile intent</td>
<td></td>
<td></td>
<td>Available in C2 data models, dissertation work in progress</td>
</tr>
<tr>
<td>Confirmation and permission requirements</td>
<td></td>
<td></td>
<td>Implementable via messaging</td>
</tr>
<tr>
<td>ROE use of deadly force, weapons releasability using brevity codes: weapons safe, hold, tight, free</td>
<td>partial</td>
<td>partial</td>
<td>Requires weapons model</td>
</tr>
<tr>
<td>Proportional weapons response</td>
<td></td>
<td></td>
<td>Requires weapons and threat models</td>
</tr>
<tr>
<td>After-action reporting</td>
<td>partial</td>
<td>partial</td>
<td>AVCL goals</td>
</tr>
<tr>
<td>Damage assessment</td>
<td></td>
<td></td>
<td>Requires models of interest</td>
</tr>
</tbody>
</table>
Key insight #2

Ethical behaviors don’t define the mission plan.

rather

Ethical constraints inform the mission plan.
Goal-based Mission Example, with constraints

Strategic Level

Phase 1: Search Area A

- MEA Start
  - Phase 2: Sample Environment in Area A
    - Sample Environment
    - Phase Controller
      - Area Definition
      - Sampling Requirements

- Operational and Ethical Constraints Checking
  - Controller Parameters
  - Sensor Settings, Responses

- System State and Sensor Data
- Vehicle Commands
- Real-time orders
- Execution Level
  - handles real-time control and response

Tactical Level

Phase 2: Sample Environment in Area A

- Queries
  - Responses: Succeed or Fail

Phase 3: Search Area B

- Constraints

Phase 4: Rendezvous with UUV-2 in Area C

- Constraints

Phase 5: Transit Return to Base

- Fail
- Succeed

Mission Abort
Mission Complete

Surface
Recovery position

Mission Abort
Mission Complete

Fail
Succeed
Succeed
Fail
Succeed
Fail
Succeed
Fail
Succeed

Mission Abort
Mission Complete
Example goal-planning logic: Search

Planner utilized to generate a script capable of accomplishing a Search goal in an AVCL declarative mission.

Upper Level State Machine of the Search Planner:

```
+---------------------------+               +---------------------------+   
| TRANSIT_AND_ALERT        | start, alert = true | COMPLETE                |   
| alertEnded               | Execute            | reportStatusChange      |   
|                          | V                  | targetFound             |   
| targetFound = true       | EXECUTE_SEARCH     | reportMade              |   
| reportStatusChange = false | V          | reportStatusChange = true |   
| singleTarget = false     | REPORT_FOUND       | targetFound             |   
|                          | reportMade, singleTarget = true | V |   
+---------------------------+                       +-------------------------------+
```

- `attributes`: Specifies whether the search datum is a point (center of the goal area) or the entire area.
- `requiredPD`: Required probability of detection of the search.
- `singleTarget`: Set to true if the search is only successfully complete upon finding a single instance of a search target. False means that the search continues in order to locate multiple instances, and search is successful upon completion.
- `timeStamp`: Time value in seconds.
- `description`: Descriptive text explaining the intended effect of this element.
- `id`: Unique identifier for this element.
Example goal definition

**Goal Element Type**

**Name/ID**

**Description**

- Conduct a simple box search

**Datum Type**

- Point

**Single Target**

**Required PD**

0.8

**Search Targets**

- name/id
- target
- description

**Search**

**Operation Area**

- Northwest Corner
  - x: 350.0
  - y: 300.0
  - width: 100.0
  - height: 100.0
  - orientation: 0.0

**Timing Type**

**Reporting Criteria**

<table>
<thead>
<tr>
<th>Period</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>goal completion</td>
<td></td>
</tr>
</tbody>
</table>
# AVCL mission goals support

<table>
<thead>
<tr>
<th>AVCL mission goals</th>
<th>Define</th>
<th>Test</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack</td>
<td>partial</td>
<td></td>
<td>To conduct a type of offensive action characterized by employment of firepower and maneuver to close with and destroy an enemy.</td>
</tr>
<tr>
<td>Decontaminate</td>
<td>✓</td>
<td></td>
<td>To provide purification making an area safe by absorbing, destroying, neutralizing, making harmless, or removing chemical, biological, or nuclear contamination.</td>
</tr>
<tr>
<td>Demolish</td>
<td>✓</td>
<td></td>
<td>To destroy structures, facilities, or material by any available means.</td>
</tr>
<tr>
<td>IlluminateArea</td>
<td>✓</td>
<td></td>
<td>To provide locale lighting by searchlight or pyrotechnics.</td>
</tr>
<tr>
<td>Jam</td>
<td>✓</td>
<td></td>
<td>To deliberately radiate, re-radiate or reflect electromagnetic energy with the object of impairing the use of electronic devices or systems.</td>
</tr>
<tr>
<td>MarkTarget</td>
<td>✓</td>
<td></td>
<td>To make visible (by the use of light, infrared, laser, smoke, etc.) of an object in order to allow its identification by another object.</td>
</tr>
<tr>
<td>MonitorTransmissions</td>
<td>✓</td>
<td></td>
<td>To conduct electronic warfare support operations with a view to searching, locating, recording and analyzing radiated electromagnetic energy.</td>
</tr>
<tr>
<td>Patrol</td>
<td>✓</td>
<td>✓</td>
<td>To gather information or carry out a security mission.</td>
</tr>
<tr>
<td>Rendezvous</td>
<td>✓</td>
<td>partial</td>
<td>Achieve a meeting at a specified time and place.</td>
</tr>
<tr>
<td>Reposition</td>
<td>✓</td>
<td></td>
<td>To change position from one location to another.</td>
</tr>
<tr>
<td>SampleEnvironment</td>
<td>partial</td>
<td></td>
<td>Collect environmental samples for testing for chemical compounds, biological creatures, or nuclear hazards.</td>
</tr>
<tr>
<td>Search</td>
<td>✓</td>
<td>✓</td>
<td>To look for lost or unlocated objects or persons.</td>
</tr>
</tbody>
</table>
Theory versus practice

In theory: theory and practice are the same.

In practice: they’re not.

- Yogi Berra

Current tests: simulations
Challenge: broad implementation

Can we

• Define mission goals readable by humans + robots
• Produce actionable tasking for different AUVs
• Produce software examples that can run properly in simulation and in robots

Can we also

• Define goal constraints ethically and measurably

Initial tests successful
Autonomous Vehicle Command Language (AVCL)

AVCL is a command and control language for humans running autonomous unmanned vehicles

- Close correspondence to human naval terminology
- Common XML representations for mission scripts, agenda plans, and post-mission recorded telemetry

Operators can utilize a single archivable and validatable format for robot tasking, results

- directly convertible to and from a wide variety of different robot command languages

https://savage.nps.edu/Savage/AuvWorkbench/AVCL/AVCL.html
AVCL vocabulary: AllCommandsUuv.xml

List, unit test all execution, tactical AVCL commands
- Simple unit testing of workbench software
- Confirm AUV Workbench interface, XML validation
- Mission unit tests for comprehensive coverage likely to follow

Also for other robotics platforms
- AllCommandsUsv.xml  AllCommandsUav.xml
- AllCommandsUgv.xml
- Each provides a vocabulary list showing the current tactical repertoire (rather than runnable missions)
0. Meta Command [TODO] (one command throws an error: Bad command position found
1. Composite Waypoint USV Search: sector pattern, expanding square or creeping line
2. Follow Beacon Tracking mode
3. GpsFix enabled Go to surface, get navigational fix
4. Help Help text
5. Hover Hover at current location
6. Hover (20.0, 0.0) Hover at relative location away from current position
7. Hover [1.0, 0.0] Hover at given location
8. Loiter USV Loiter at current position
9. Loiter USV (20.0, 20.0) Loiter at relative location away from current position
10. Loiter USV [10.0, 20.0] Loiter at specific location
11. Make Altitude 0.0 m Height above ocean floor, bottom-following mode
12. Make Depth 0.0 m Ordered depth
13. Make Heading 0.0° Ordered course
14. Make Knots 0.0 knots Ordered speed
15. Make Speed 0.0 m/s Ordered speed
16. Mission Script Can load another AVCL script
17. Mission Script Inline inserts AVCL commands from a script
18. Move Lateral 0.0 m/s Use thrusters, move sideways
19. Move Rotate 0.0 °/s Use thrusters, rotate in place
20. Mission enabled Prevents (or triggers) simulation speedup
21. Recover Perform a recovery maneuver with a docking station or support ship
22. Set Position USV (0.0, 0.0) Apply fix position to reset previous estimated position
23. Set Power Set forward thrust as percentage of full authority
24. Set Rudder 0.0% Set angle as percentage of full authority
25. Set Standoff 2.5 m Standoff range for successful proximity to waypoint or loiter point
26. Set Thrustr Settings Lateral thruster strength as percentage of full power
27. Set Thrustr Settings Vertical thruster strength as percentage of full power
28. Set Time 10000.0 seconds (27:46:40) Reset system clock
29. Set Time Step 0.1 seconds 10 Hz sense-decide-act control-loop refresh rate
30. Take Station Use scanning sonar to maintain position relative to tracked object
31. Trace enabled Enable or disable verbose run-time trace messages and logging
32. Wait 0.0 seconds (0:00:00) Time-delay wait interval until continuing with next command
33. Wait Until Time 0.0 seconds (0:00:00) Wait until specified time before continuing
34. Waypoint USV Waypoint at current location
35. Waypoint USV (0.0, 0.0) Waypoint at fixed location
36. Waypoint USV (0.0, 5.0) Waypoint at relative location
37. Waypoint USV (5.0, 0.0) Waypoint at absolute location
38. Quit Mission complete
39. Mission complete
40. Quit Mission complete
0. **Composite Waypoint UAV**  
Search: sector pattern, expanding square or creeping line

1. **Help**  
Display help text (when in user-interactive mode)

2. **Loiter UAV**  
Loiter at current position

3. **Loiter UAV [0.0, 0.0, 1000.0]**  
Loiter near fixed location

4. **Loiter UAV (20.0, 20.0)**  
Loiter near point relative to current ordered position

5. **Make Altitude AGL 1000.0m**  
Above Ground Level

6. **Make Altitude MSL 2000.0m**  
Above Mean Sea Level

7. **Make Climb Rate -5.0 m/s**  
Ordered ascent/descent rate

8. **Make Heading 90.0°**  
Ordered course East

9. **Make Knots 250.0 knots**  
Ordered air speed

10. **Make Speed 100.0 m/s**  
Ordered air speed

11. **Make Turn Rate 5.0 °/s**  
clockwise to right since adding degrees to heading

12. **Meta Command [Note]**  
[Can include vehicle-specific commands, notes, comments]

13. **Mission Script**  
Can load another AVCL script

14. **Mission Script**  
Inline inserts AVCL commands from a script

15. **Realtime enabled**  
Prohibit (or trigger) simulation speedup

16. **Meta Command [TODO]**  
[SendMessage is not yet working]

17. **Set Aileron -25.0%**  
set angle as percentage of full authority

18. **Set Elevator 50.0%**  
set angle as percentage of full authority

19. **Set Position UAV [0.0, 0.0]**  
Apply fix position to reset previous estimated position

20. **Set Rudder 0.0%**  
Set angle as percentage of full authority

21. **Set Standoff 250.0m**  
Standoff range for successful proximity to waypoint or loiter

22. **Set Time 1000.0 seconds [0:16:40]**  
Reset system clock

23. **Set Time Step 0.05 seconds [20 Hz sense-decide-act control-loops refresh rate**

24. **Trace enabled**  
Enable or disable verbose run-time trace messages and logging

25. **Wait 1000.0 seconds [0:16:40]**  
Time-delay wait interval until continuing with next

26. **Wait Until Time 2000.0 seconds [0:33:20]**  
Wait until specified time before continuing

27. **Waypoint**  
Waypoint at current position

28. **Waypoint [0.0, 0.0]**  
Waypoint at fixed position

29. **Waypoint (0.0, 1000.0)**  
Waypoint position relative to current ordered position

30. **Quit**  
Mission complete, shut down

31. **Meta Command [TODO isolate and fix AVCL commands causing error]**  
[Bad command position and Note]
Example mission, as pseudo-code XML

```xml
<?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>
```

XML is readable by human or robot, captures logic of canonical mission.
<MissionPreparation>
  <UnitsOfMeasure time="seconds" mass="kilograms" angle="degrees" distance="meters"/>
  <GeoOrigin longitude="-121.88500213623047" latitude="36.606998443603516"/>
</MissionPreparation>

<AgendaMission>
  <LaunchPositionAH description="Start point">
    <XYPosition time="0.0" y="6350.0" x="12300.0"/>
  </LaunchPositionAH>
  <RecoveryPosition description="Finish point">
    <XYPosition y="6500.0" x="12300.0"/>
  </RecoveryPosition>
</AgendaMission>

<GoalList>
  <Goals description="search operating area A" alert="false" nextOnFail="goalC" nextOnSucceed="goalB" id="goalA">
    <Search singleTarget="false" requiredPD="0.8" dataType="area"/>
  </Goals>
  <OperatingArea>
    <Rectangle>
      <XYPosition y="6425.0" x="12625.0"/>
    </Rectangle>
    <Width description="" value="50.0"/>
    <Height description="" value="150.0"/>
  </OperatingArea>
</GoalList>
AUV Workbench

Autonomous Unmanned Vehicle Workbench supports underwater, surface and air vehicles
- physics-based mission rehearsal
- real-time task-level control of robot missions, and
- replay of recorded results
- Industry-friendly open-source license, Sourceforge
- Basis: RBM 3-level architecture, AVCL commands

Used to rehearse strategic-level MEAMission.xml
- https://savage.nps.edu/AuvWorkbench
4 example missions, UUV and USV
MeaMission.xml simulation preview

Search planning produces waypoints

All searches complete

Launch

Recovery

Rendezvous not supported

All events logged

Map View  2D View  X3D View  Telemetry Plots
MeaMission.xml simulation 0, launch
MeaMission.xml simulation 1, transit
Mission.xml simulation 2, search
MeaMission.xml simulation 3, transit
MeaMission.xml simulation 4, sample
MeaMission.xml simulation 5, transit
MeaMission.xml simulation 6, search
MeaMission.xml simulation 7, rdu fail
MeaMission.xml simulation 8, transit
MeaMission.xml simulation 9, recovery
MeaMission.xml simulation 10, complete.
Mission definition observations

MEA formalism assumes human generation of mission orders

- Example expressed using AVCL
- Can also be expressed in other languages such as Java, Prolog, CLIPS rule sets, C++, etc.
- As shown in multiple RBM theses, dissertations

**Key insight #3.** We continue our exploration of producing general mission orders that are:

- Understandable by (legally culpable) humans
- Reliably, safely executable by robots
Generality to other robots

High-level mission planners tend to be dissimilar
- Much of AI can be posed as search algorithms
- Various RBM strategic-level implementations possible

Strong match between RBM tactical, execution AVCL vocabularies and many other robots
- We’ve written many AVCL importers/exporters

Generality principle still holds, however:
- Outputs of higher-level mission planners can be expressed as AVCL tactical commands!
- Perhaps unsurprising since we’ve designed AVCL command sets using human procedures at sea
Looking ahead
Thoughts on autonomous ethics

Ethics: everything old is new again!

- Possibly unexplored or unconsidered territory?
- Significant history of experience: mine warfare

Assume success... then what?

- Even if perfectly executable, proper robot logic is not useful unless it is a directly extension of warfighter logic
- Rules of Engagement, Concepts of Operations, Doctrine, Tactics, etc. must be expressible in equivalent terms in order to be effective, usable

Civilian equivalents: Rules of Road, safety, etc.
Multi-layer robot architectures... onward and upward!

3 layers defined in Rational Behavior Model (RBM) robot architecture are fairly common in robotics
- Strategic: planning, goal evaluation, strategy
- Tactical: navigation, operations, mission conduct
- Execution: low-level control of systems

Autonomous Vehicle Command Language (AVCL)
- Maps to various commands, goals, robot dialects

Possible 4th layer above: Rules of Engagement?
- provide ethical constraints and boundary conditions on robot strategic planning and tactics
- Must work cooperatively, satisfactorily with humans
Candidate student theses

Build, test both operations order and ROEs for unmanned systems supporting fleet assets
Explore missions that test ethical conundrums related to robot lethality
Test missions afloat in Monterey Bay or at TNT exercises Camp Roberts, Fort Hunter Liggett
  - mapping AVCL, AUV Workbench to NPS robots
Numerous further topics possible, let’s discuss
  - Your topic here, perhaps...
Software development TODO

Keep building realistic models for AUV Workbench
- Have added our first ROV model + joystick control
- Mission reports, KML, X3D graphics visualization
- Control and dynamics coefficients, sensor models
- Emphasis on education and undergraduate students

Develop larger repertoire of missions
- XSLT stylesheet to convert AVCL strategic mission into usable source code (Prolog, Java, C++, etc.)
- Let virtual world provide sensor responses

Integrate standard Prolog within AUV Workbench
- Gnu open-source Java-based Gnu Prolog
Discussion

“There are more things in heaven and earth than are dreamt of in our philosophy, Horatio.”

William Shakespeare, *Hamlet*

Collaboration, contributions, insight and dialog are always welcome
Contact

Don Brutzman, Ph.D.

brutzman@nps.navy.mil
brutzman@nps.navy.smil.mil
http://faculty.nps.edu/brutzman

Code USW/Br, Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.2149 work
1.831.402.4809 cell
Contact

Bob McGhee, Ph.D.
robertbmcghee@gmail.com

MOVES Institute
Naval Postgraduate School
Monterey California 93943-5000 USA
Contact

CDR Duane Davis USN (Ret.) Ph.D.

dtdavi1@nps.edu

Code CS/Da
Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.7980 work