

Ethical Mission Definition and Execution for Maritime Robotic Vehicles

A Practical Approach

CRUSER TechCon

11 July 2016

Don Brutzman, Duane Davis, Curt Blais and Bob McGhee
Naval Postgraduate School (NPS), Monterey California USA



Motivating challenge: 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 teams of national or coalition forces

Legal basis under challenge: are robots unethical?

- Military readiness must be prepared for combat
- Opposing forces don't read the same memos...



Motivation: human-taskable maritime robots

We look to near-term future when UMVs can offload force projection, reduce danger to fleet
Variety of emerging 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



Practical approach

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

- No embedded homunculus or abstract 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

- Human accountability remains central
- Otherwise still need human at end of remote tether



Key insight #1

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

Careful definitions are provided for

- Tasks, missions, objectives, coordinated operations
- Ethical constraints and rules of engagement

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



Enabling factor: maritime environment

Major international controversy unfolding:
drone use for conduct of reduced-risk warfare

- Many factors involved: technical, political, social
- Remote human “control” is highly questionable
- Complex, confounded environments

Maritime environment is much less ambiguous

- Fewer IFFN issues, identification friend foe neutral
- Presence of bad actors usually confirmable
- Law of Sea, Laws of Armed Conflict, etc.



Goal-based Mission Example

- Simple yet general mission goals, decision logic
- Common approach, adaptable to other vehicles
- Extendable and refinable mission tasking



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.

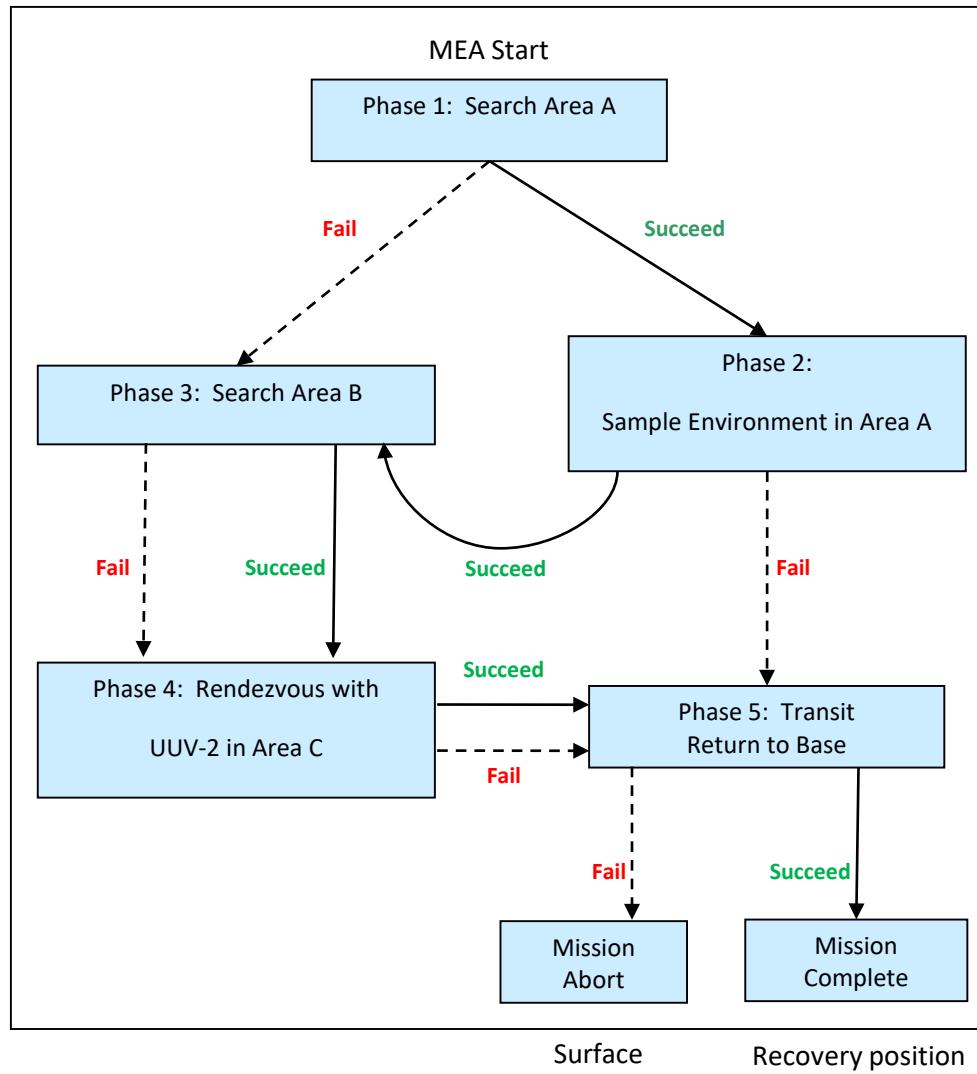
Goal 4. Proceed to Area C, **rendezvous** with UUV-2.

Upon rendezvous success or failure, **transit** to recovery position to complete the mission.



Goal-based Mission Example

Strategic Level



Robot mission conduct
can be independent of
software implementation

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.



Key insight #2

Ethical behaviors don't define the mission plan.

rather

Ethical constraints inform the mission plan.



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.



Civil ethical constraint support in AVCL, AUV Workbench

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			

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

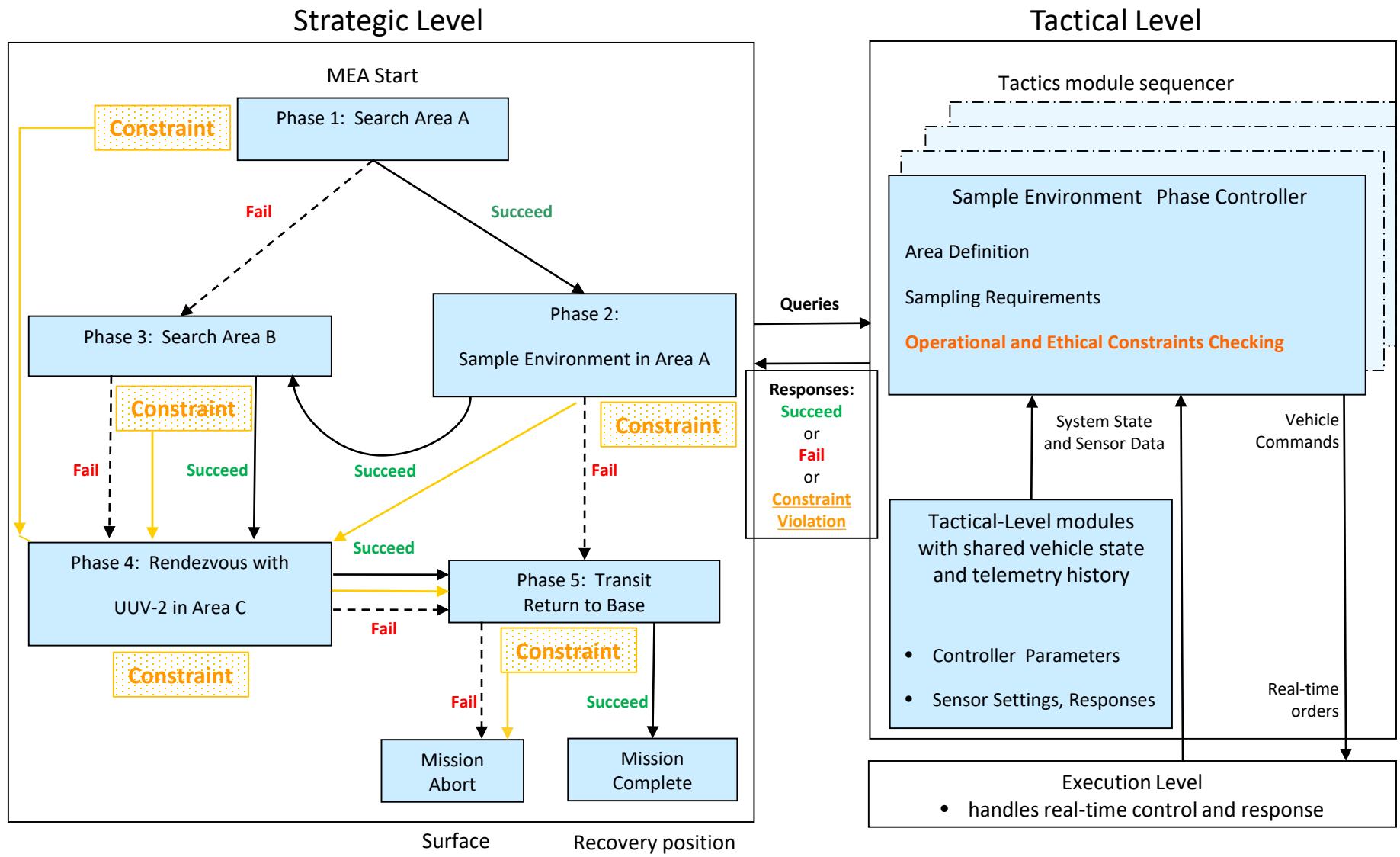
Et cetera, et cetera...



Military ethical constraint support in AVCL, AUV Workbench

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

Goal-based Mission Example, with constraints



Constraints applied to sample mission

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

Constraint 2: All safety equipment must be fully functional. *

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. *

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.



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

Yes

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>

Example mission, as pseudo-code 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>
```

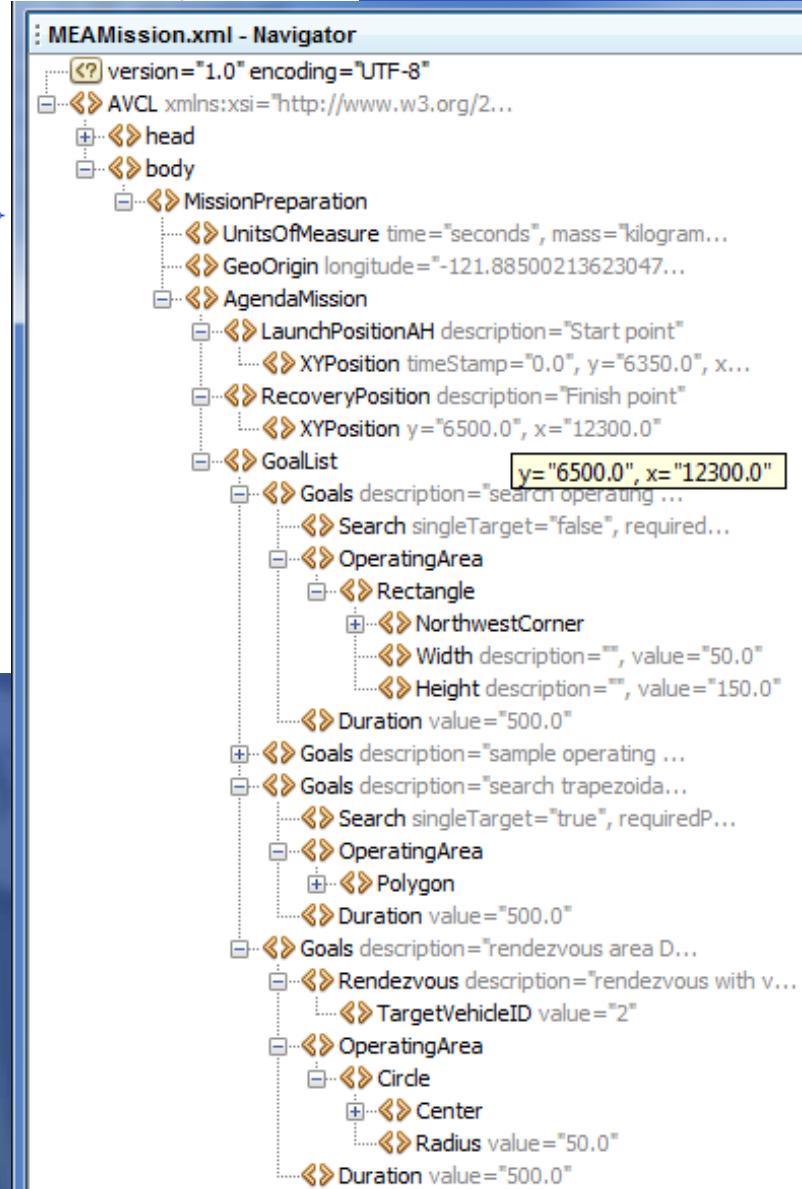
XML is readable by human or robot,
captures logic of canonical mission



```

<MissionPreparation>
  <UnitsOfMeasure time="seconds" mass="kilograms" angle="degrees" distance="meters" />
  <GeoOrigin longitude="-121.88500213623047" latitude="36.606998443603516" />
<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>
  </GoalList>

```



Corresponding example: MEAMission.xml using AVCL xml



AVCL mission goals support

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

Example
open-source
implementation

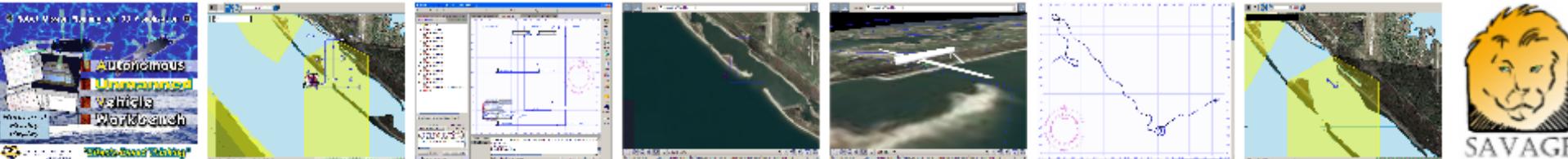
AUV Workbench

Autonomous Unmanned Vehicle Workbench
supports underwater, surface and air vehicles

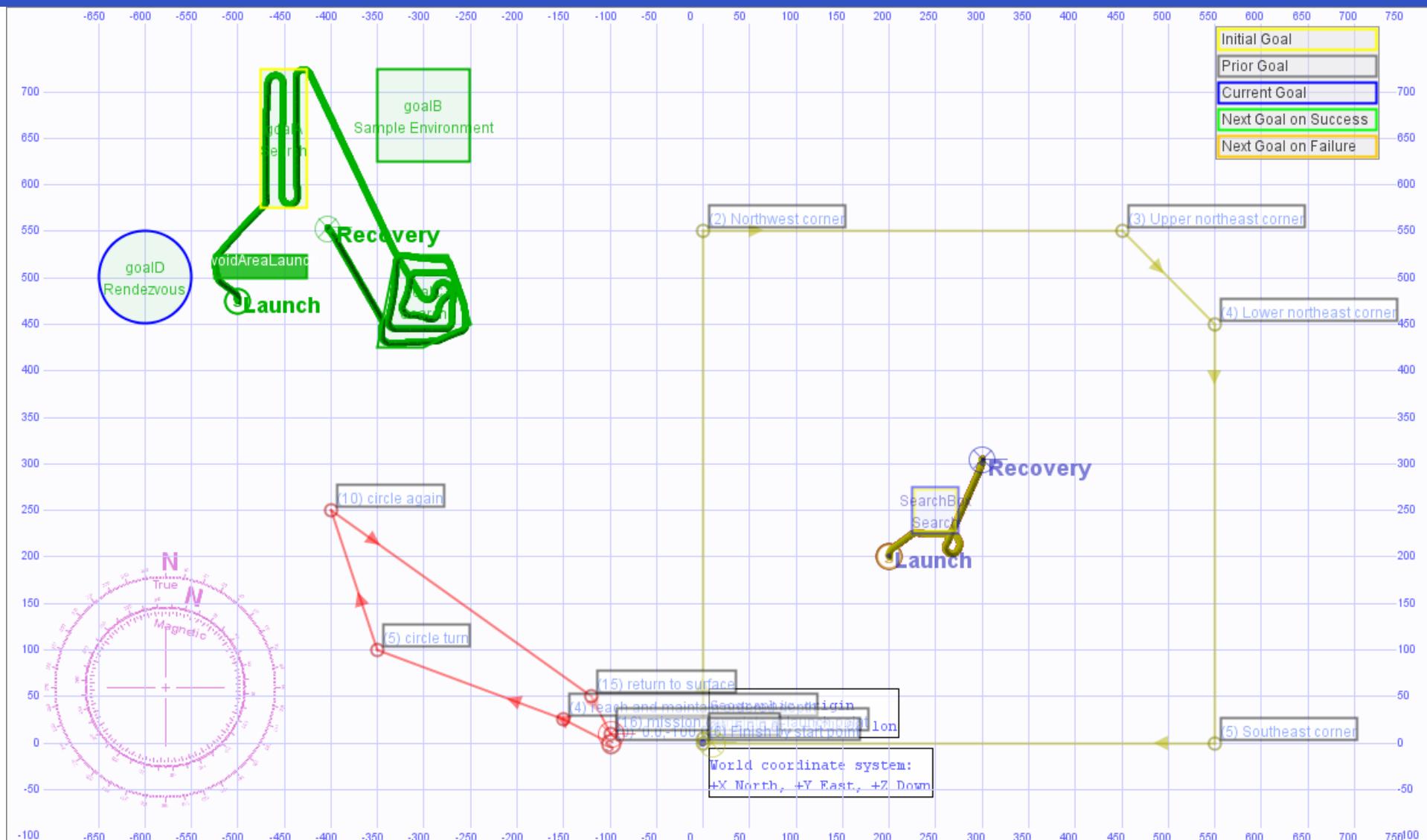
- physics-based mission rehearsal
- real-time task-level control of robot missions, and
- replay of recorded results
- Industry-friendly open-source license, Sourceforge
- Basis: RBM 3-level architecture, AVCL commands

Used to rehearse strategic-level MEAMission.xml

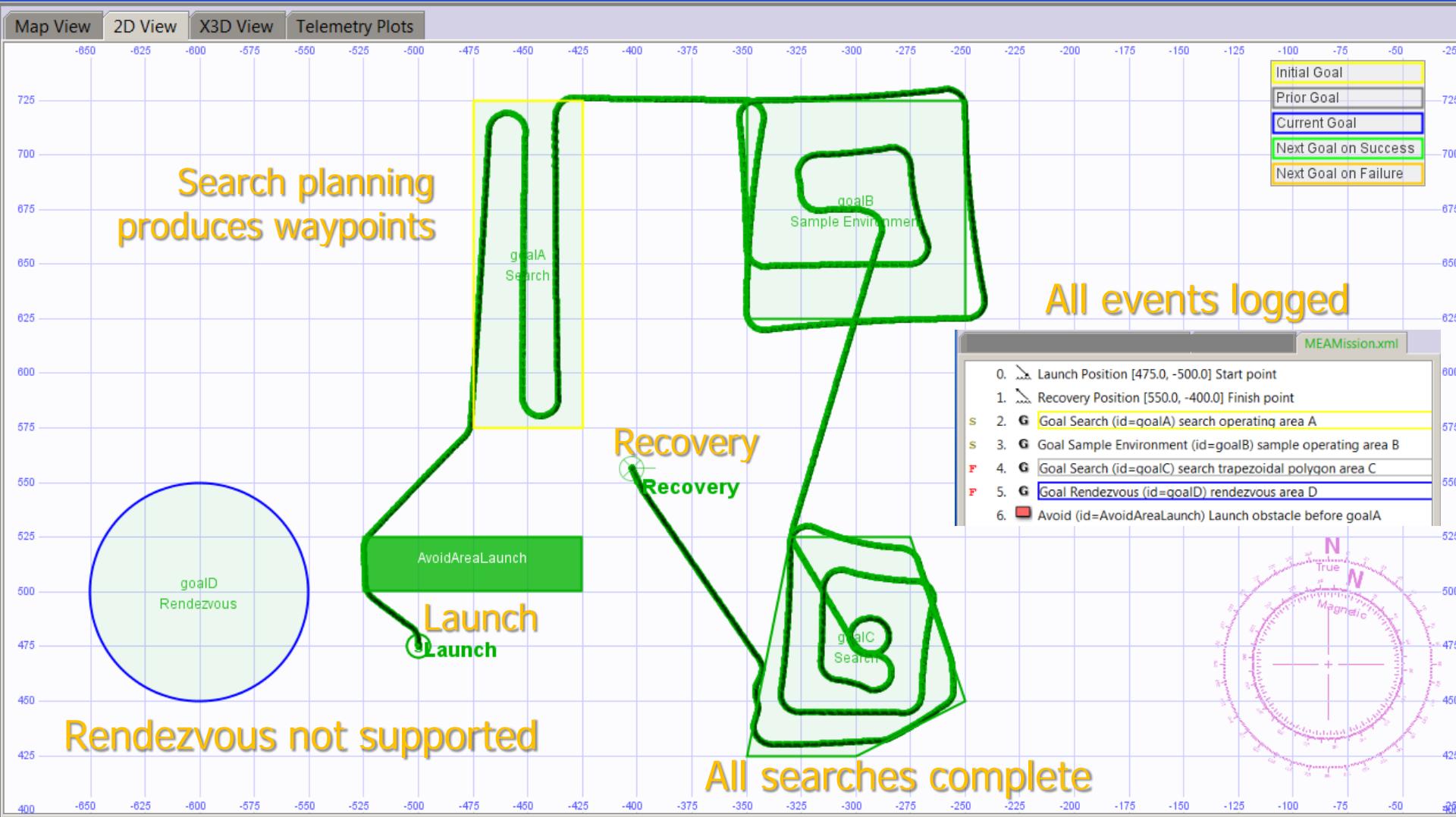
- <https://savage.nps.edu/AuvWorkbench>



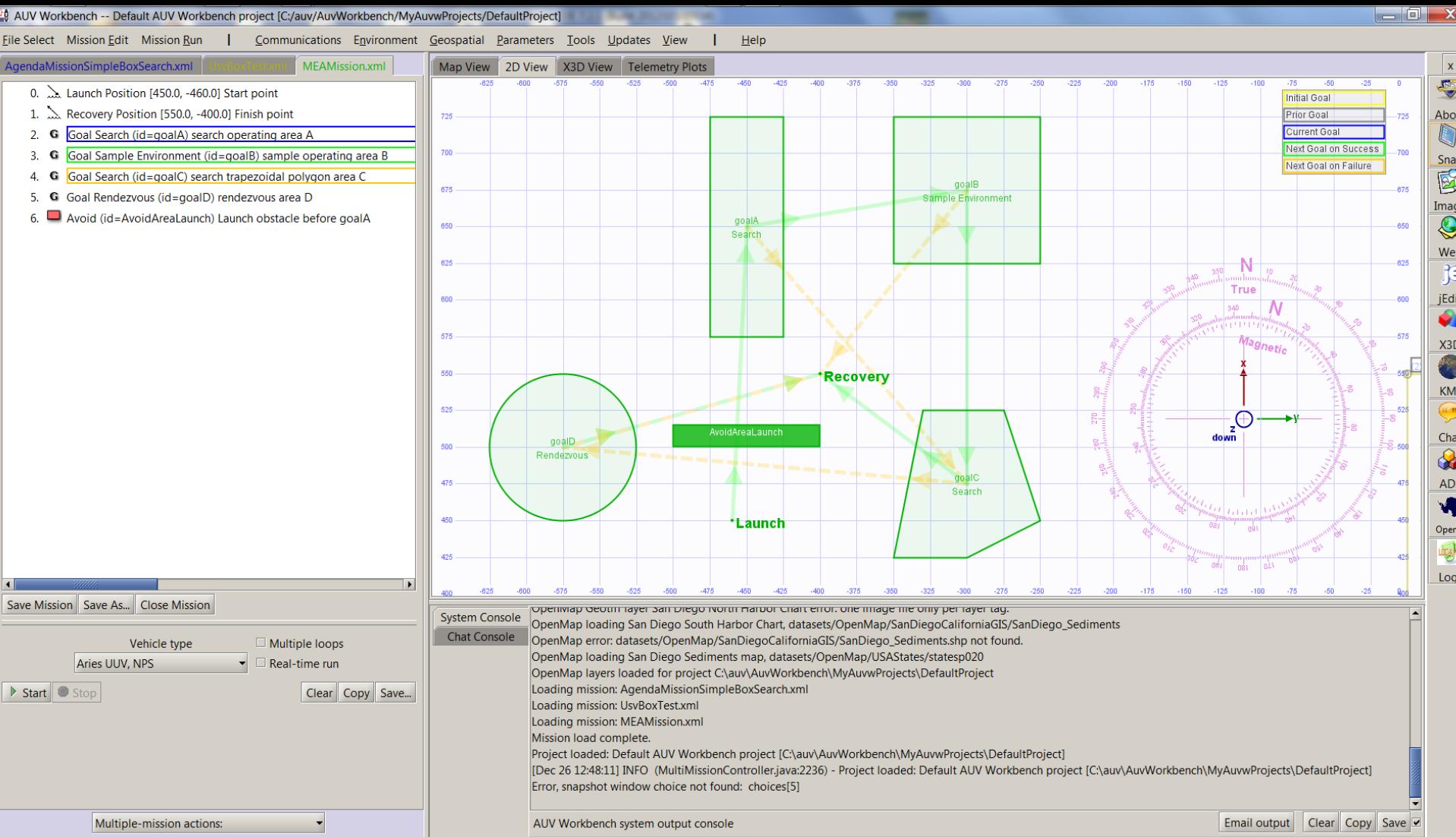
4 example missions, UUV and USV



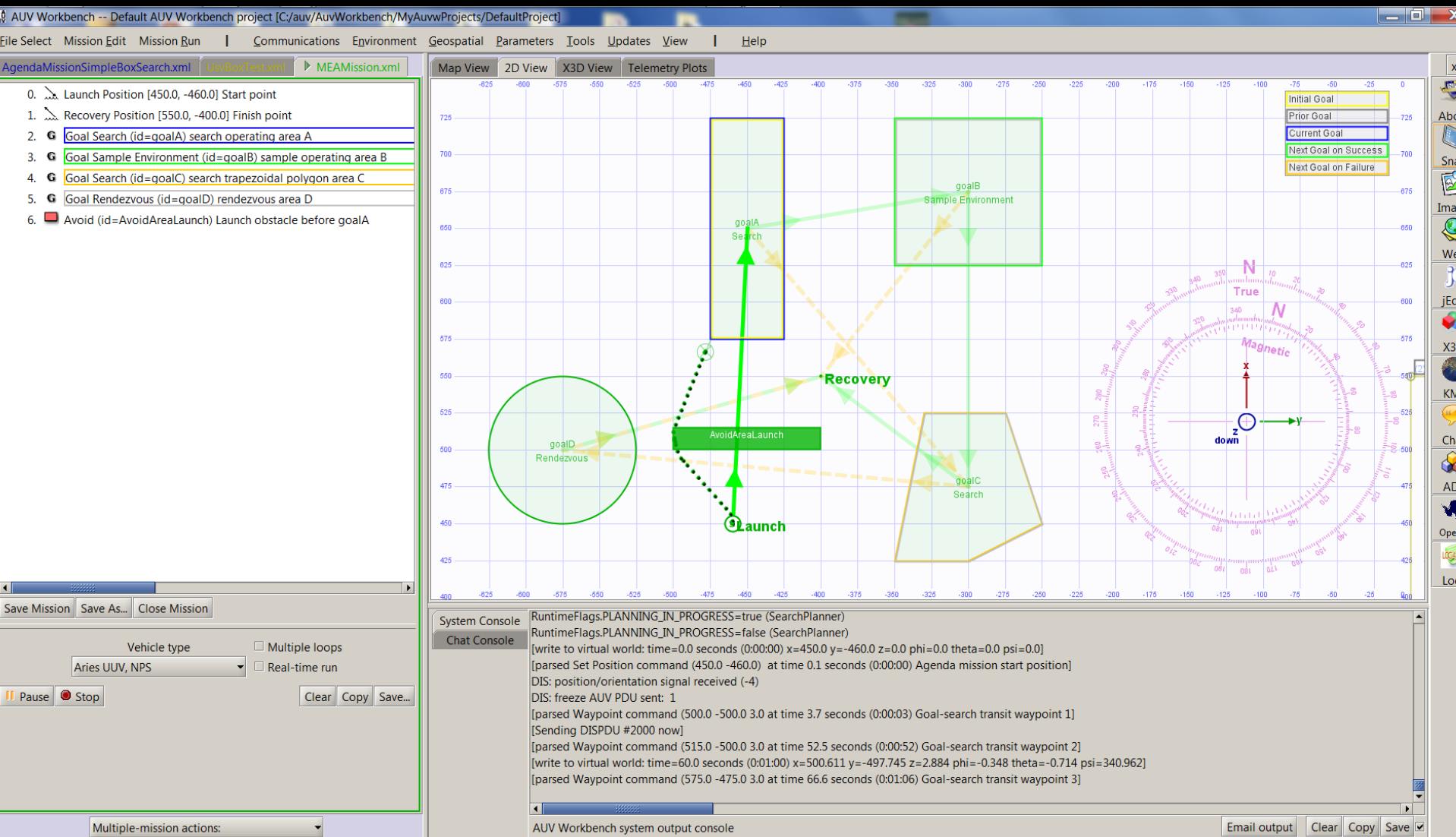
MeaMission.xml simulation preview



MeaMission.xml simulation 0, launch



MeaMission.xml simulation 1, transit



MeaMission.xml simulation 2, search

AUV Workbench -- Default AUV Workbench project [C:/auv/AuvWorkbench/MyAuwProjects/DefaultProject]

File Select Mission Edit Mission Run | Communications Environment Geospatial Parameters Tools Updates View | Help

AgendaMissionSimpleBoxSearch.xml UsvBoxTest.xml MEAMission.xml

Map View 2D View X3D View Telemetry Plots

Initial Goal Prior Goal Current Goal Next Goal on Success Next Goal on Failure

Launch Position [450.0, -460.0] Start point
Recovery Position [550.0, -400.0] Finish point
Goal Search (id=goalA) search operating area A
Goal Sample Environment (id=goalB) sample operating area B
Goal Search (id=goalC) search trapezoidal polygon area C
Goal Rendezvous (id=goalD) rendezvous area D
Avoid (id=AvoidAreaLaunch) Launch obstacle before goalA

The screenshot shows the AUV Workbench software interface. On the left, a sidebar displays the mission agenda with various tasks like 'Launch Position' and 'Recovery Position'. The main window features a 'Map View' showing a grid-based environment with several goals marked by colored shapes (yellow, blue, green). A green trajectory line with arrows indicates the path the AUV is expected to follow. To the right of the map is a detailed 'System Console' window showing log messages related to mission execution, such as 'Strategic level reading message from tactical level: SearchPatternComplete' and 'Strategic goal succeeded: Goal Search id=goalA goalIndex=0 displayIndex=2 search operating area A'. At the bottom, there's a 'Multiple-mission actions:' dropdown and a set of control buttons for 'Resume', 'Restart', 'Clear', 'Copy', and 'Save...'. The overall layout is typical of a scientific or engineering software for vehicle control and monitoring.

Save Mission Save As... Close Mission

Vehicle type Multiple loops
Aries UUV, NPS Real-time run

Resume Restart Clear Copy Save...

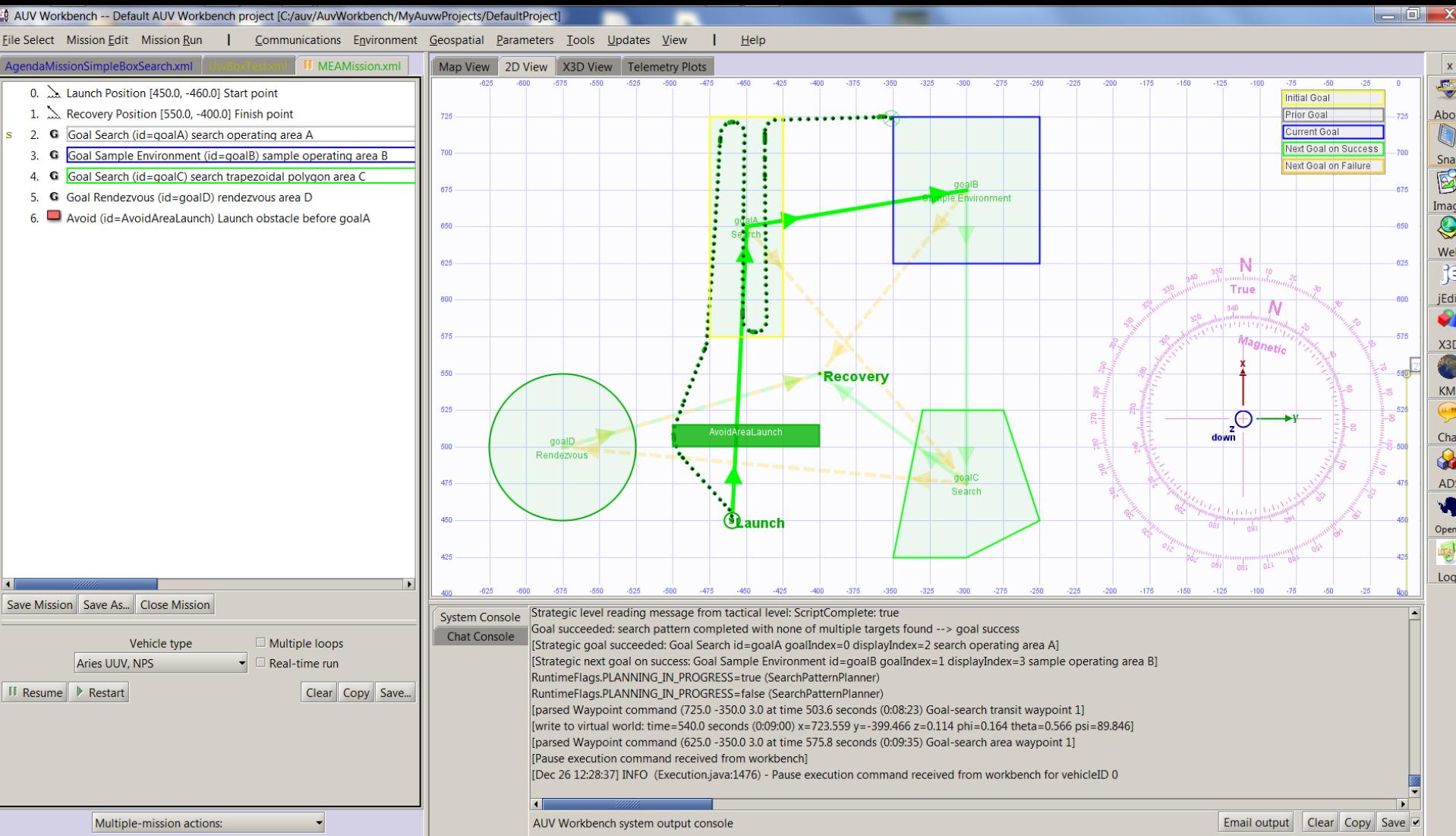
Multiple-mission actions:

System Console Chat Console

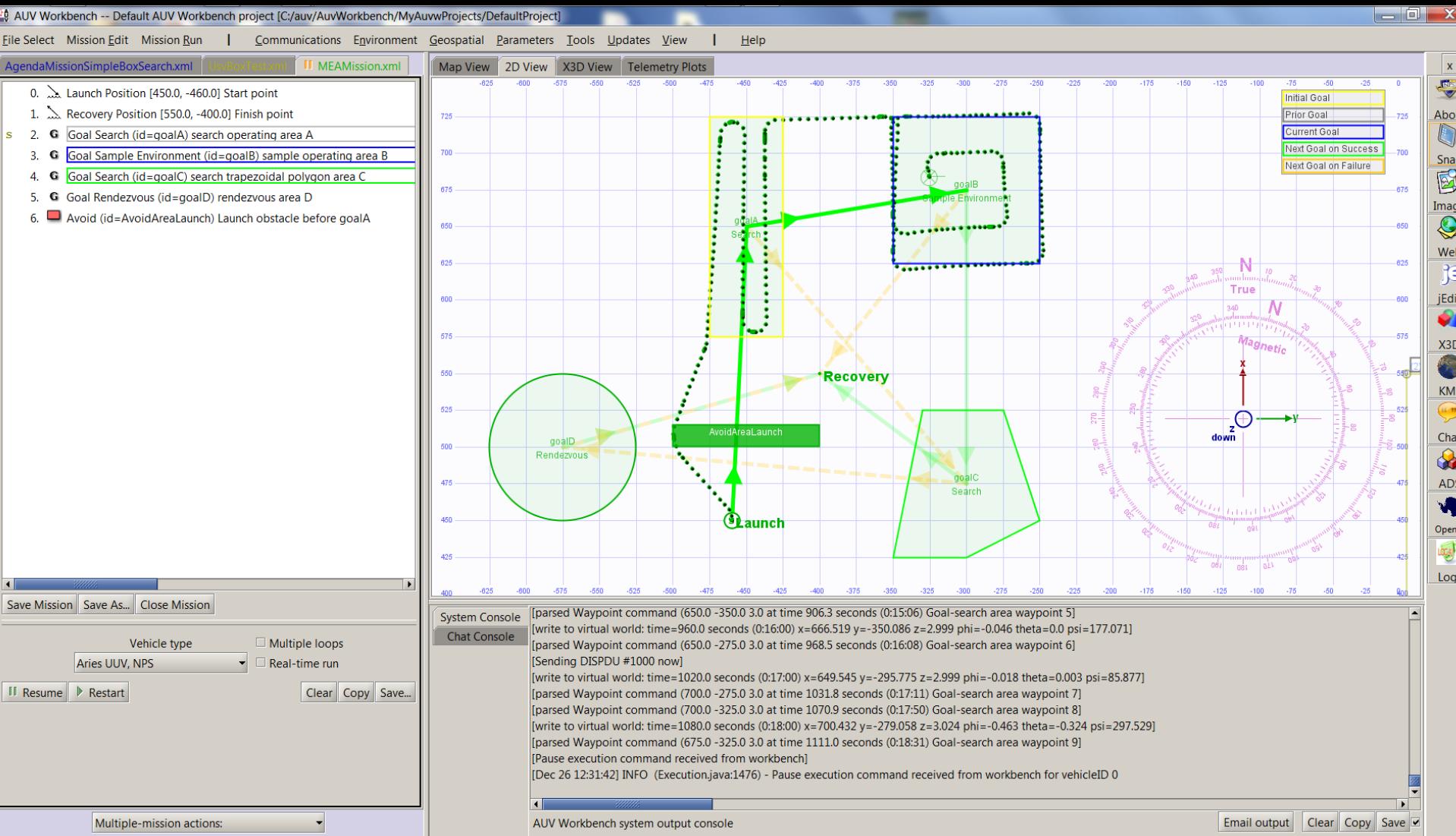
[Pause execution command received from workbench]
[Dec 26 12:46:00] INFO (Execution.java:1476) - Pause execution command received from workbench for vehicleID 0
[Pause execution command received from workbench]
[Dec 26 12:46:00] INFO (Execution.java:1476) - Pause execution command received from workbench for vehicleID 0
Strategic level reading message from tactical level: SearchPatternComplete
Strategic level reading message from tactical level: ScriptComplete: true
Goal succeeded: search pattern completed with none of multiple targets found --> goal success
[Strategic goal succeeded: Goal Search id=goalA goalIndex=0 displayIndex=2 search operating area A]
[Strategic next goal on success: Goal Sample Environment id=goalB goalIndex=1 displayIndex=3 sample operating area B]
RuntimeFlags.PLANNING_IN_PROGRESS=true (SearchPatternPlanner)
RuntimeFlags.PLANNING_IN_PROGRESS=false (SearchPatternPlanner)

Email output Clear Copy Save

MeaMission.xml simulation 3, transit



MeaMission.xml simulation 4, sample



MeaMission.xml simulation 5, transit

AUV Workbench -- Default AUV Workbench project [C:/auv/AuvWorkbench/MyAuwwProjects/DefaultProject]

File Select Mission Edit Mission Run | Communications Environment Geospatial Parameters Tools Updates View | Help

AgendaMissionSimpleBoxSearch.xml UsyBoxTestxml MEAMission.xml

Map View 2D View X3D View Telemetry Plots

Initial Goal Prior Goal Current Goal Next Goal on Success Next Goal on Failure

Launch Position [450.0, -460.0] Start point Recovery Position [550.0, -400.0] Finish point

Goal Search (id=goalA) search operating area A

Goal Sample Environment (id=goalB) sample operating area B

Goal Search (id=goalC) search trapezoidal polygon area C

Goal Rendezvous (id=goalD) rendezvous area D

Avoid (id=AvoidAreaLaunch) Launch obstacle before goalA

0. Launch Position [450.0, -460.0] Start point

1. Recovery Position [550.0, -400.0] Finish point

2. Goal Search (id=goalA) search operating area A

3. Goal Sample Environment (id=goalB) sample operating area B

4. Goal Search (id=goalC) search trapezoidal polygon area C

5. Goal Rendezvous (id=goalD) rendezvous area D

6. Avoid (id=AvoidAreaLaunch) Launch obstacle before goalA

Save Mission Save As... Close Mission

Vehicle type Multiple loops
Aries UUV, NPS Real-time run

Pause Stop Clear Copy Save...

System Console Chat Console

[Strategic next goal on success: Goal Search id=goalC goalIndex=2 displayIndex=4 search trapezoidal polygon area C]
RuntimeFlags.PLANNING_IN_PROGRESS=true (SearchPatternPlanner)
RuntimeFlags.PLANNING_IN_PROGRESS=false (SearchPatternPlanner)
RuntimeFlags.PLANNING_IN_PROGRESS=true (SearchPlanner)
RuntimeFlags.PLANNING_IN_PROGRESS=false (SearchPlanner)
[parsed Waypoint command (525.0 -330.0 3.0 at time 1155.1 seconds (0:19:15) Goal-search transit waypoint 1]
[write to virtual world: time=1200.0 seconds (0:20:00) x=629.857 y=-301.416 z=0.832 phi=-0.363 theta=-13.014 psi=196.446]
[parsed Waypoint command (475.0 -300.0 3.0 at time 1284.9 seconds (0:21:24) Goal-search area waypoint 0]
Mark setOnStation (vehicleID=0) Search goal=goalC (search trapezoidal polygon area C)
[parsed Waypoint command (490.0 -300.0 3.0 at time 1330.6 seconds (0:22:10) Goal-search area waypoint 1]
[parsed Waypoint command (490.0 -285.0 3.0 at time 1334.5 seconds (0:22:14) Goal-search area waypoint 2]

Multiple-mission actions: Email output Clear Copy Save

The screenshot shows the AUV Workbench software interface. The main window displays a 2D map view of an operating area with various mission points and obstacles. The map includes a coordinate grid from -625 to 0 on both axes. Key mission points are labeled: 'Launch' at (450.0, -460.0), 'Recovery' at (550.0, -400.0), 'goalA Search' (operating area A), 'goalB Sample Environment' (operating area B), 'goalC Search' (trapezoidal polygon area C), and 'goalD Rendezvous' (rendezvous area D). An 'AvoidAreaLaunch' obstacle is shown as a green rectangle. The 'System Console' tab in the bottom right shows the execution of mission steps, including planning and waypoint commands. The 'Chat Console' tab is also visible.

MeaMission.xml simulation 6, search

AUV Workbench -- Default AUV Workbench project [C:/auv/AuvWorkbench/MyAuwwProjects/DefaultProject]

File Select Mission Edit Mission Run | Communications Environment Geospatial Parameters Tools Updates View | Help

AgendaMissionSimpleBoxSearch.xml UsrBoxTestxml MEAMission.xml

0. Launch Position [450.0, -460.0] Start point
1. Recovery Position [550.0, -400.0] Finish point
2. Goal Search (id=goalA) search operating area A
3. Goal Sample Environment (id=goalB) sample operating area B
4. Goal Search (id=goalC) search trapezoidal polygon area C
5. Goal Rendezvous (id=goalD) rendezvous area D
6. Avoid (id=AvoidAreaLaunch) Launch obstacle before goalA

Map View 2D View X3D View Telemetry Plots

The interface shows a mission map with various goals and areas. A green arrow labeled 'Launch' points from the bottom towards a yellow circle labeled 'goalD Rendezvous'. Another green arrow labeled 'Search' points from the left towards a yellow rectangle labeled 'goalA Search'. A green arrow labeled 'Recovery' points from the center towards a yellow rectangle labeled 'goalB Sample Environment'. A blue arrow labeled 'AvoidAreaLaunch' points from the bottom towards a blue trapezoid labeled 'goalC Search'. The 3D plot on the right shows a circular trajectory with axes X, Y, and Z.

Initial Goal Prior Goal Current Goal Next Goal on Success Next Goal on Failure

Save Mission Save As... Close Mission

Vehicle type Multiple loops
Aries UUV, NPS Real-time run

Resume Restart Clear Copy Save...

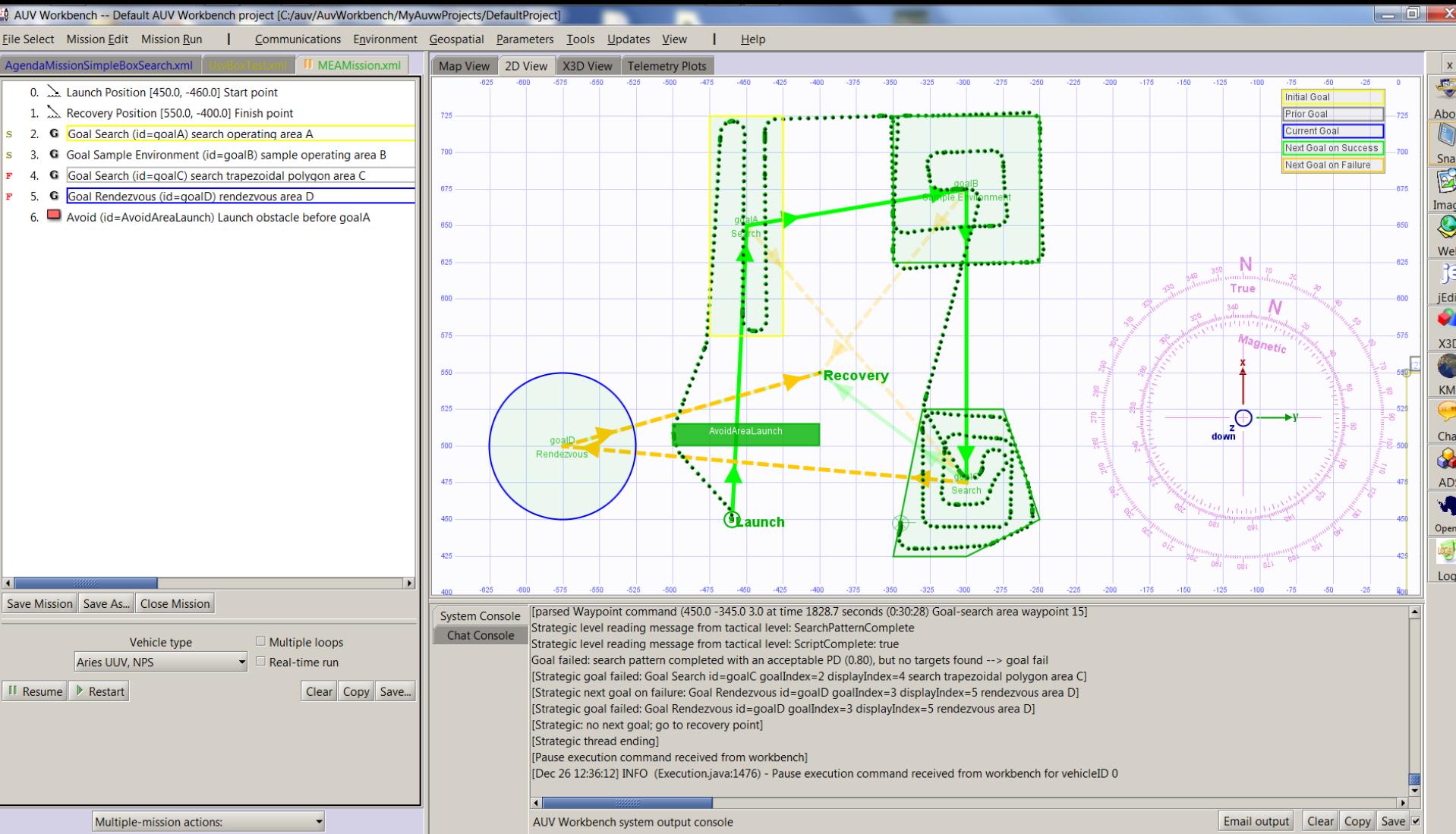
System Console Chat Console

```
[parsed Waypoint command (490.0 -285.0 3.0 at time 1334.5 seconds (0:22:14) Goal-search area waypoint 2]
[parsed Waypoint command (460.0 -285.0 3.0 at time 1348.5 seconds (0:22:28) Goal-search area waypoint 3]
[parsed Waypoint command (460.0 -315.0 3.0 at time 1393.2 seconds (0:23:13) Goal-search area waypoint 4]
[parsed Waypoint command (505.0 -315.0 3.0 at time 1416.9 seconds (0:23:36) Goal-search area waypoint 5]
[parsed Waypoint command (505.0 -270.0 3.0 at time 1452.4 seconds (0:24:12) Goal-search area waypoint 6]
[parsed Waypoint command (445.0 -270.0 3.0 at time 1489.5 seconds (0:24:49) Goal-search area waypoint 7]
[ Sending DISPDU #1500 now]
[parsed Waypoint command (445.0 -330.0 3.0 at time 1536.7 seconds (0:25:36) Goal-search area waypoint 8]
[parsed Waypoint command (520.0 -330.0 3.0 at time 1583.7 seconds (0:26:23) Goal-search area waypoint 9]
[Pause execution command received from workbench]
[Dec 26 12:34:28] INFO (Execution.java:1476) - Pause execution command received from workbench for vehicleID 0
```

Multiple-mission actions: Email output Clear Copy Save

AUV Workbench system output console

MeaMission.xml simulation 7, rdvu fail



MeaMission.xml simulation 8, transit

AUV Workbench -- Default AUV Workbench project [C:/auv/AuvWorkbench/MyAuwProjects/DefaultProject]

File Select Mission Edit Mission Run | Communications Environment Geospatial Parameters Tools Updates View | Help

AgendaMissionSimpleBoxSearch.xml UsvBoxTest.xml MEAMission.xml

Map View 2D View X3D View Telemetry Plots

Initial Goal Prior Goal Current Goal Next Goal on Success Next Goal on Failure

Launch Position [450.0, -460.0] Start point
Recovery Position [550.0, -400.0] Finish point
Goal Search (id=goalA) search operating area A
Goal Sample Environment (id=goalB) sample operating area B
Goal Search (id=goalC) search trapezoidal polygon area C
Goal Rendezvous (id=goalD) rendezvous area D
Avoid (id=AvoidAreaLaunch) Launch obstacle before goalA

0. Launch Position [450.0, -460.0] Start point
1. Recovery Position [550.0, -400.0] Finish point
2. Goal Search (id=goalA) search operating area A
3. Goal Sample Environment (id=goalB) sample operating area B
4. Goal Search (id=goalC) search trapezoidal polygon area C
5. Goal Rendezvous (id=goalD) rendezvous area D
6. Avoid (id=AvoidAreaLaunch) Launch obstacle before goalA

Map View: A 2D grid-based map showing mission areas and vehicle trajectories. The map includes several regions defined by dashed lines: 'goalA Search' (yellow), 'goalB Sample Environment' (green), 'goalC Search' (orange), and 'goalD Rendezvous' (blue). A green arrow labeled 'Recovery' points from the center towards the top right. A red arrow labeled 'Launch' points upwards. A yellow arrow labeled 'AvoidAreaLaunch' points downwards. A blue circle labeled 'goalD Rendezvous' is located in the bottom left corner.

2D View: A 2D coordinate system showing vehicle trajectories. It features concentric circular and elliptical paths, likely representing orbital or circular mission profiles. A vertical axis is labeled 'True' and 'Magnetic'.

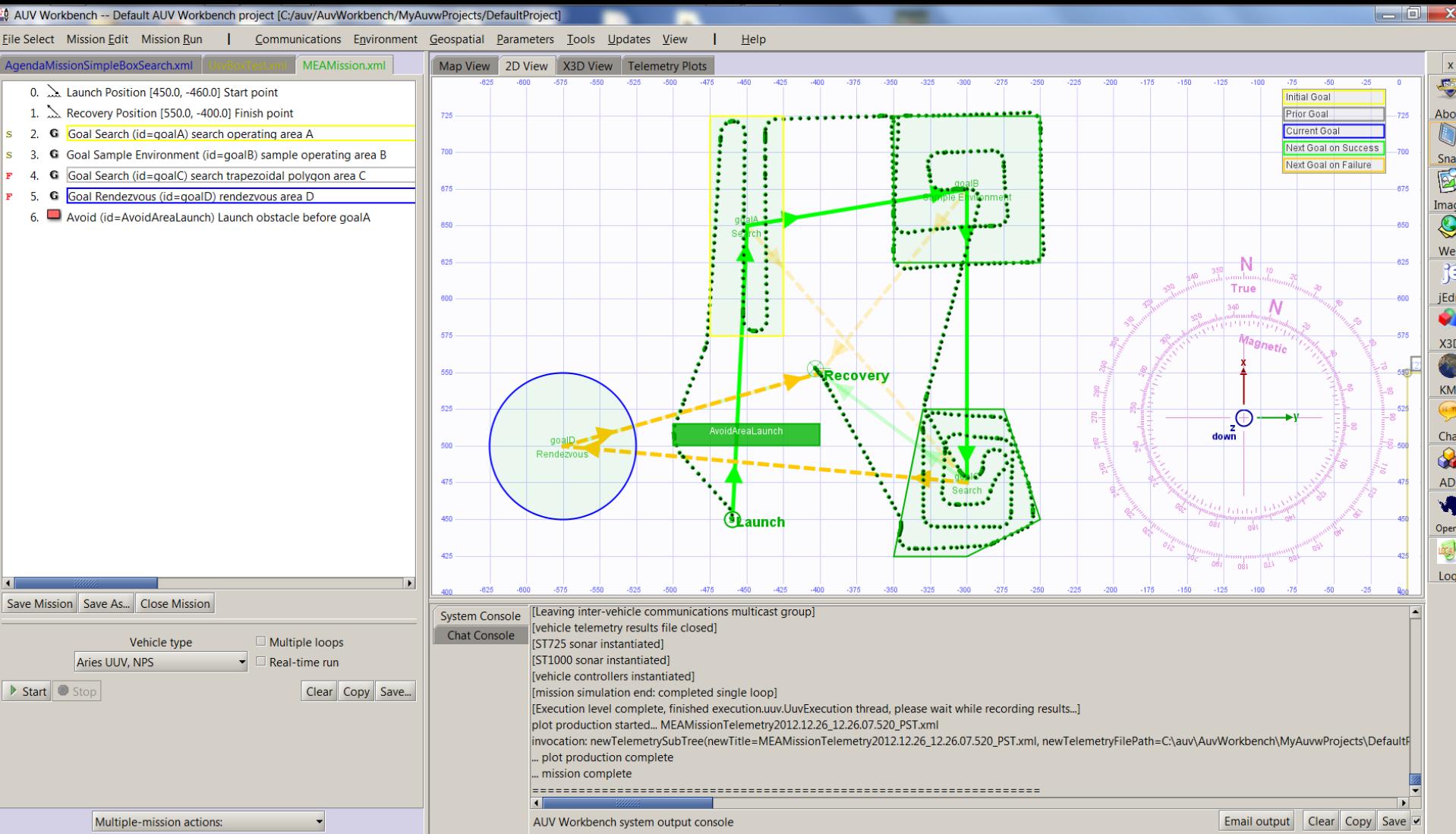
X3D View: A 3D coordinate system showing vehicle trajectories. It features concentric spherical shells, likely representing orbital or circular mission profiles. A vertical axis is labeled 'True' and 'Magnetic'.

Telemetry Plots: A plot showing vehicle position over time. The x-axis represents position and the y-axis represents time. The plot shows a series of points connected by lines, representing the vehicle's path through the mission areas.

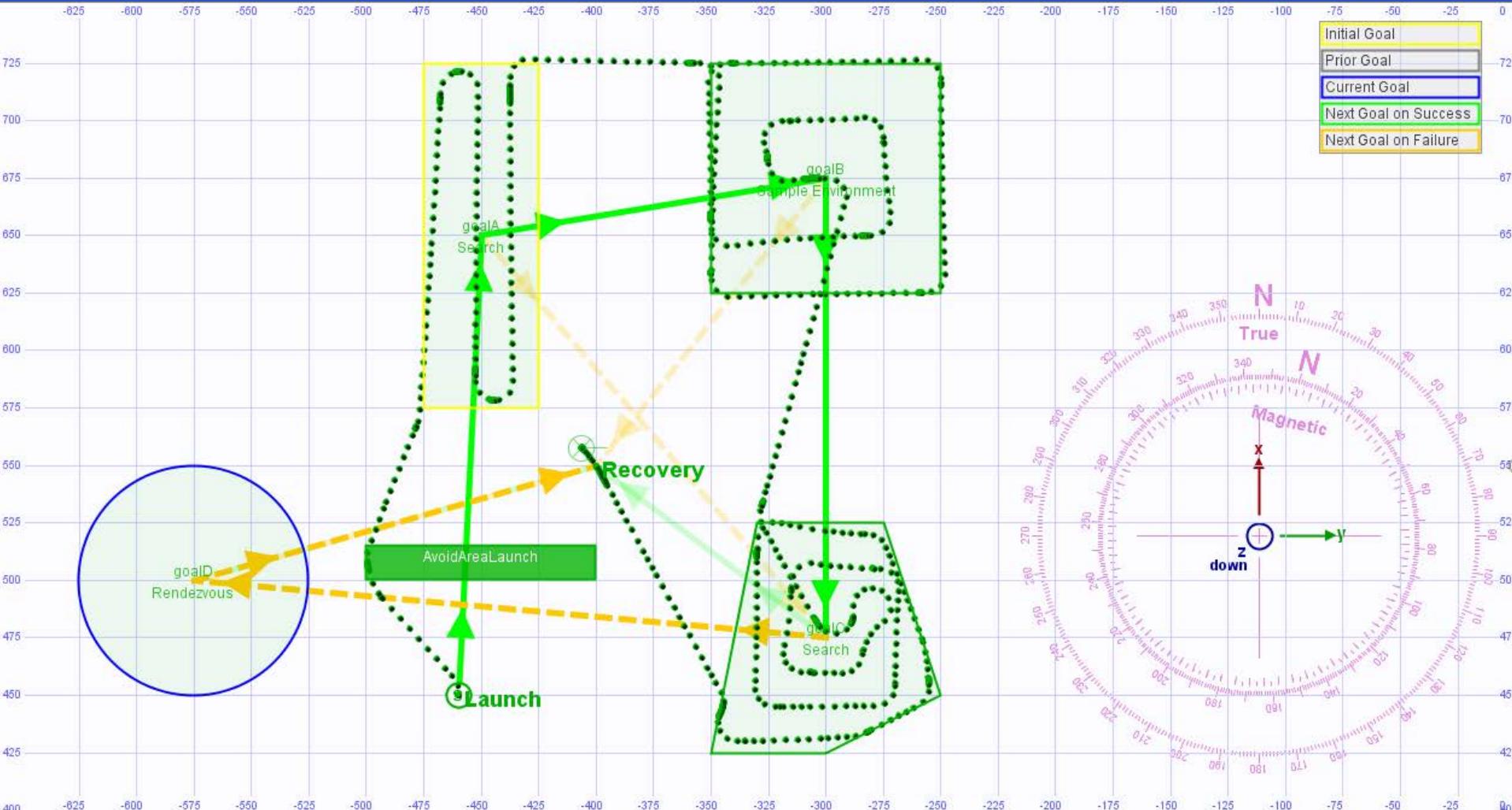
System Console: Strategic next goal on failure: Goal Rendezvous id=goalD goalIndex=3 displayIndex=5 rendezvous area D
Strategic goal failed: Goal Rendezvous id=goalD goalIndex=3 displayIndex=5 rendezvous area D
Strategic: no next goal; go to recovery point
[Strategic thread ending]
[Pause execution command received from workbench]
[Dec 26 12:36:12] INFO (Execution.java:1476) - Pause execution command received from workbench for vehicleID 0
[Resume execution command received from workbench]
[Dec 26 12:37:19] INFO (Execution.java:1482) - Resume execution command received from workbench for vehicleID 0
[parsed Waypoint Command (550.0 -400.0 3.0 at time 1849.5 seconds (0:30:49) Goal-search transit waypoint 1]
[Pause execution command received from workbench]
[Dec 26 12:37:31] INFO (Execution.java:1476) - Pause execution command received from workbench for vehicleID 0

Chat Console: Multiple-mission actions: Email output Clear Copy Save

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. Naval forces can only command mission orders that are:

- Understandable by (legally culpable) humans
- Reliably and safely executable by robots



Humans + system validation
that orders are well-defined

together with

ethically sound constraints



Mission Execution Ontology (MEO)

Semantic Web supports well-defined expression of logical rules and relationships

Mission goals, capabilities, tasks and ethical requirements can be defined formally

- With tractable computation by reasoning engines

MEO applies Web Ontology Language (OWL), Resource Description Framework (RDF) lets mission correctness be validated logically

- Open standards, World Wide Web Consortium (W3C)
- Multiple proven implementations available



Responsibility and accountability

Culpability, liability are problematic for AI agents

Command responsibility must be accompanied by Authority and Accountability

In this approach, humans remain responsible for correct mission tasking and approval of ethical constraints – preserving necessary oversight for release of potentially lethal force.



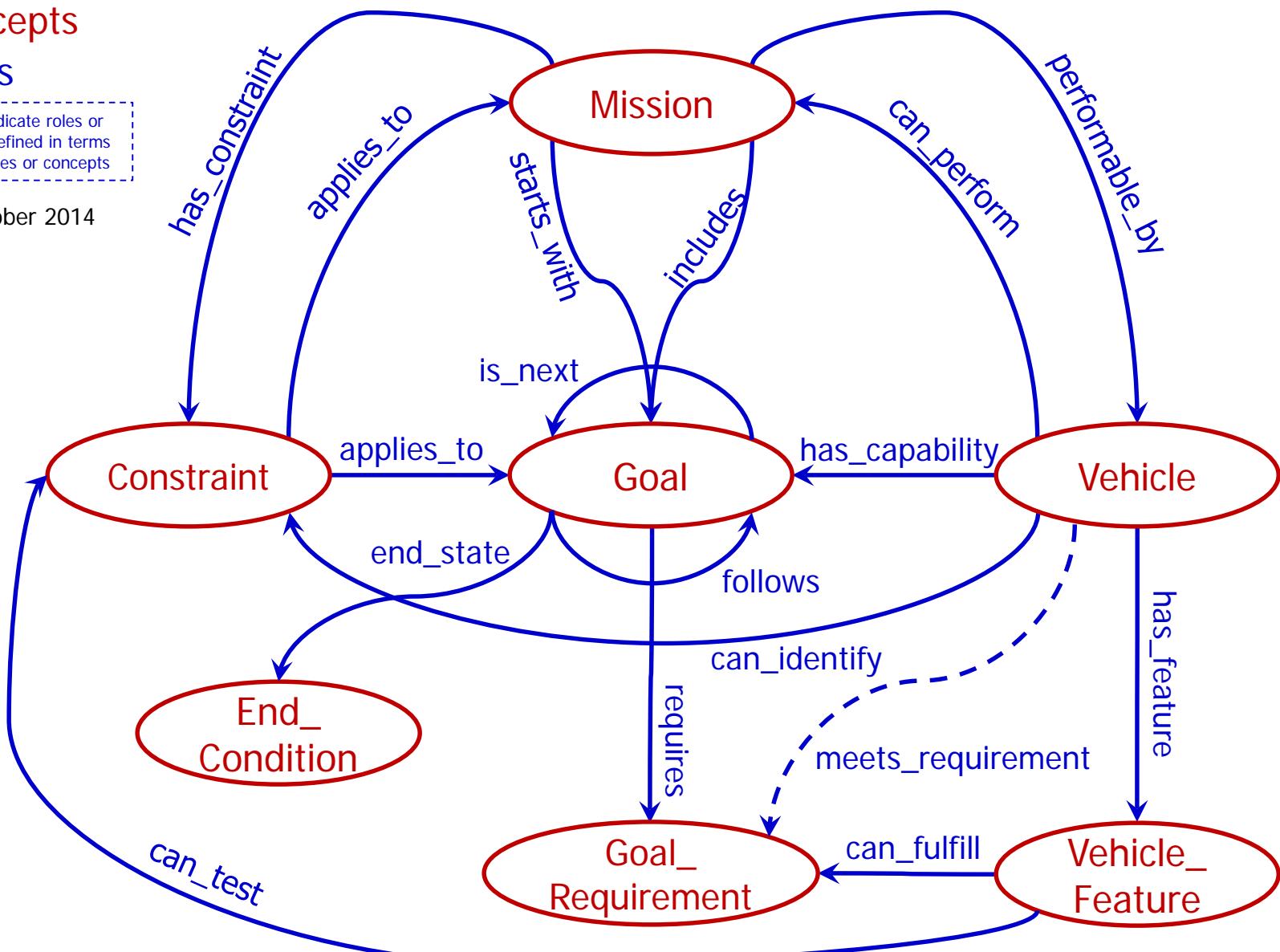
Unmanned Vehicle Ethics Ontology

Concepts

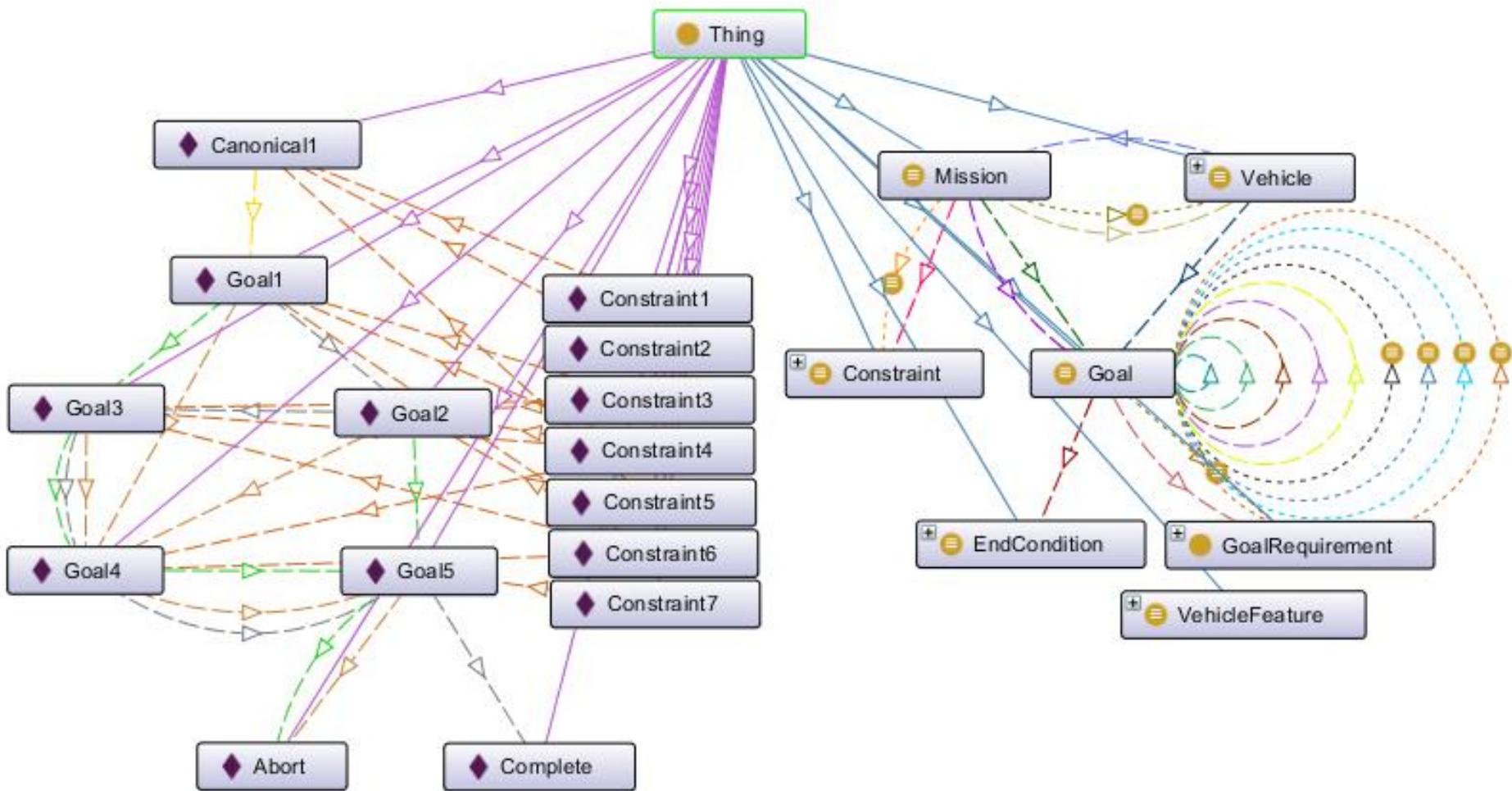
Roles

Dashes indicate roles or concepts defined in terms of other roles or concepts

15 October 2014



Example Mission Validation using Protégé Tool



Review of take-away points

1. Humans in military units effectively deal with moral challenges without ethical quandaries.
2. Ethical behaviors don't define mission plans; ethical constraints inform mission plans.
3. Naval personnel can only issue orders that are:
 - Understandable by (legally culpable) humans, and
 - Reliably and safely executable by robots.
4. Robot mission tasking can be carefully reviewed and approved by humans, with formal validation of correctness and ethical completeness.



Influential student theses now possible

Define, explore core missions for today's Navy

Use scenario simulations to test ethical challenges related to autonomous robot lethality

Apply ethical tasking to many unmanned systems, support military human-robot team concepts



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

Duane Davis, Ph.D.

dtdavi1@nps.edu

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



Contact

Curt Blais

cblais@nps.edu

Code MV/BI
Naval Postgraduate School
Monterey California 93943-5000 USA
1.831.656.3215 work



Contact

Robert B. McGhee, Ph.D.

robertbmcghee@gmail.com

Professor Emeritus
Computer Science
Naval Postgraduate School
Monterey California 93943-5000 USA





ETHICAL MISSION TASKING AND EXECUTION FOR MARITIME ROBOTS

- Lethality requires ethical & legal basis, supervised by military teams.
- Effective robot tasking can resemble tactical tasking of humans afloat.
- Careful application of goal constraints makes ethical control feasible.
- Robot missions then complement and extend naval operation orders.
- Semantic Web logic can confirm ethical correctness and completeness.
- Next steps: extend 2 decades of work with simulation scenario testing.

"Ethical constraints on robot mission execution are possible today. There is no need to wait for future developments in Artificial Intelligence (AI). It is a moral imperative that ethical constraints in some form be introduced immediately into the software of all robots that are capable of inflicting unintended or deliberate harm to humans or property." - Robert McGhee, April 2016

- Draft paper, [IEEE/MTS OCEANS 2016 Conference](#), Monterey CA, 19-23 SEP 2016.
- Authors Don Brutzman, Duane Davis, Curtis Blais, and Robert McGhee, NPS.
- Review requests and feedback are welcome. Contact: brutzman@nps.edu

