MOOS-IvP Release 20.08 – What is new and Upcoming

Aug 10th, 2022

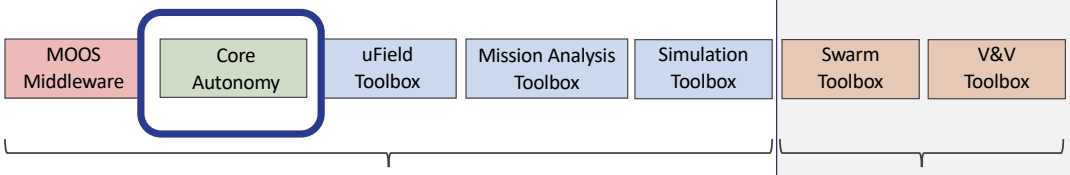
Michael Benjamin,
MIT Building 5-214
mikerb@mit.edu

1

Marine Autonomy Software

The MOOS-IvP Open Source Autonomy Project

- In Development since ~1998.
- First launched online in 2006.
- Latest release in August 2022.



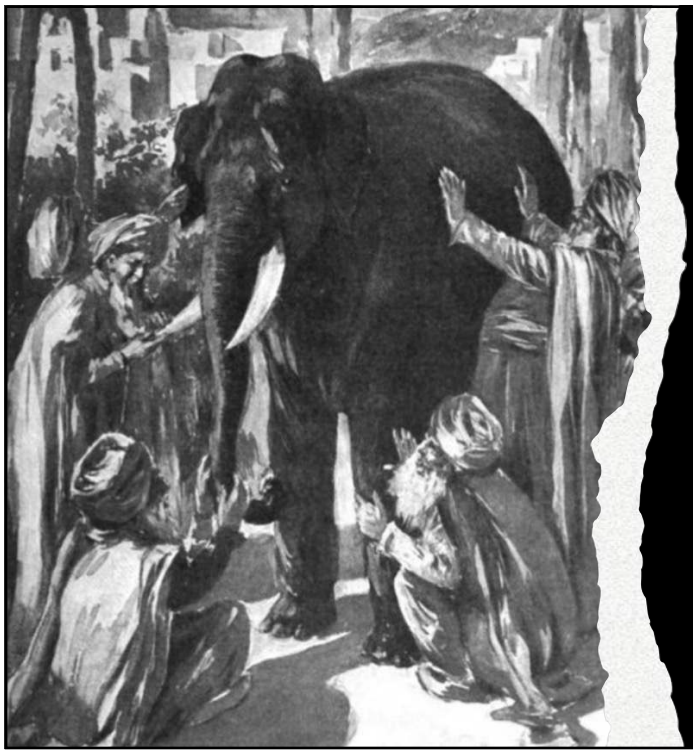
MOOS-IvP Public code

- ~40 work years of effort
- Ported to dozens of platform types
- Full documentation and training

Currently Non-Public code

- ~5 work years of recent efforts

2

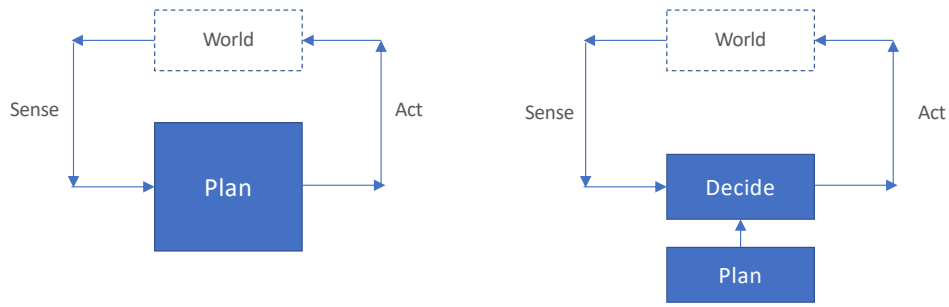


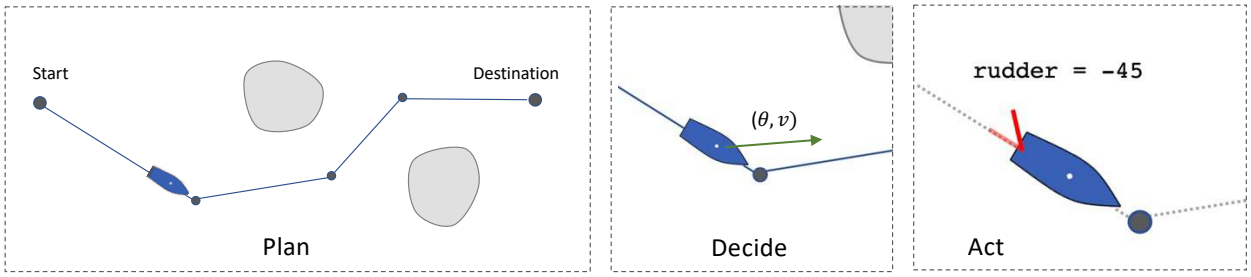
What is "Core" Marine Autonomy

- Decision-Making (the helm plus behaviors)
- Event Managers (the middleman between sensing and decision-making)

3

Sense – Decide - Act

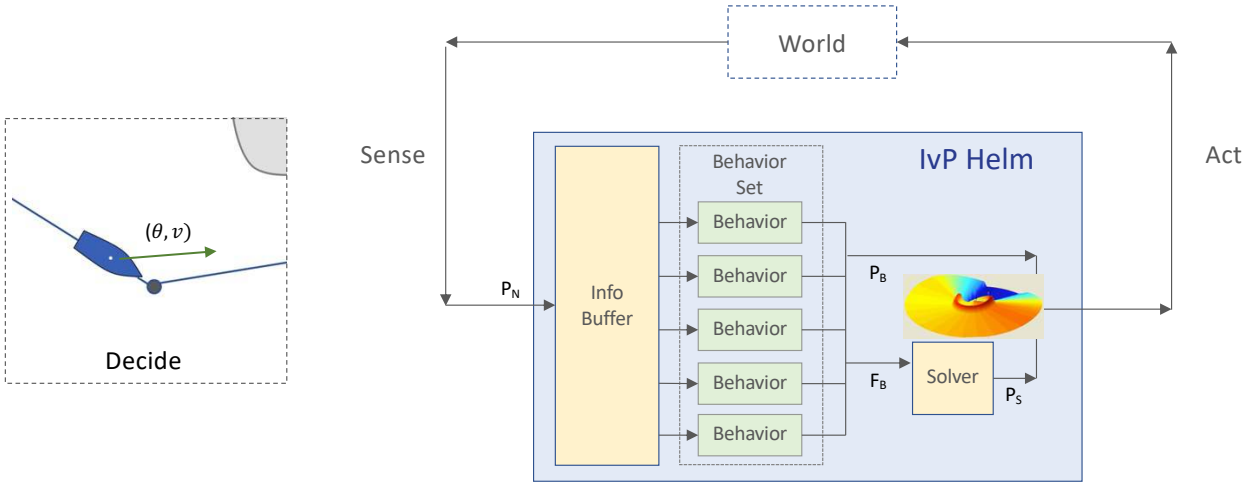




Michael Benjamin, mikerb@mit.edu

4

IvP Helm Autonomous Decision-Making



5

Behavior Compromising

Behavior 1 → $f_1(x_1, x_2, \ldots, x_n)$

Behavior 2 → $f_2(x_1, x_2, \ldots, x_n)$

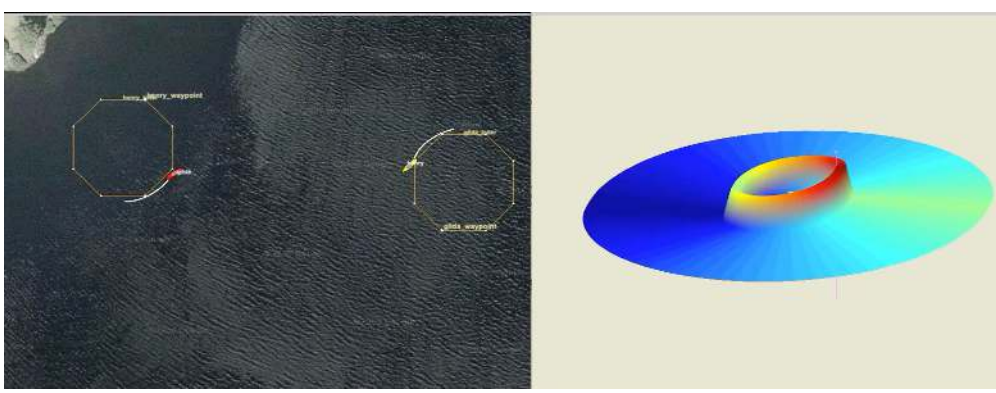
Behavior 3 → $f_3(x_1, x_2, \ldots, x_n)$

Individual Objective Functions

$$\mathbf{x}^* = argmax_{\mathbf{x}} \sum_{i=1}^{k} (w_i \cdot f_i(\mathbf{x}))$$

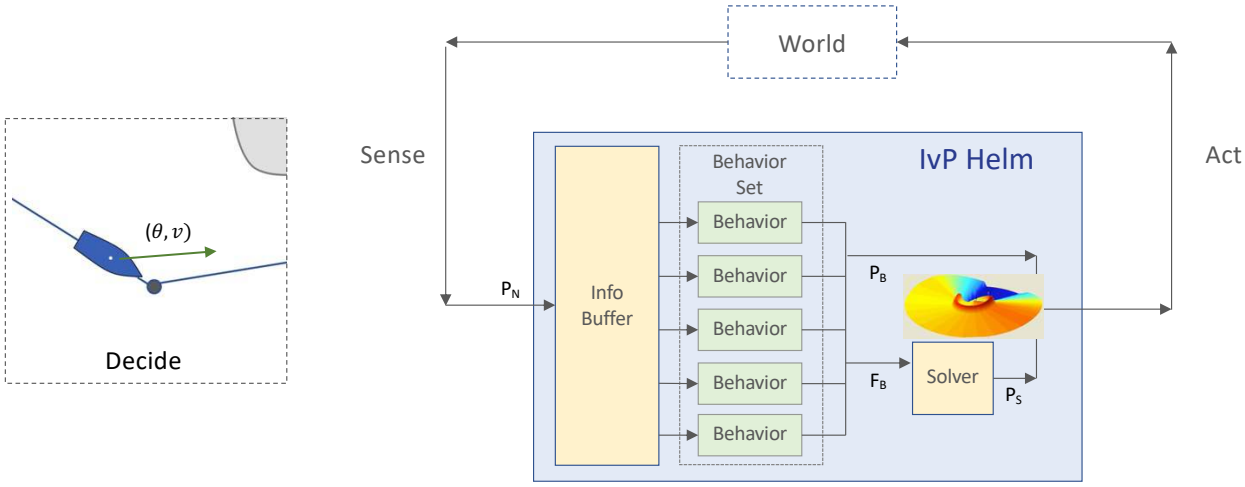
IvP Solver

→ Action

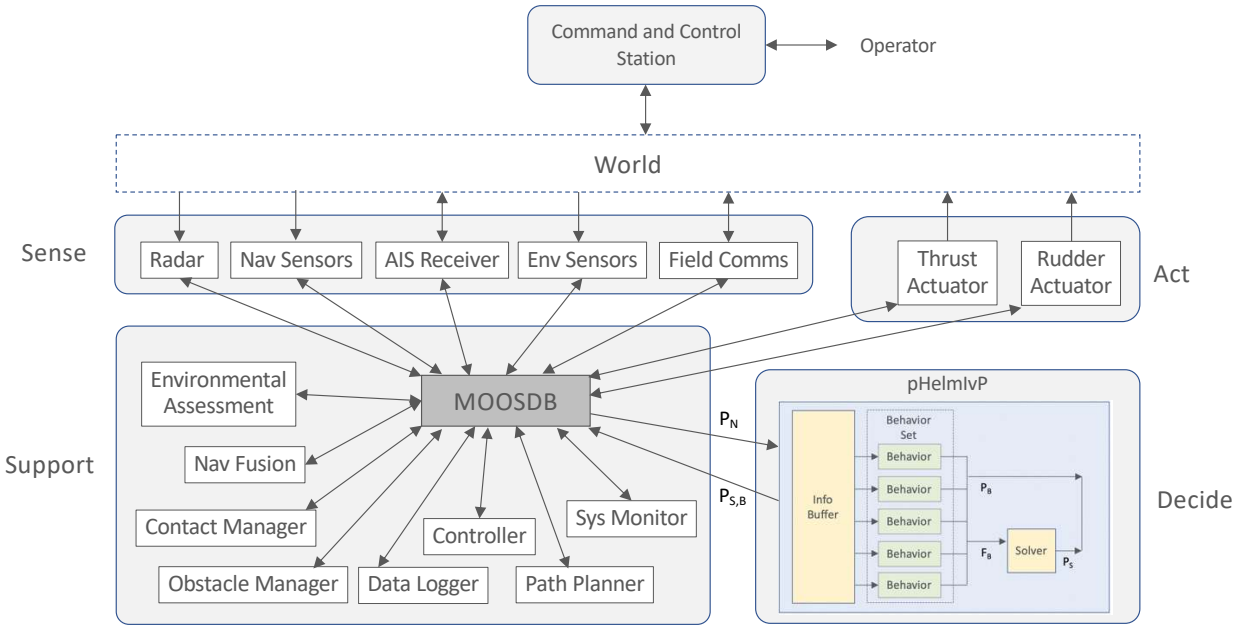


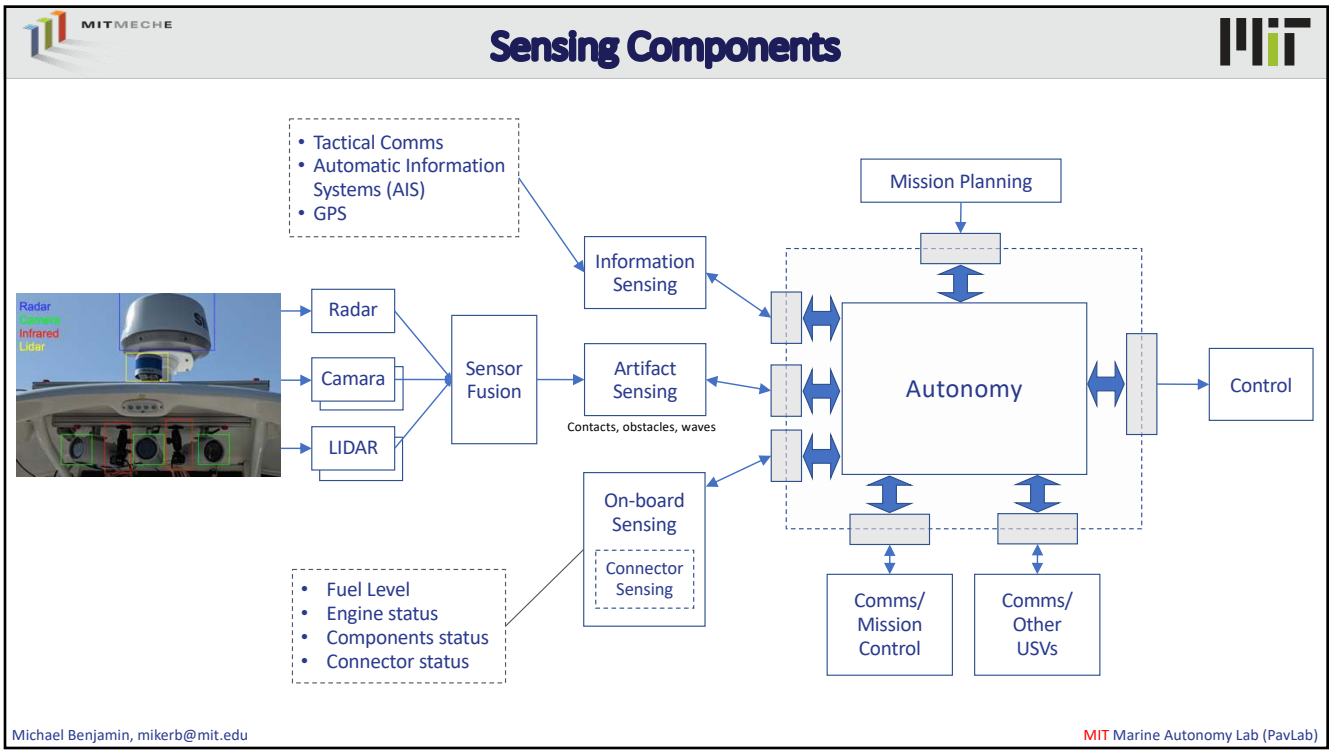
6

IvP Helm Autonomous Decision-Making

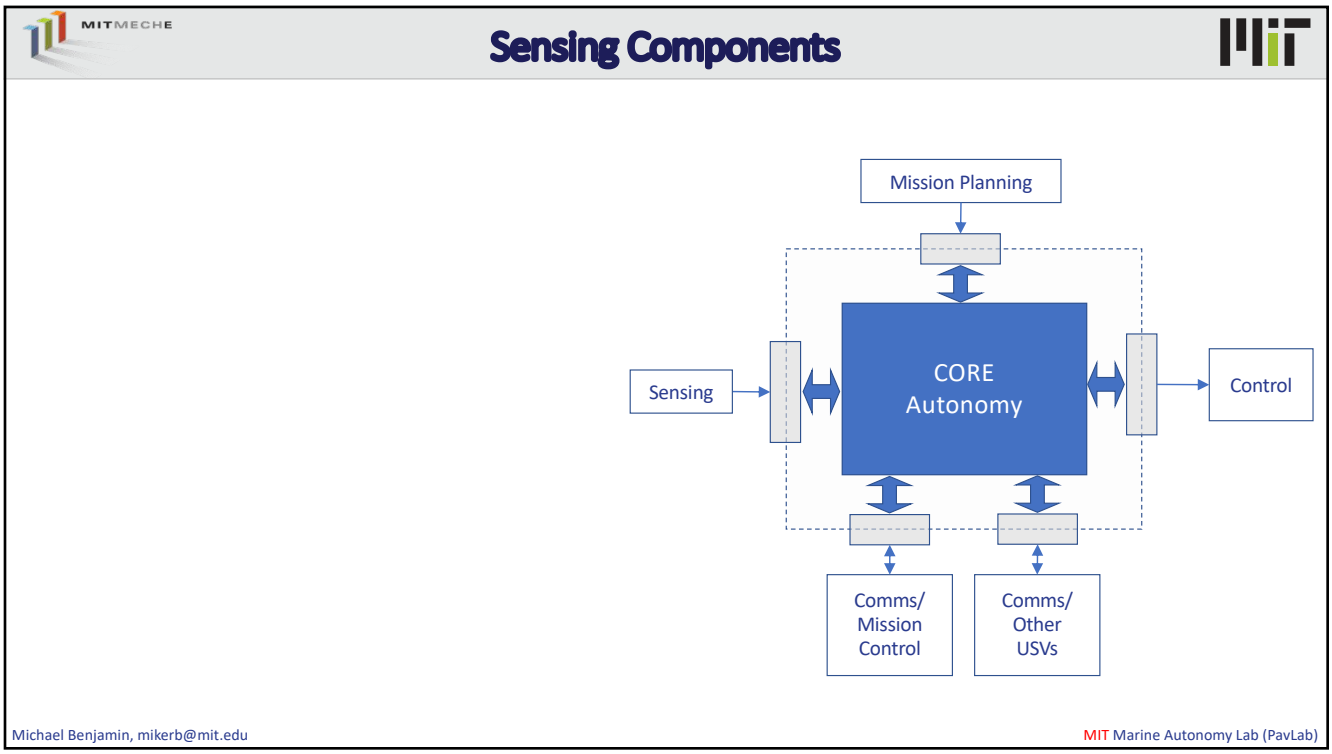


Autonomy System MOOS Perspective




MITMECHE
Sensing Components
• Tactical Comms
• Automatic Information Systems (AIS)
• GPS
Mission Planning
Information Sensing
Radar
Camara
LIDAR
Sensor Fusion
Artifact Sensing
Contacts, obstacles, waves
Autonomy
Control
On-board Sensing
Connector Sensing
• Fuel Level
• Engine status
• Components status
• Connector status
Comms/ Mission Control
Comms/ Other USVs
Michael Benjamin, mikerb@mit.edu
MIT Marine Autonomy Lab (PavLab)

9

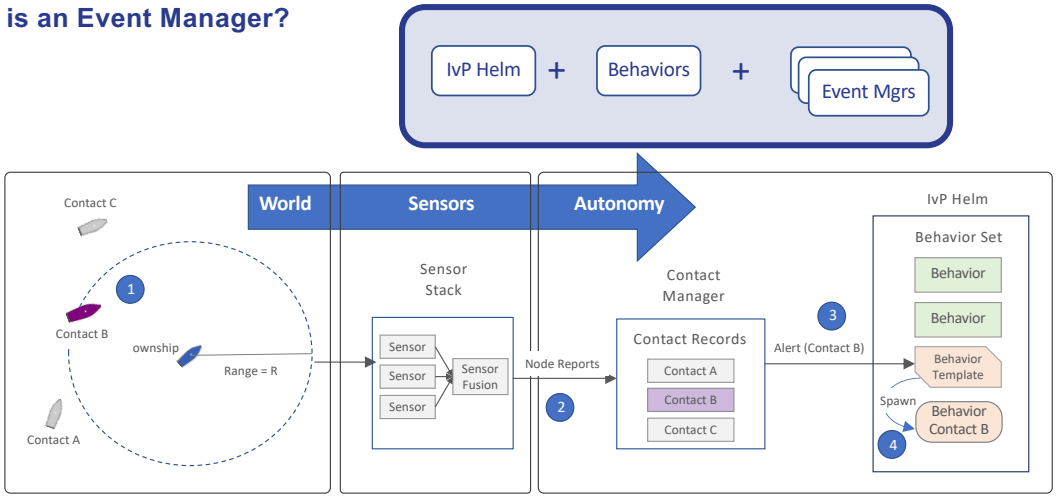

MITMECHE
Sensing Components
Mission Planning
Sensing
CORE Autonomy
Control
Comms/ Mission Control
Comms/ Other USVs
Michael Benjamin, mikerb@mit.edu
MIT Marine Autonomy Lab (PavLab)

10

Core Autonomy

Core Autonomy: The Helm, Native Behaviors, Behavior Inftrastructure, and Event Managers

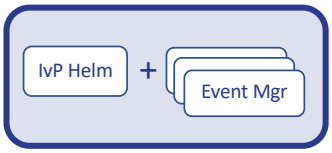
What is an Event Manager?

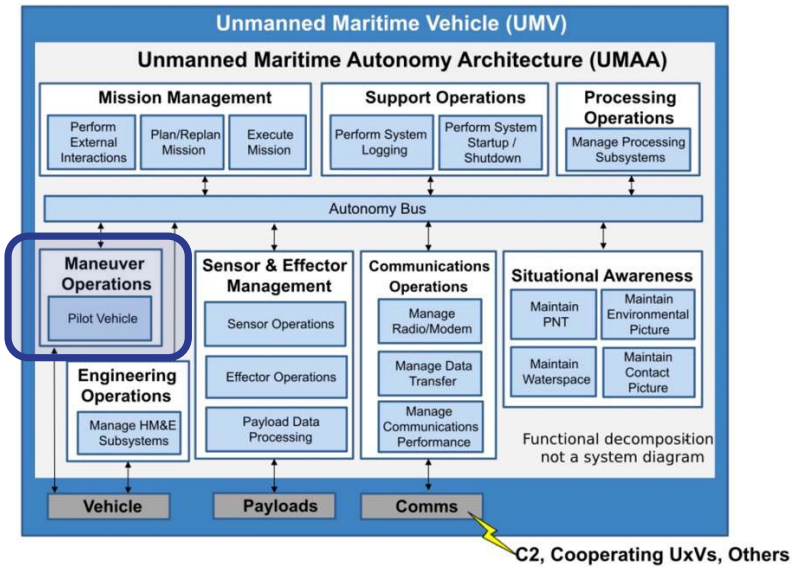


Autonomy System UMAA Perspective

Unmanned Maritime Autonomy Architecture (UMAA) Maneuver Operations (MO) Interface Control Document (ICD) (UMAA-SPEC-MOICD)
MDE Version 4.2.1 Commit 7c2b513
UMAA Spec Commit f2fa426

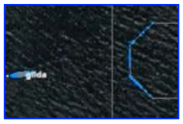
Core Autonomy:
Helm + Event Managers



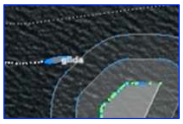


Example Mission
Helm, Event Managers and Simulated Sensing

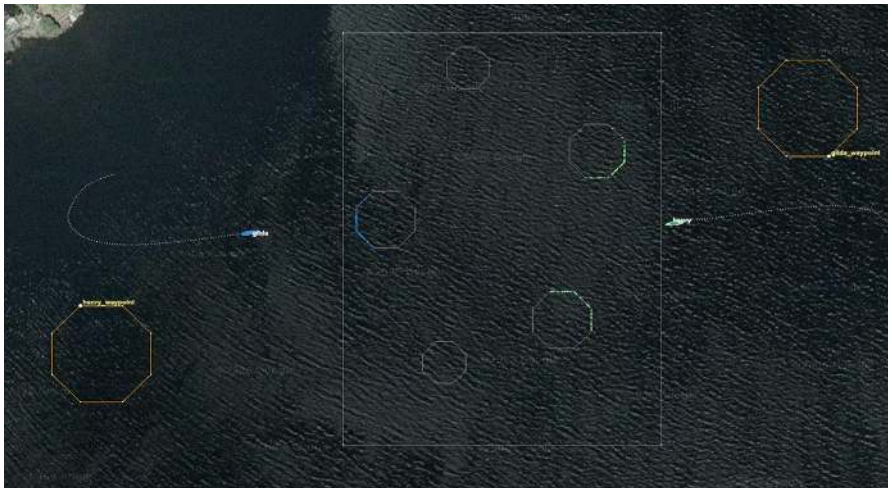
Simulated LIDAR



Obstacle Avoidance:



Collision Avoidance:



Core Autonomy
Modifications/Additions in Release 22.8

Contact Manager (pContactMgrV20)

- New support for Exclusion Filters to selectively ignore contacts based on group, name, type, or region.
- Exclusion filter configured per behavior.
- New algorithm for *resolving* contacts.
- Better memory management and guards agains unbounded growth.
- Support for managing contacts comprising collaborative groups (to support Swarm Toolbox)

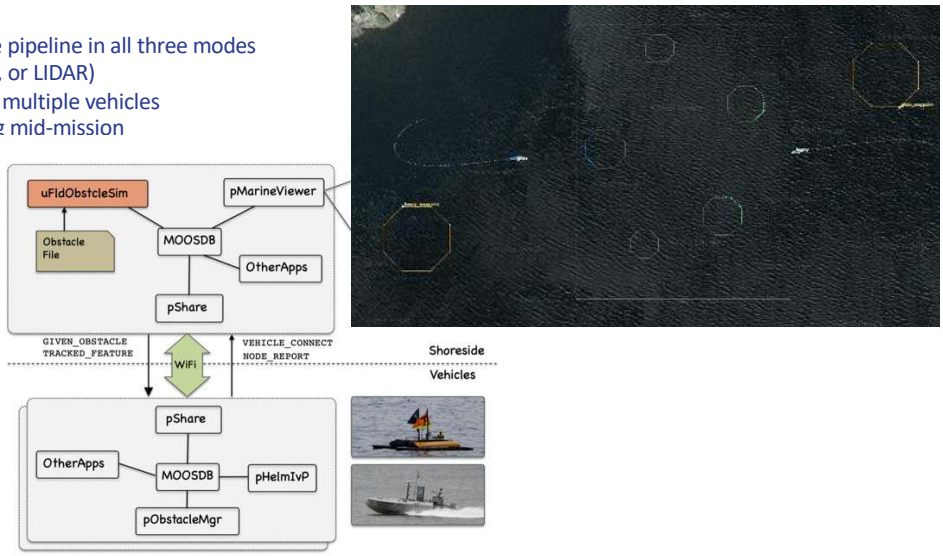
Obstacle Manager (pObstacleMgr)

- Support for obstacles derived from (a) LIDAR points (b) processed obstacles, or (c) configured obstacles
- Incoming (processed) obstacles can be configured with a duration and unique key
- Zero duration obstacles are essentially false-detection retractions.
- Better memory management and guards against unbounded growth

Core Autonomy
Modifications/Additions in Release 22.8

uFldObstacleSim

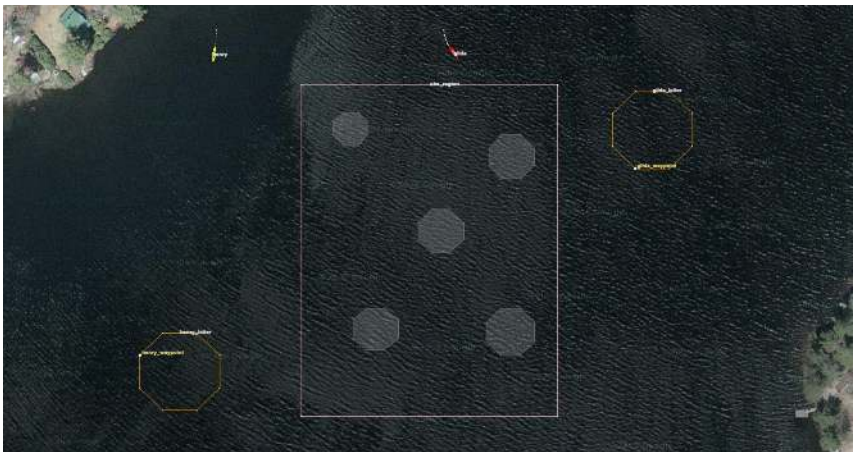
- Simulates obstacle pipeline in all three modes (fixed, processed, or LIDAR)
- Distributed across multiple vehicles
- Dynamic re-setting mid-mission



Core Autonomy
Modifications/Additions in Release 22.8

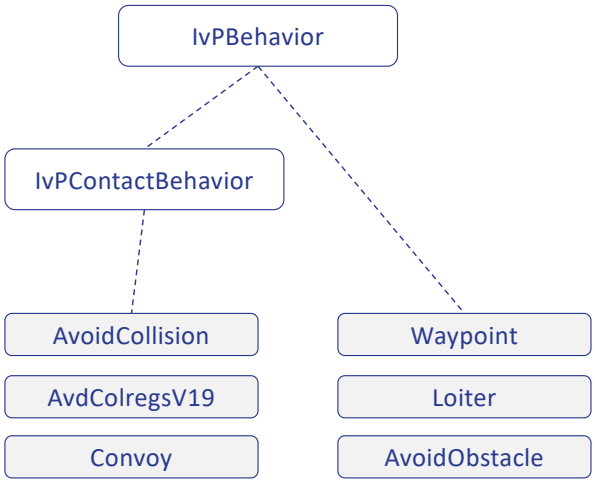
uFldObstacleSim

- Simulates obstacle pipeline in all three modes
- Distributed across multiple vehicles
- Dynamic re-setting mid-mission



Core Autonomy
Modifications/Additions in Release 22.8

Behavior Structure



IvPBehavior Superclass

- Explicit declaration of Goal and Constraint behaviors. Helm will guard to ensure continuous presence of a goal behavior.

IvPContactBehavior Superclass

- Expanded support of Events and Macros
- Customizable **cn_flag** to support mission and automated testing.
- Exclusion Filters on a per-contact basis

Michael Benjamin, mikerb@mit.edu

17

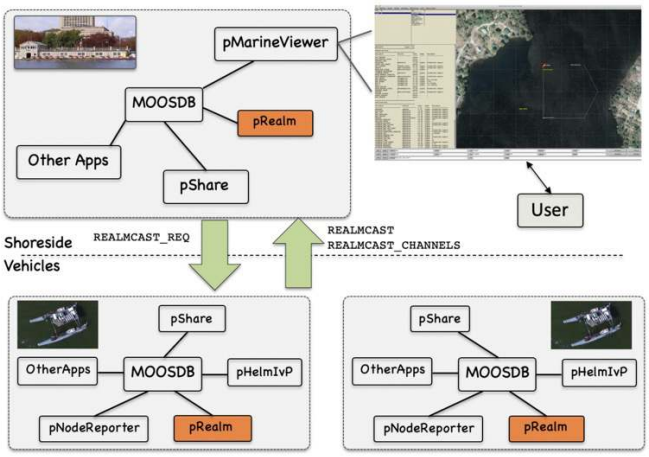
RealmCasting

RealmCasting

- Powerful new tool for *debugging* and *mission monitoring*
- Similar to AppCasting: on-demand info generation

Using RealmCasting

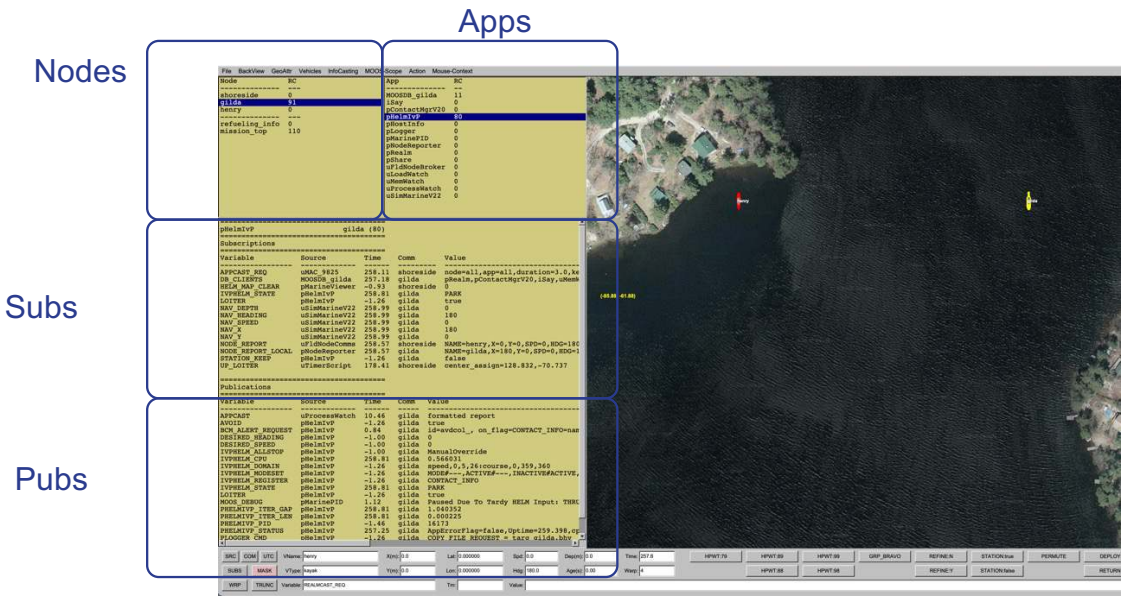
- Add pRealm to each MOOS community
- No configuration block required.



Michael Benjamin, mikerb@mit.edu

18

RealmCasting



Notable Ease-Of-Use Improvements in 20.8

AppLogging

- Any AppCasting MOOSApp now supports AppLogging
- By setting **app_logging=true**, all std output, per iteration, is posted to the **APP_LOG** variable
- Or stdout can be logged to a dedicated file with **app_logging=file**
- By saving to the **APP_LOG** variable, stdout information can be re-played in alogview.

Quick Access to documentation

- On any computer where MOOS-IvP is installed, the documentation web page can be summoned from the command-line:

```
$ pHelmIvP --web   or -w
```

```
$ pContactMgr --web or -w
```

- A new web-browser window will be opened at the documentation page for the App

Current Activities for Upcoming Releases

Dynamic Datums

- Migration of Apps, including the Helm to only communicate coordinates using Lat/Lon Global cords
- Individual apps (and behaviors) will still reason in local coordinates
- The Datum, currently fixed for the entire mission, will be fluid during the course of the mission.

- **Support for UMAA and DDS Interfaces**
- **Support for multiple simultaneous command and control Interfaces**
- **Full migration to git**
- **Core Missions for Automated V&V**
- **Support for V&V and Competitions via Containers**

Michael Benjamin, mikerb@mit.edu

21

END

Michael Benjamin, mikerb@mit.edu

22