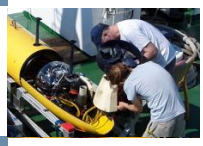
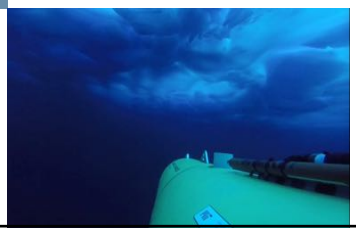




MOOS-DAWG'17

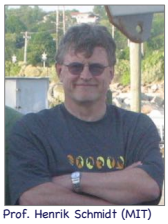
August 1st, 2017

Michael R. Benjamin
Henrik Schmidt
MIT Dept. of Mechanical Engineering
Laboratory for Autonomous Marine Sensing Systems
Computer Science and Artificial Intelligence Lab

mikerb@mit.edu
henrik@mit.edu

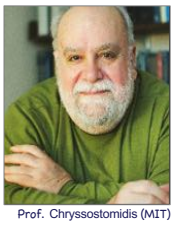
Battelle
DARPA
TTO
Thank You!!
THALES
UK
AMS
AUTONOMOUS MARINE SYSTEMS
LOCKHEED MARTIN

Collaborators


Prof. Henrik Schmidt (MIT)

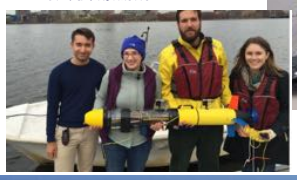

Prof. John Leonard (MIT)


Prof. Paul Newman (Oxford)

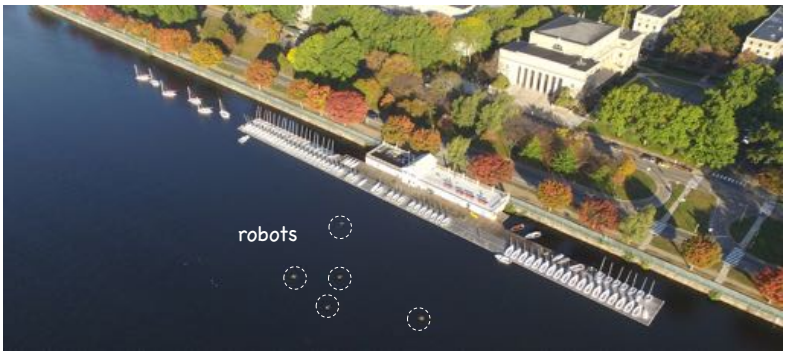

Prof. Chryssostomidis (MIT)


Dr. Michael Novitzky


Dr. Paul Robinette



The MIT Marine Autonomy Bay



- Laboratory for Autonomous Marine Sensing Systems (MECHE)
  henrik@mit.edu

- Marine Robotics Group (CSAIL)
  jleonard@mit.edu

- The AUV Laboratory (MIT Sea Grant)
  mistetri@mit.edu

Our Awesome Summer 2017 Interns

(high school and undergrad)

Hugh Dougherty
hughrrdougherty@gmail.com
Rising Freshman, UMASS Amherst

Danielle Gleason
dgleason@mit.edu
Rising Sophomore at MIT

Mayank Mali
emykion@gmail.com
Rising Freshman, U of Wisconsin, Madison

Oliver MacNeely
oliverm@mit.edu
Rising Freshman at MIT

Conlan Cesar
conlanc@gmail.com
Rising Junior at Melrose High School

Arjun Gupta,
argupta@mit.edu
Rising Sophomore MIT

Abbie Lee
abbielee@mit.edu
Rising Sophomore MIT

Carter Fendley
carter.fendley@gmail.com
Rising Penn State Freshman

Sebastian Carpenter
sebcarp@mit.edu
Rising Freshman, Carnegie Mellon

MIT Marine Robotic Platforms

Two Bluefin 21-inch UUVs (Macrura and Unicorn)

MIT Marine Robotic Platforms

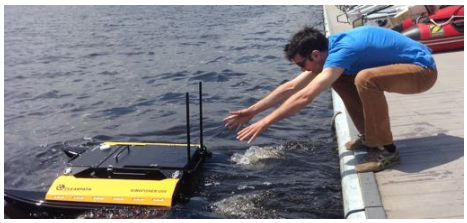
The WAM-V Unmanned Surface Vehicle

MIT Marine Robotic Platforms

The Bluefin SandShark One-Person Portable UUV

MIT Marine Robotic Platforms

The Clearpath Robotics Heron USV



Robot Architectures: Ground, Air and Sea

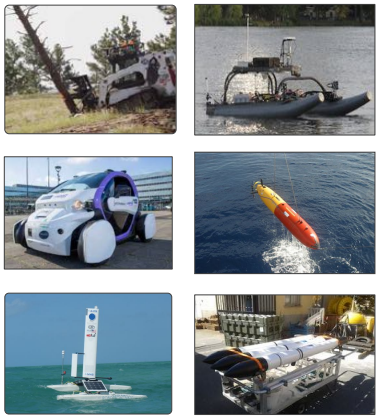
Robot Software

(What's Inside?)

Robot Architectures: Ground, Air and Sea

Proprietary Autonomy

(50 work years
My estimate)

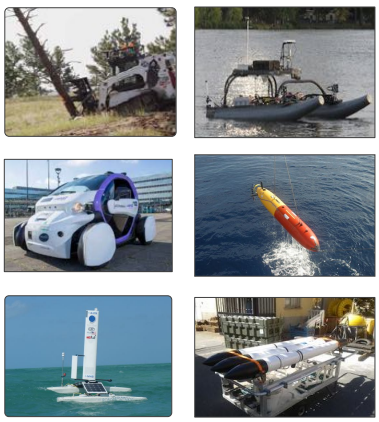


Robot Architectures: Ground, Air and Sea

Proprietary Autonomy

(30 work years)

GNU/Linux



Robot Architectures: Ground, Air and Sea

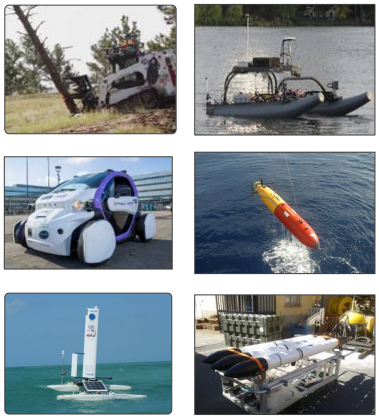
Proprietary Autonomy

(10 work years)

Component Software

Robot Middleware

GNU/Linux



Robot Architectures: Ground, Air and Sea

Proprietary Autonomy
(~0 Work Years)

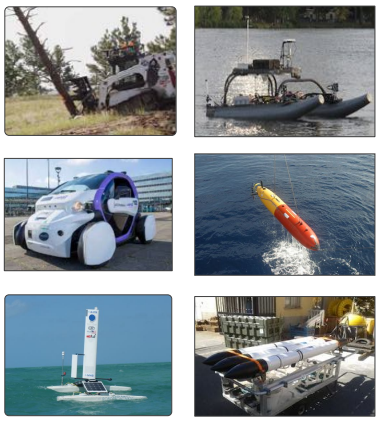
Component Autonomy

Robot Autonomyware

Component Software

Robot Middleware

GNU/Linux



Robot Architectures: Ground, Air and Sea

Proprietary Autonomy
(~0 Work Years)

MOOS-IvP
Open Robot Autonomy

(35 Work Years)
(www.moos-ivp.org)

GNU/Linux






Robot Architectures: Ground, Air and Sea

Proprietary Autonomy
(~0 Work Years)

MOOS-IvP
Open Robot Autonomy

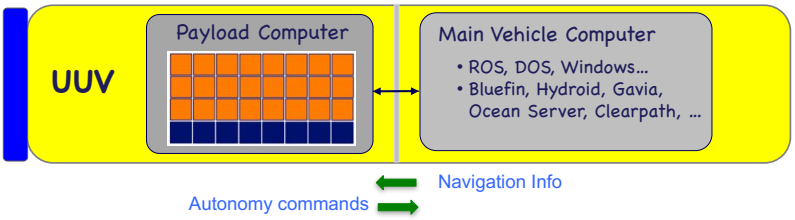
(35 Work Years)
(www.moos-ivp.org)

GNU/Linux

Lab Overview | Robot Platforms | Robot Software | Three Architectures | Software Ecosystem | Projects MOOS-IvP | Release 17.7
Michael Benjamin, Henrik Schmidt
MIT Dept of Mechanical Engineering

Payload UUV Autonomy

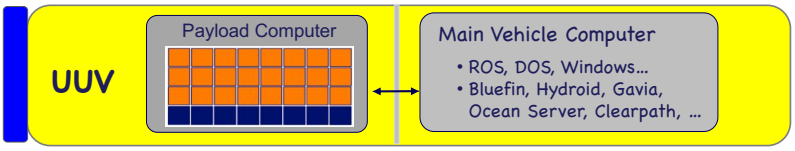
(Architecture Principle #1: Payload Autonomy)

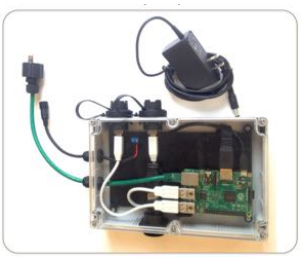


Payload Autonomy

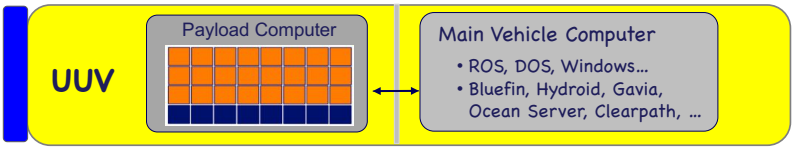
(Architecture Principle #1: Payload Autonomy)

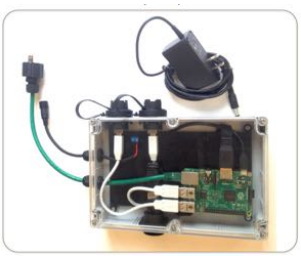
Payload Autonomy
(Architecture Principle #1: Payload Autonomy)





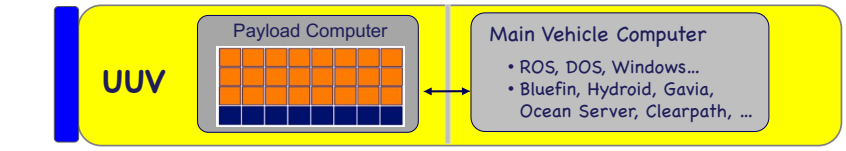
Payload Autonomy
(Architecture Principle #1: Payload Autonomy)



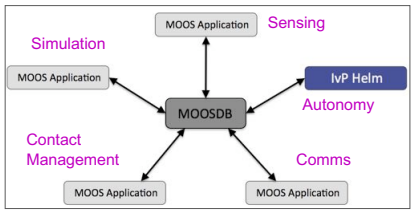


Payload Autonomy

(Architecture #2: Publish-Subscribe Middleware)



Payload Computer



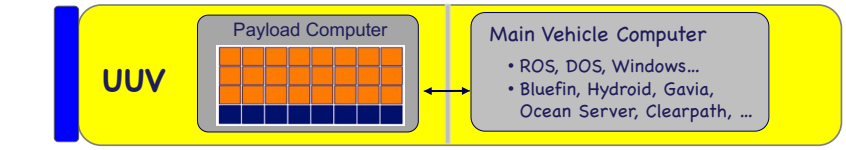
MOOS Middleware
MOOS Applications

Architecture Principle #2
Autonomy System Middleware

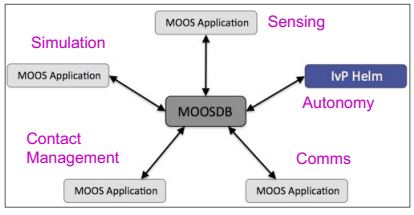
De-couple Software Procurements
Sensing, Autonomy, Simulation, Comms...

Payload Autonomy

(Architecture #2: Publish-Subscribe Middleware)

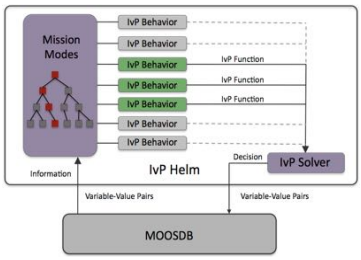


Payload Computer



MOOS Middleware
MOOS Applications

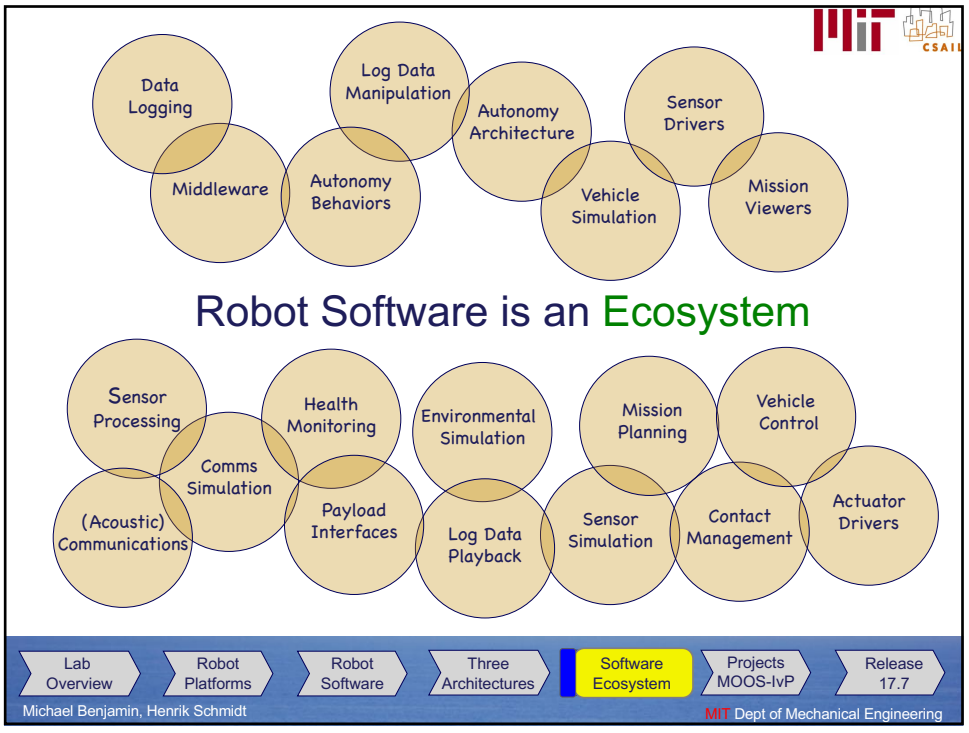
IvP Helm



Behavior-Based
Modular HELM

Autonomy Education

Robot Software is an Ecosystem



Lab Overview | Robot Platforms | Robot Software | Three Architectures | Software Ecosystem | Projects MOOS-IvP | Release 17.7
Michael Benjamin, Henrik Schmidt
MIT Dept of Mechanical Engineering

The MOOS-IvP Ecosystem

- 800+ pages of text
- Many Working examples
- Agreement w/ Cambridge Univ. Press
- 35 workyears
- 120,000 lines of code
- 20 vehicle types
- 40+ projects
- 9+ countries

Example Missions

Documentation

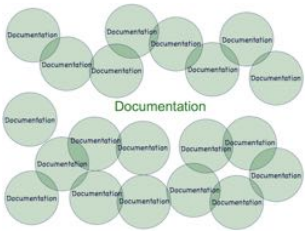
Tools
- Device/Sensor interfaces
- Vehicle Simulation
- Environmental Simulation
- Comms Simulation
- Debugging Tools
- Mission Planning Tools
- Mission Operation Tools

Autonomy Behaviors

Math and Utility Libraries

Autonomy Architecture

Robot Middleware



Evolution of the Ecosystem, Education and Research

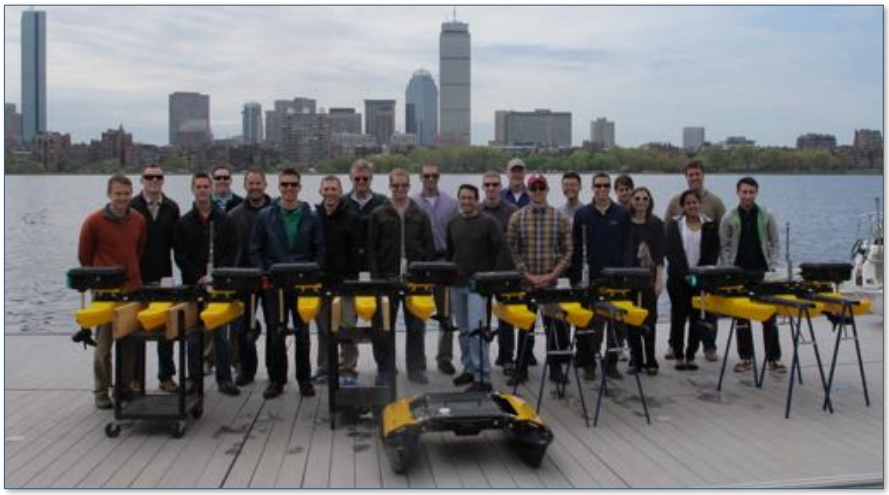
2000 — MOOS written
- Build 100% of the ecosystem yourself
- Faculty
- Post-docs

2005 — IvP Integrated with MOOS; Payload autonomy introduced
- Build 40% of ecosystem
- Learn from apprenticing
- Faculty
- Post-docs
- PhD students

2010 — First MOOS-DAWG Users workshop
- Build 20% of ecosystem
- Learn from documentation
- Learn from apprenticing
- Faculty
- Post-docs
- PhD students
- Masters students

2017 — First offering of MIT 2.680
- Build 5% of ecosystem
- Learn from coursework
- Learn from documentation
- Learn from apprenticing
- Faculty
- Post-docs
- PhD students
- Masters students
- Undergrads

MIT 2.680 Students



MIT 2.680 Students

(May 16th 2017)

MOOS-IvP Open Source Marine Robotics Community (MOOS-DAWG)

moos-dawg.org

MOOS-IvP An Open Source Project

- MOOS-IvP has been online since 2006
- Initially available only via checkout of the development head
- First tagged release, Release 4.2, July 17th 2011
- Latest release, Release 17.7, July 31st, 2017

- Applications available under GPL license
- App and behavior libraries under LGPL licence

Sloccount:
- MOOS ~30K lines of code, 6 workyears
- IvP ~122K lines of code, 32 workyears

Autonomy Research Focus Areas

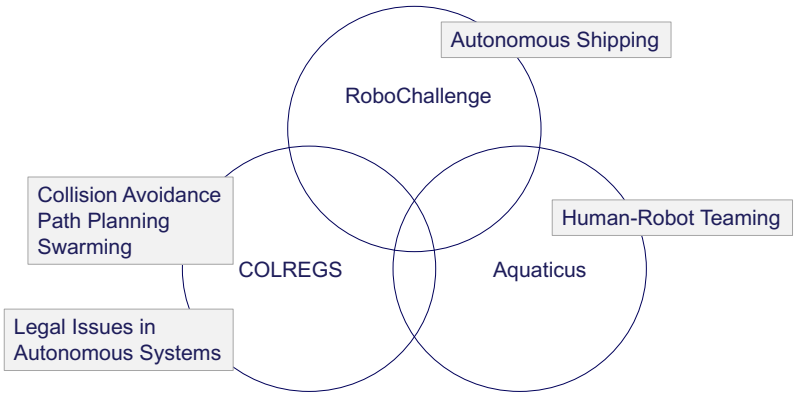
- Aquaticus
- RoboChallenge
- COLREGS
- Training/Documentation/V&V
- MOOS-IvP

Autonomy Research Focus Areas

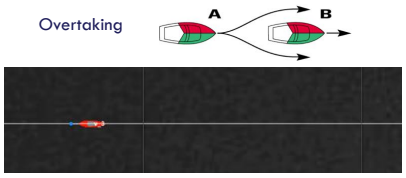
DARPA ONR Battelle Lockheed

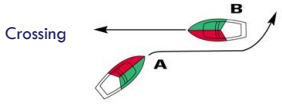
- Aquaticus
- RoboChallenge
- COLREGS — ONR THALES Battelle
- Education/Documentation/V&V — Battelle
- MOOS-IvP — ONR Battelle

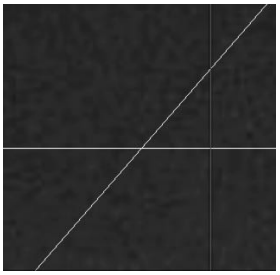
Autonomy Research Focus Areas

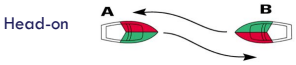


COLREGS Autonomy







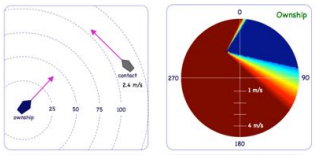


From: Joseph Leavitt, “Intent-Aware Collision Avoidance”, Masters Thesis, Massachusetts Institute of Technology, June 2017.

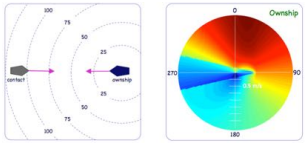
COLREGS Autonomy

- Funded by: Office of Naval Research (ONR)
- Idea:
  - Enable autonomous surface vehicles to obey the "Rules of the Road" COLREGS.
  - Establish a road test for validating the autonomous collision avoidance.
- Research Focus:
  - Map the protocols written for humans into algorithmic format.
  - Find minimal set of field tests that validate widest set of scenario permutations.

Crossing Situation



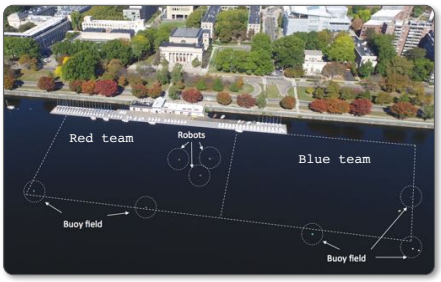
Head-on Situation

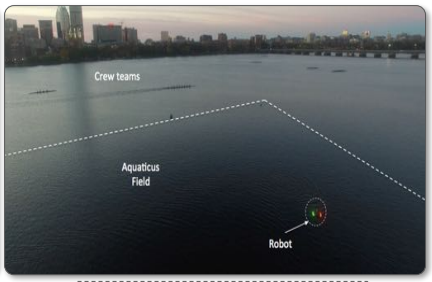


- Technical Approach:
  - Collision avoidance protocols mapped to set of modes, and submodes.
  - Modes map to a unique form of objective function. Multi-objective optimization with IvP to solve.
- Impact:
  - Autonomous long-duration coastal sampling with autonomous platforms.
  - Multi-vehicle/swarm capabilities can be built on COLREGS foundation.

What is Aquaticus

- Aquaticus is a human-robot competition developed at MIT on the Charles River.
- It pits teams of humans and robots against other teams of humans and robots.
- It explores advanced marine *autonomy*, *human-robot trust*, *operator load* and the interface between robots and humans embedded in the field with robots.

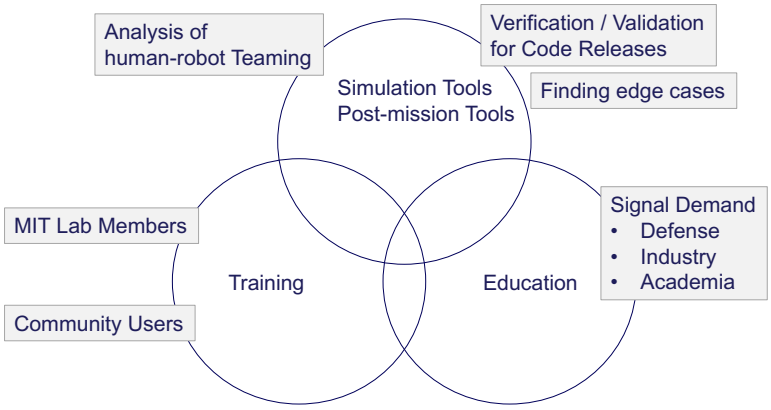




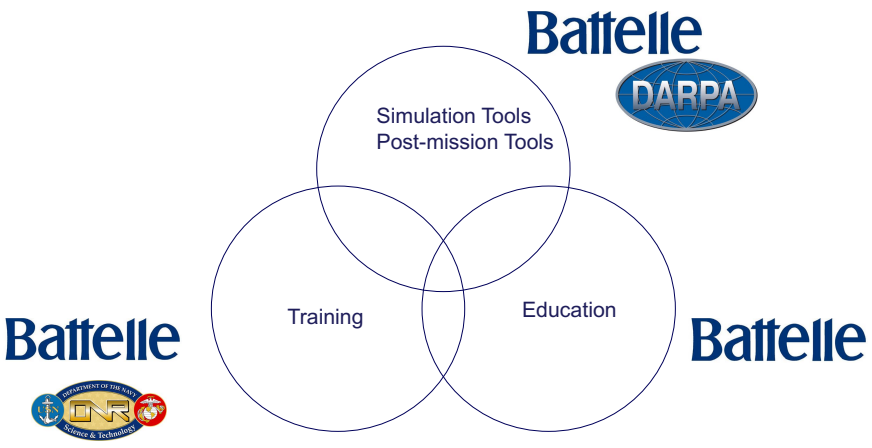
Boston Harbor Robo-Challenge

The Boston Harbor Robo-Challenge: Remote Ocean Sensing from MIT

Non-Autonomy Focus Areas



Non-Autonomy Focus Areas



Release 17.7

Release 17.7 reflects two threads

1. COLREGS
   - The COLREGS behavior (and libraries)
   - Enhanced CPAEngine (faster and expanded features)

2. Large-batch simulations (evaluating Effeciency alongside Safety)
   - The uFldCollisionDetect App
   - The uFldWrapDetect App
   - The pEvalLoiterApp
   - Encounter Plots library and integration into alogview
   - The alogcd utility
   - The alogpare utility
   - The alogeplot utility
   - The uQueryDB App for conditionally interrupting a simulation

Release 17.7.X Plans

Planned focus areas in the short term:

1. COLREGS
   - Further testing, further refining, in swarm scenarios
   - Further in-water testing
   - Incorporation of better vehicle model (advance/transfer properties)
2. Helm Efficiencies
   - Lazy behavior evaluation (allowing behaviors to set their own AppTick, based on it's own determination of circumstances)
3. Post-mission analysis
   - Alogview integration with video and audio playback
4. Verification and Validation
   - Automatic build testing
   - Automatic mission testing

END