CARACaS multi-agent maritime autonomy for USVs in the Swarm II harbor patrol demonstration

Michael Wolf, Amir Rahmani, Jean-Pierre de la Croix, Gail Woodward, Joshua Vander Hook, David Brown, Steve Schaffer, Christopher Lim, Philip Bailey, Scott Tepsuporn, Marc Pomerantz, Viet Nguyen, Cristina Sorice, and Michael Sandoval

Jet Propulsion Laboratory
California Institute of Technology

© 2017 California Institute of Technology. U.S. Government sponsorship acknowledged. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the Office of Naval Research through an agreement with the National Aeronautics and Space Administration.

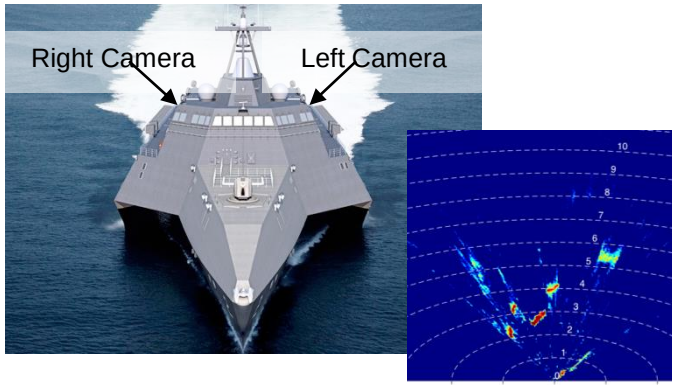
Current Projects in Maritime Autonomy

ASW Continuous Trail Unmanned Vehicle (ACTUV)

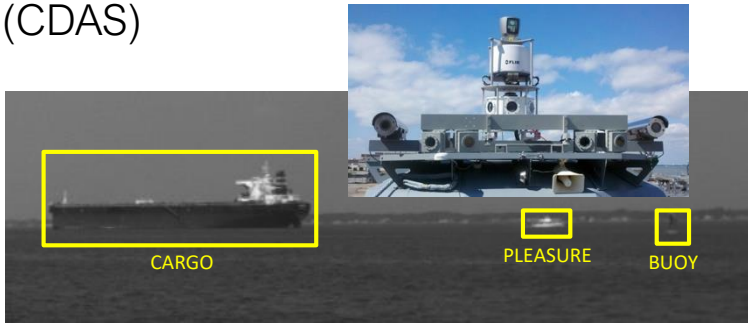
USV Swarm

Unmanned Influence Sweep System (UISS)

Mechanically Uncoupled Stereo EO/IR (MUSE)



Contact Detection and Analysis System (CDAS)



CARACaS (Control Architecture for Robotic Agent Command and Sensing)

A multi-mission, multi-agent autonomy development platform

Open Architecture

Standards based

Fee free licensing for government use

CARACaS Autonomy Architecture establishes module decomposition and provides execution engines

CARACaS Software Framework (CSF) libraries/services used by developers in creating modules

CARACaS Autonomy Toolkits for testing, executing, and analyzing the software

CARACaS Autonomy Architecture

Autonomy Application
- Command Dictionaries
- Mission Models & Behaviors
- Customized Motion Planning
- Hardware Interfaces

Autonomy Modules
- Mission Executive
- Behavior Engine(s)
- Planning Libraries
- Perception & Control Modules

Architecture Definition
- Module Decomposition
- World Model Specification
- Interface Control Documents

CARACaS Autonomy Toolkits

Operator Toolkit
- Command Interface
- Mission Model Composer
- Mission Visualization
- Software Process Mgmt

Developer Toolkit
- Simulation Engine
- Introspection Tools
- Algorithm Visualization
- Module Replay

Architectural Services
- Health Monitoring
- Parameter Management
- Data Logging & Playback

Middleware
- Data-Centric Communications
- Synchronous Messaging
- Command & Data Handling

CARACaS Software Framework

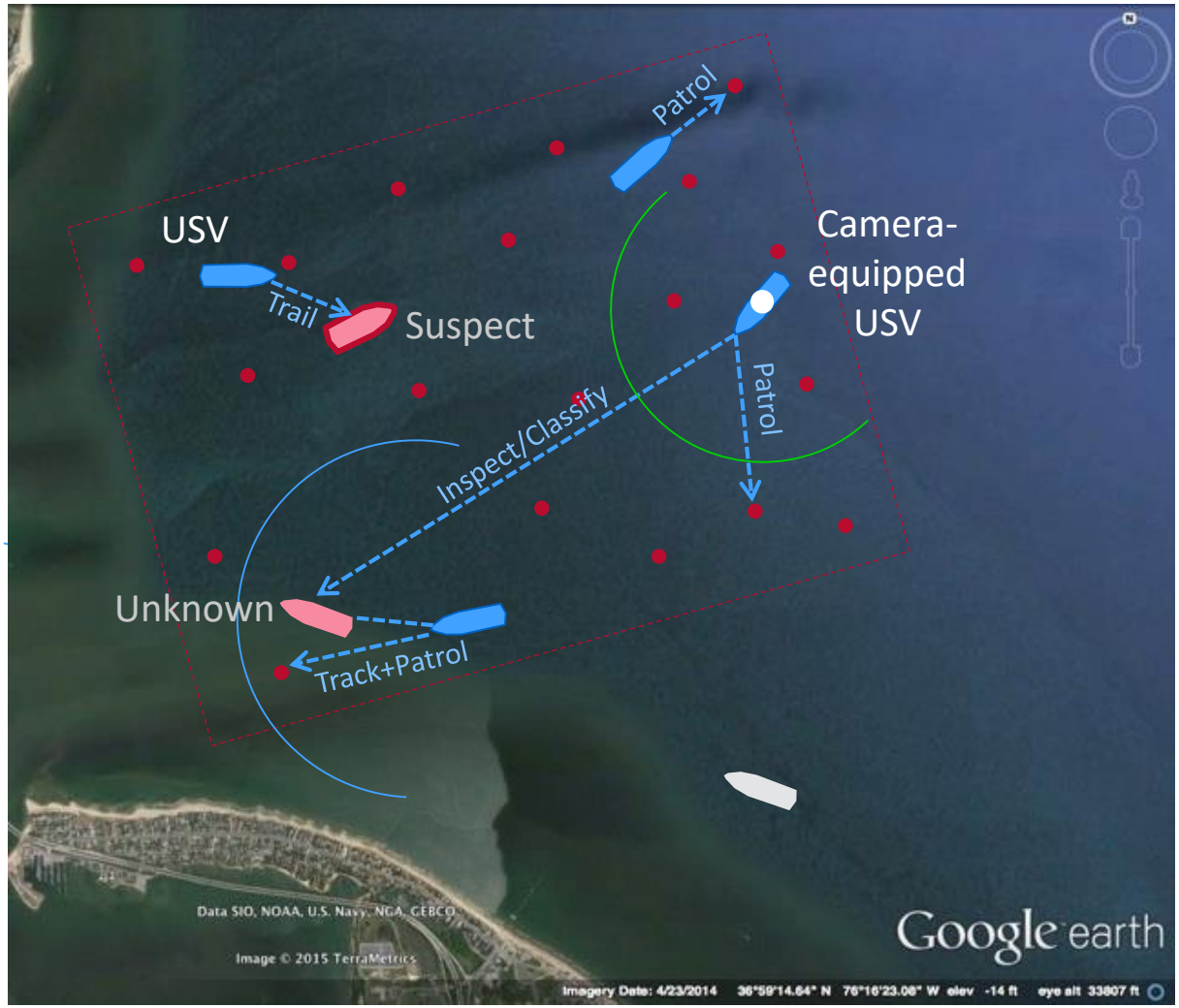
USV Swarm 2 Demonstration (SEP 2016)

Behaviors

- **Patrol** – cooperatively monitor a defined area
- **Track** – keep a vessel in radar range, while patrolling nearby
- **Inspect** – dispatch a USV with cameras to classify a vessel
- **Trail** – closely follow a suspect vessel

Technology Objectives

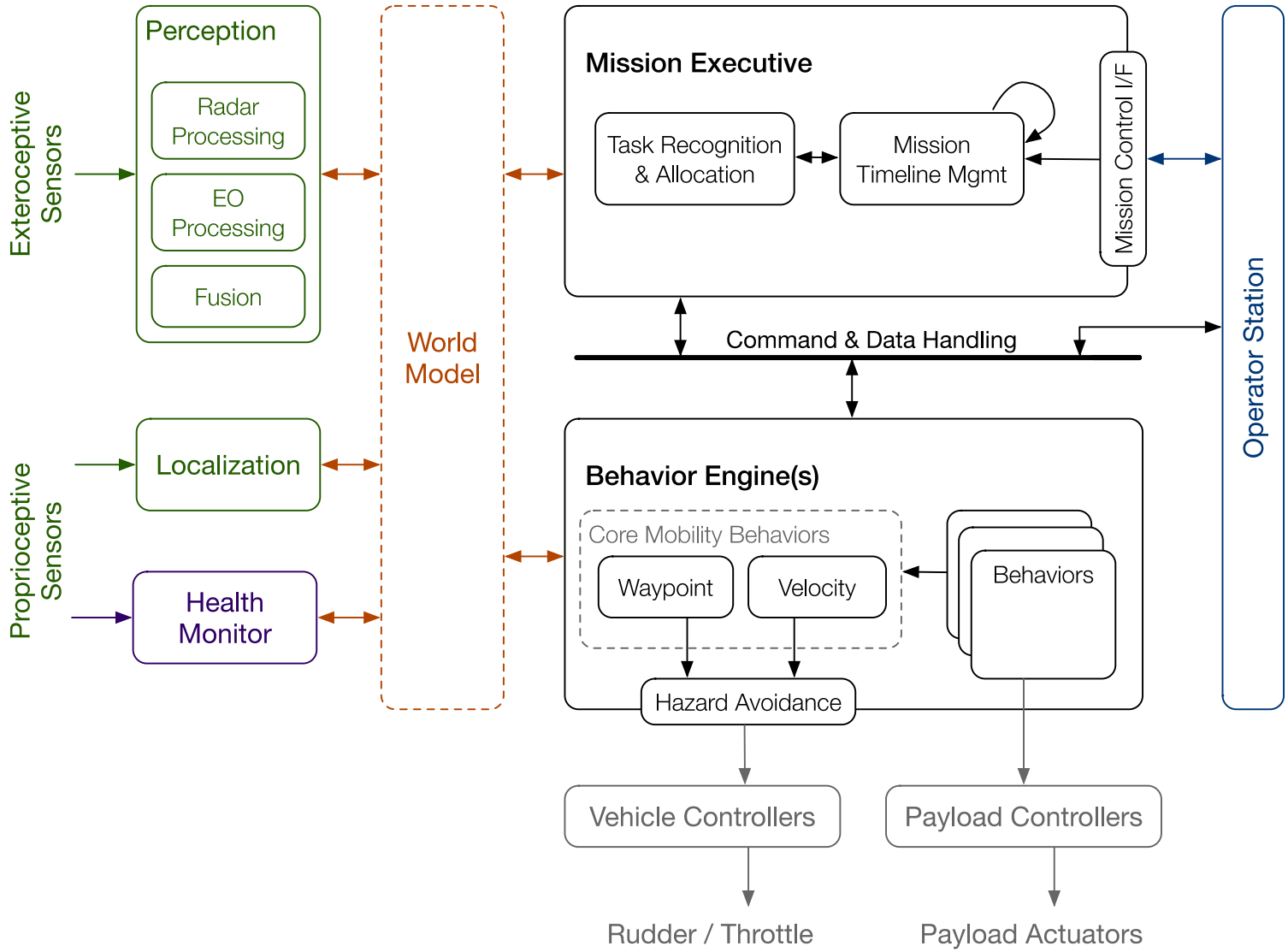
- Higher levels of autonomy / less operator input
- Autonomous task recognition and allocation
- Planning for heterogeneous teams
- Adaptation to attrition and subsystem failures
- Decreased bandwidth requirements



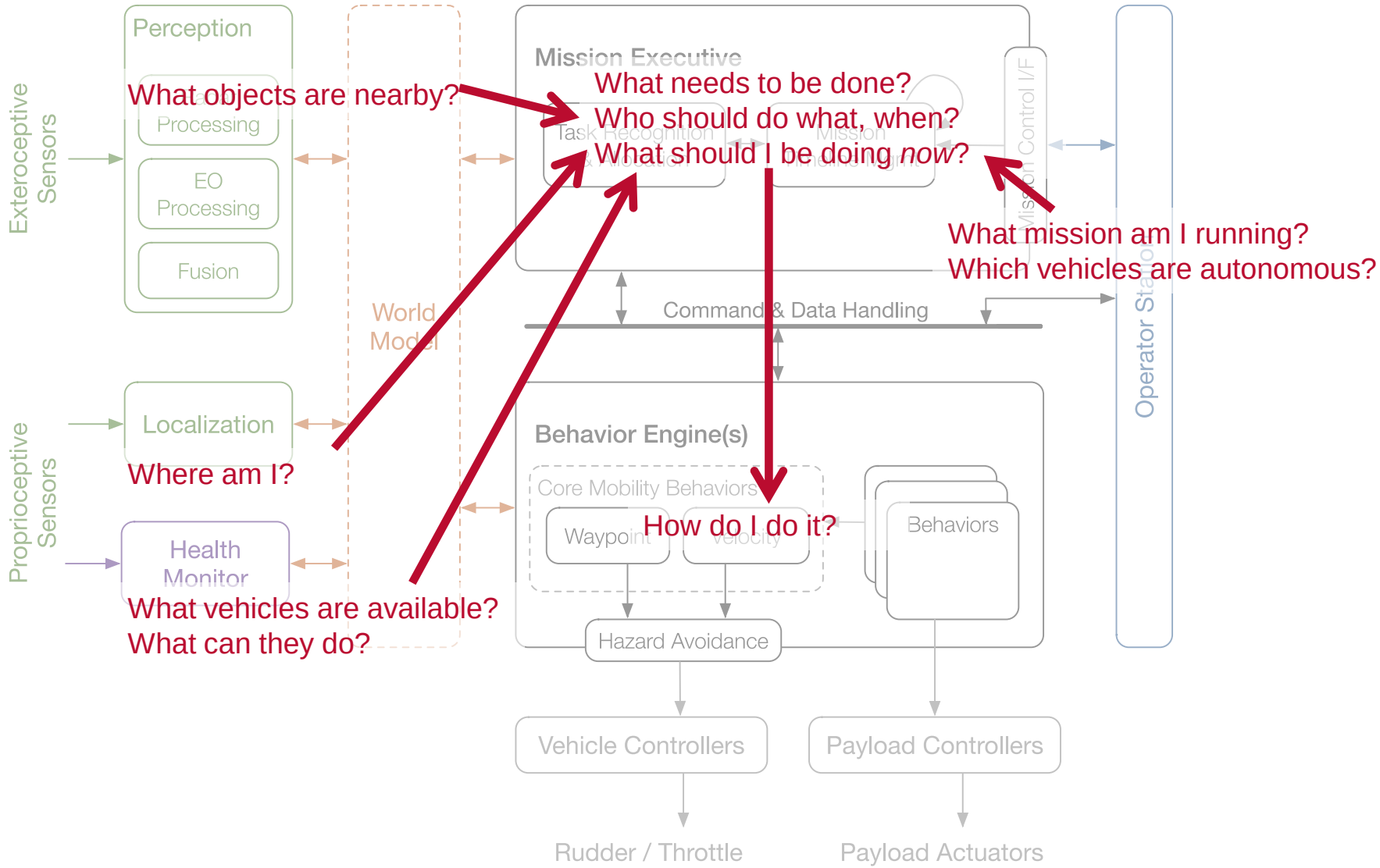
Key Ideas for Swarm 2 Autonomy

- Autonomy must be capable of event-driven operations in highly dynamic environments.
  - Not just sequenced, waypoint-driven missions
  - Swarm needs to *recognize* the tasks required to complete its mission as well as *allocate* those tasks to agents
  - Missions are complex and composed of multiple behaviors
- Focus was multi-mission autonomy architecture with customizable behaviors.
  - Not necessarily optimizing a “patrol agents” system
  - Use a specific, sufficiently complex example to define and exercise a multi-agent autonomy architecture and corresponding cooperative behaviors
- Autonomy systems should be robust and flexible
  - No software component knew any of the following beforehand:
    - how many USVs would be participating
    - how many USVs had camera systems
    - how many contacts might be encountered
  - Discovering conditions in runtime and re-planning accordingly enables both operational flexibility and robustness to health faults / changing conditions

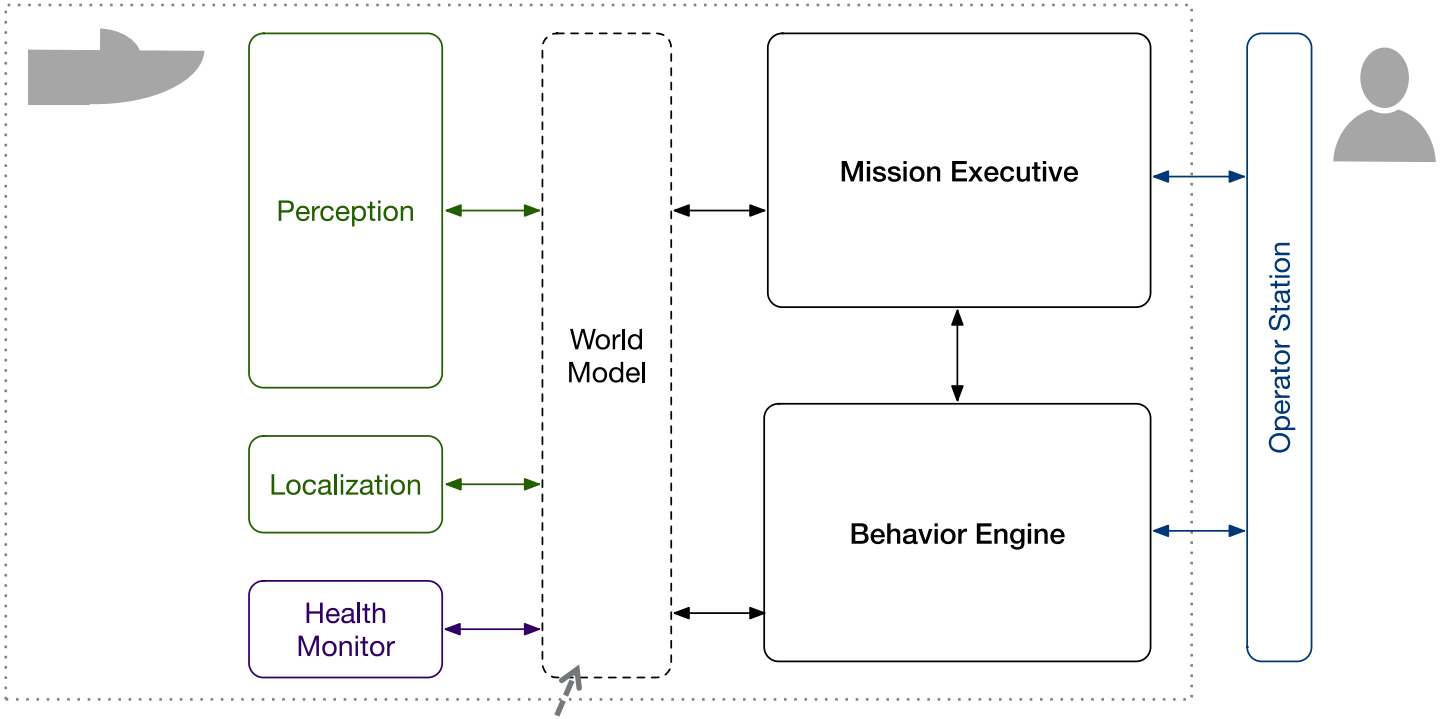
CARACaS: Functional Architecture

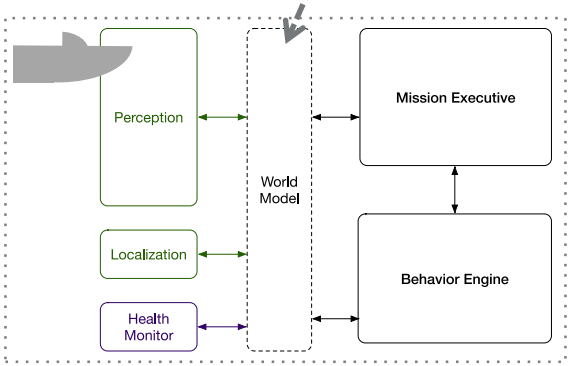


CARACaS: Functional Architecture



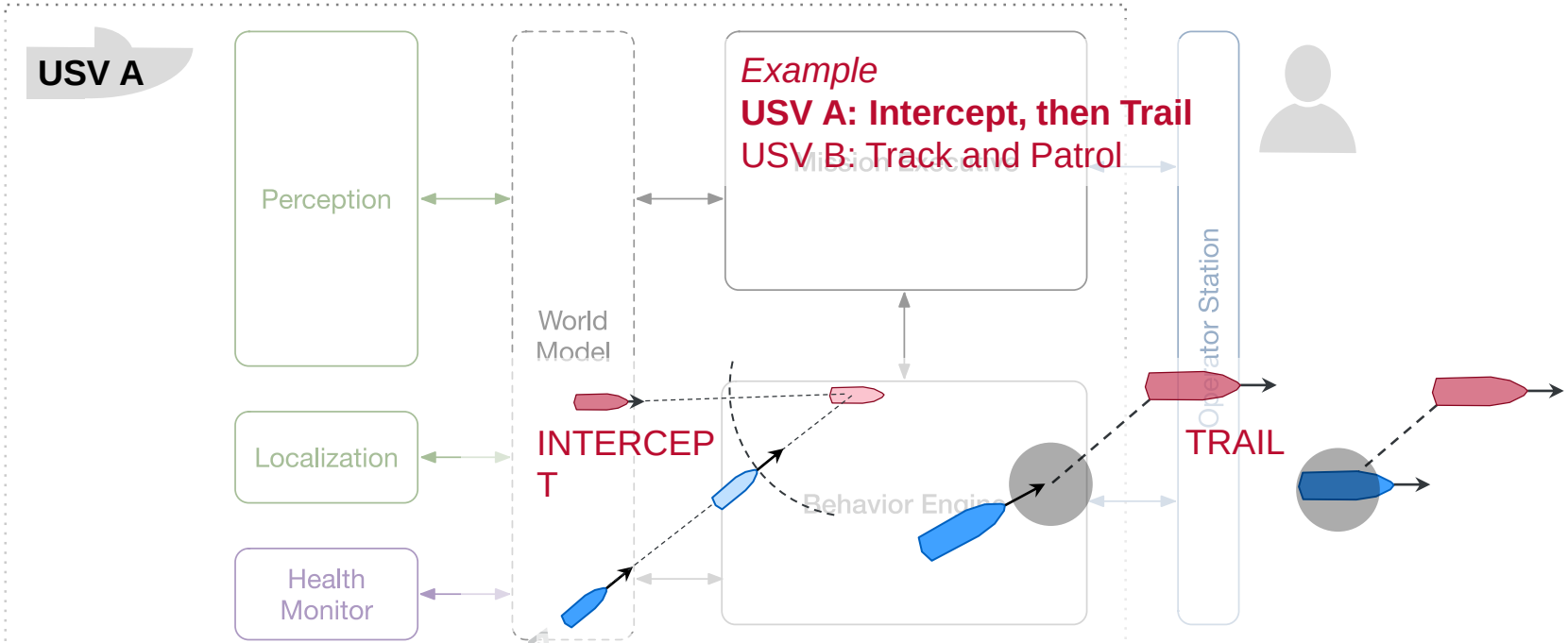
Cooperative Autonomy for USV Swarm





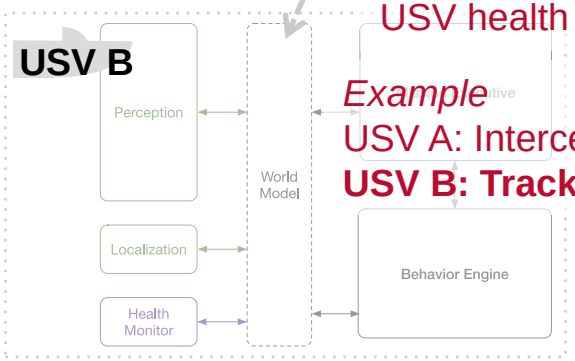
- World Models are synchronized (communications allowing).
- All Planning is decentralized.
- Mission Executive creates a timeline for itself *and* predictions of other agents' plans.
- Agents execute their own local plan; no additional communication needed.

Cooperative Autonomy for USV Swarm



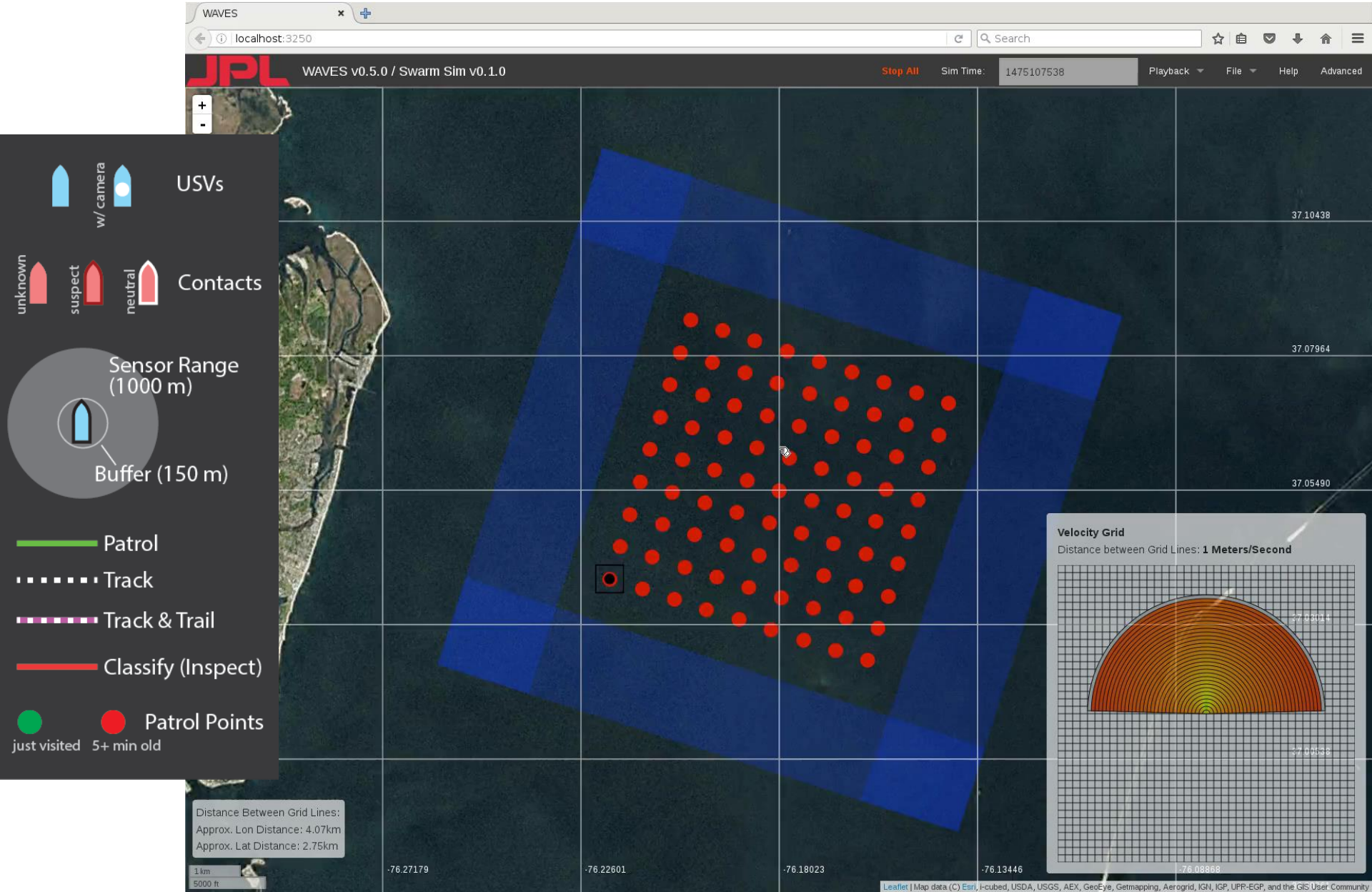
Example
USV A: Intercept, then Trail
USV B: Track and Patrol

Track list (all contacts & USVs)
Vessel classification
Patrol point visit times
USV health and status



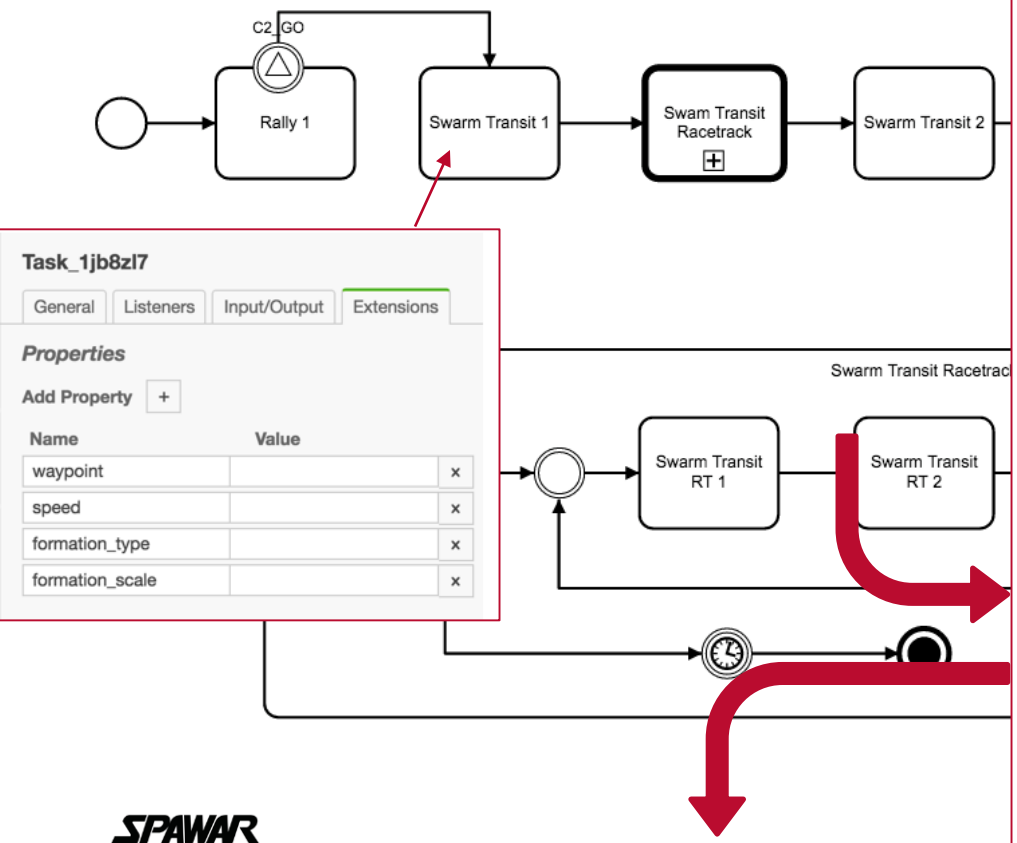
Example
USV A: Intercept, then Trail
USV B: Track and Patrol

Swarm 2 Mission Video



Human Element: Defining a Mission

CARACaS Mission Executive ingests missions in
Business Process Modeling and Notation (BPMN)

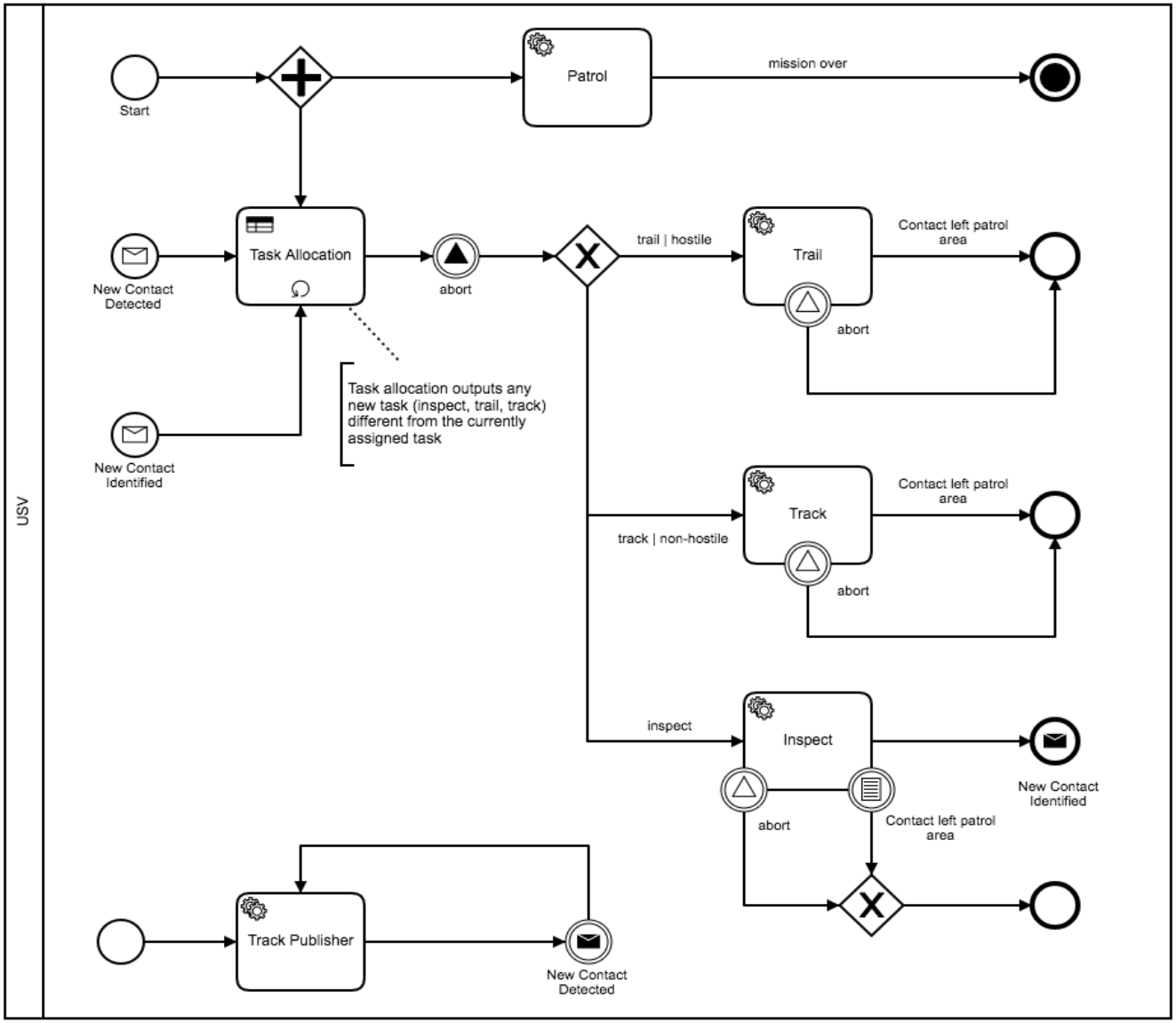



```
1.  <?xml version="1.0" encoding="UTF-8"?>
2.  <bpmn:definitions xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL" xmlns:bpmndi="http://ww
    w.omg.org/spec/BPMN/20100524/DI" xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:dc="http:
    //www.omg.org/spec/DD/20100524/DC" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:camunda="http://camunda.org/schema/1.0/bpmn" id="Definitions_1" targetNamespace="h
    ttp://bpmn.io/schema/bpmn" exporter="Camunda Modeler" exporterVersion="1.2.2">
3.    <bpmn:process id="Process_1" isExecutable="false">
4.      <bpmn:task id="Task_1qh8xc8" name="Rally 1">
5.        <bpmn:extensionElements>
6.          <camunda:properties>
7.            <camunda:property name="waypoint" value="[lat, lon]" />
8.            <camunda:property name="approach_speed" value="(default:  0.7*V_max)" />
9.            <camunda:property name="capture_radius" value="(default:  2Km)" />
10.         </camunda:properties>
11.       </bpmn:extensionElements>
12.       <bpmn:incoming>SequenceFlow_0l647uu</bpmn:incoming>
13.     </bpmn:task>
14.     <bpmn:sequenceFlow id="SequenceFlow_0l647uu" sourceRef="StartEvent_0430kc2" targetRef="Task_1q
    h8xc8" />
15.     <bpmn:task id="Task_1jb8zl7" name="Swarm Transit 1">
16.       <bpmn:extensionElements>
17.         <camunda:properties>
18.           <camunda:property name="waypoint" />
19.           <camunda:property name="speed" />
20.           <camunda:property name="formation_type" />
21.           <camunda:property name="formation_scale" />
22.         </camunda:properties>
23.       </bpmn:extensionElements>
24.       <bpmn:incoming>SequenceFlow_1o12s9v</bpmn:incoming>
25.       <bpmn:outgoing>SequenceFlow_131ezto</bpmn:outgoing>
26.     </bpmn:task>
27.     <bpmn:sequenceFlow id="SequenceFlow_1o12s9v" sourceRef="BoundaryEvent_017f9od" targetRef="Task
    _1jb8zl7" />
28.     <bpmn:sequenceFlow id="SequenceFlow_131ezto" sourceRef="Task_1jb8zl7" targetRef="CallActivity_
    0oyap06" />
29.     <bpmn:callActivity id="CallActivity_0oyap06" name="Swam Transit Racetrack">
30.       <bpmn:incoming>SequenceFlow_131ezto</bpmn:incoming>
31.       <bpmn:outgoing>SequenceFlow_03yh811</bpmn:outgoing>
32.     </bpmn:callActivity>
33.     <bpmn:subProcess id="SubProcess_0nrbbhh" name="Swarm Transit Racetrack">
34.       <bpmn:startEvent id="StartEvent_08jhtzh">
35.         <bpmn:outgoing>SequenceFlow_1tlpfxh</bpmn:outgoing>
36.       </bpmn:startEvent>
37.       <bpmn:task id="Task_0v5zecc" name="Racetrack Waypoint 1">
```

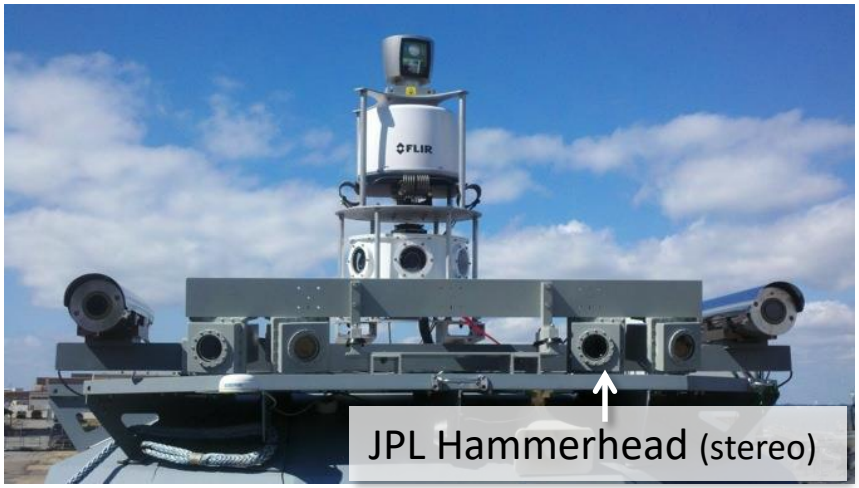
Model Verification
Interpretation by Autonomy
Mission Status Updates

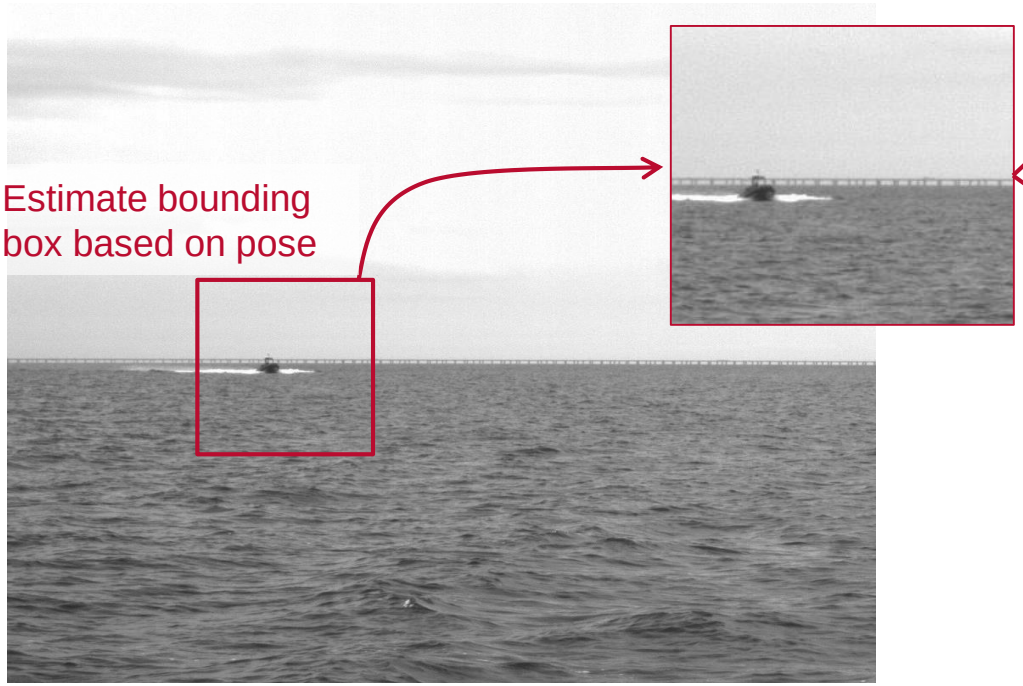
SPAWAR
Systems Center PACIFIC

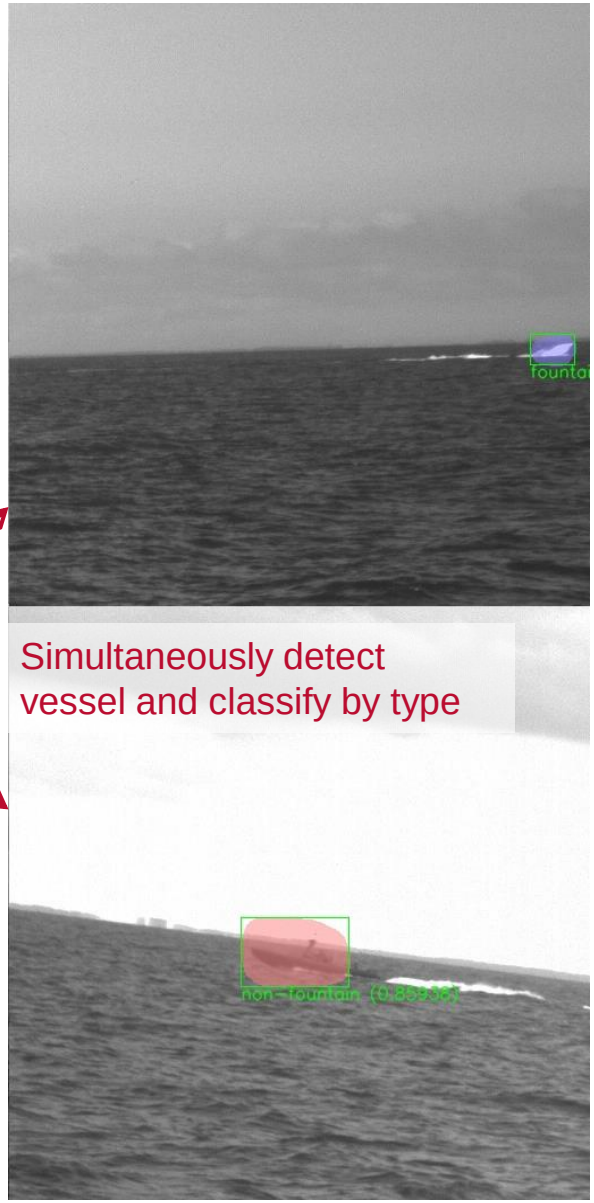
Swarm 2 Mission Model Diagram



CDAS EO Processing (Contact Detection and Analysis System)







Conclusion

- Summary
  - CARACaS provides an autonomy software framework for multi-agent coordination and control.
  - The Swarm 2 mission demonstrated performance in dynamic, event-driven missions requiring multiple behaviors and shared world modeling.
  - Autonomy has been designed for user-friendliness, robustness to failures, and flexibility for operational variables.
- Next Steps
  - Maturing situational awareness / robustness to perception noise
  - Further demonstrating “building block” approach to missions through behavior composition
  - More complex teaming scenarios (e.g., USV subgroups with closely coupled behaviors)

Acknowledgments

Office of Naval Research

Collaboration Team

Spatial Integrated Systems

DH Wagner Associates

NSWC–Carderock, Combatant Craft Division