

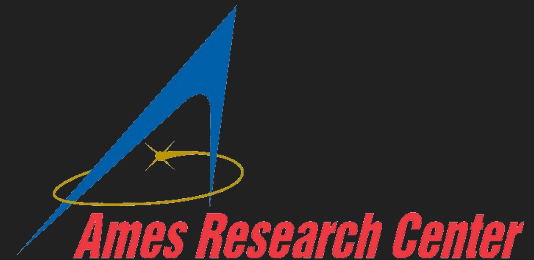
PLUMMRS: A Collection of Plan Ledgers and Unified Maps for Multi-Robot Safety

Astrobee Working Group,
November 16th, 2023

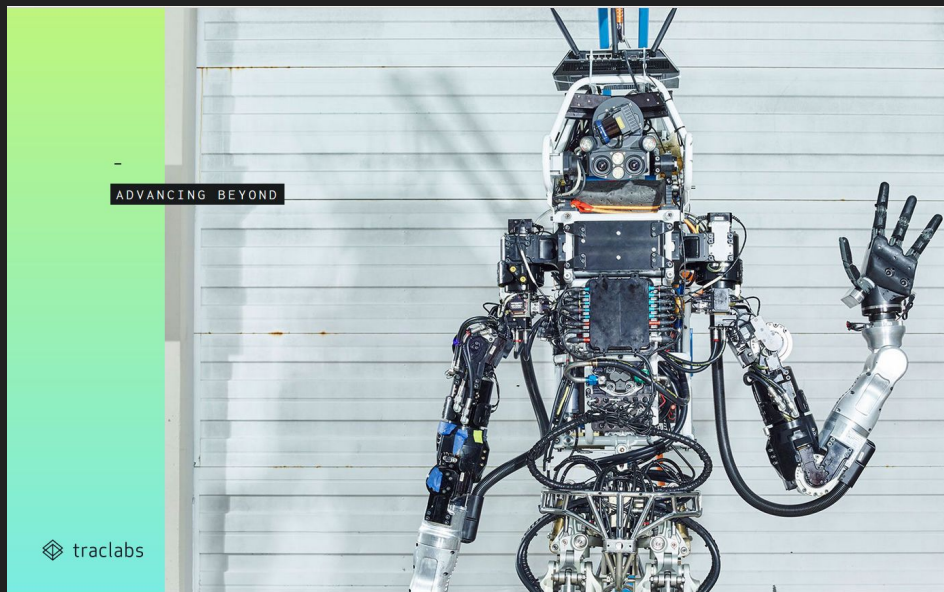
Ana Huamán Quispe
COR: José V. Benavides



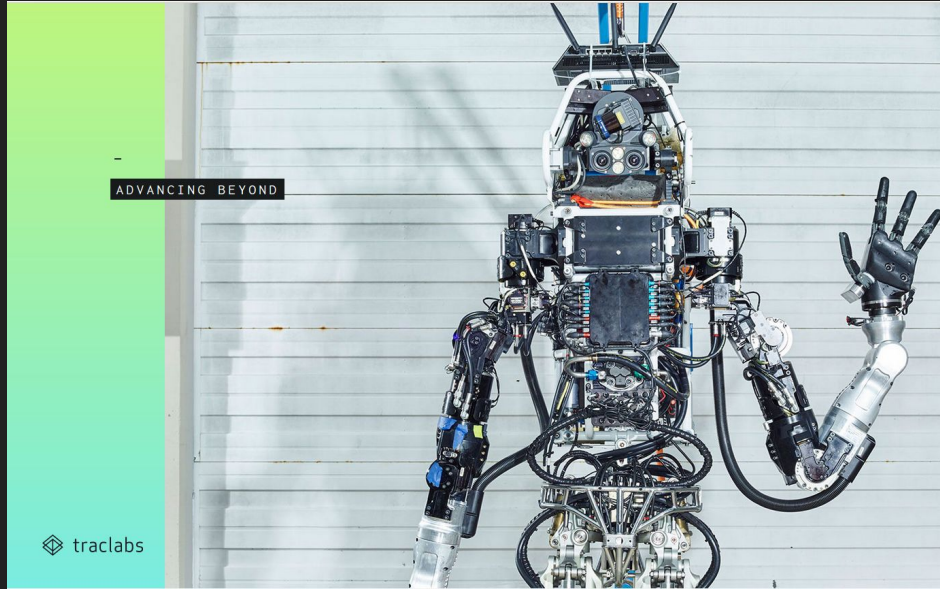
NASA SBIR 80NSSC20C0611



TRAC Labs



TRAC Labs



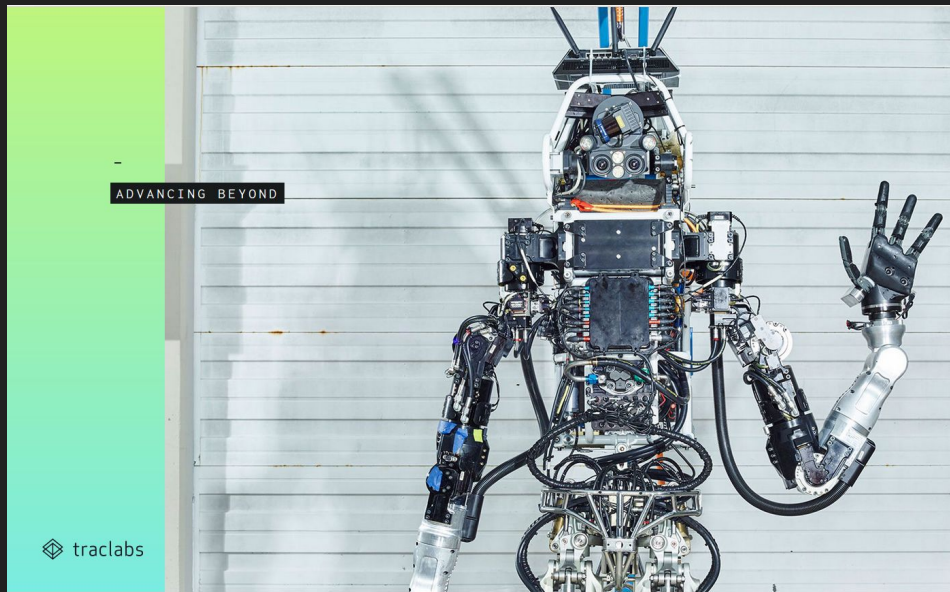
Quick facts:

- Established in 1997.
- Located in Webster, Texas.

Purpose:

Conduct world-class research in Robotics, AI and Automation, with the mission to develop new technologies and products for industry and emerging markets.

TRAC Labs



Decision Support:

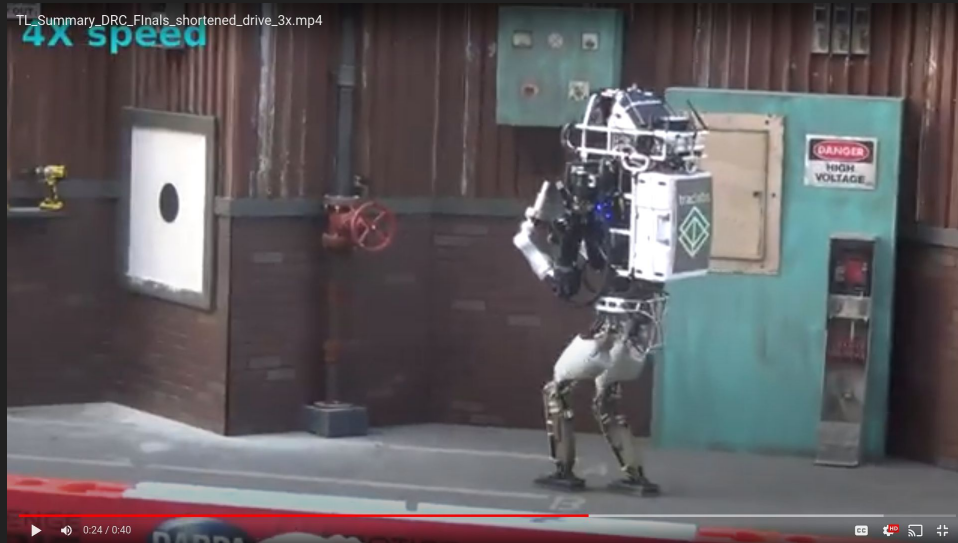
- Task execution assistance.
- System modeling
- Task planning.
- Data summarization.

Robotics:

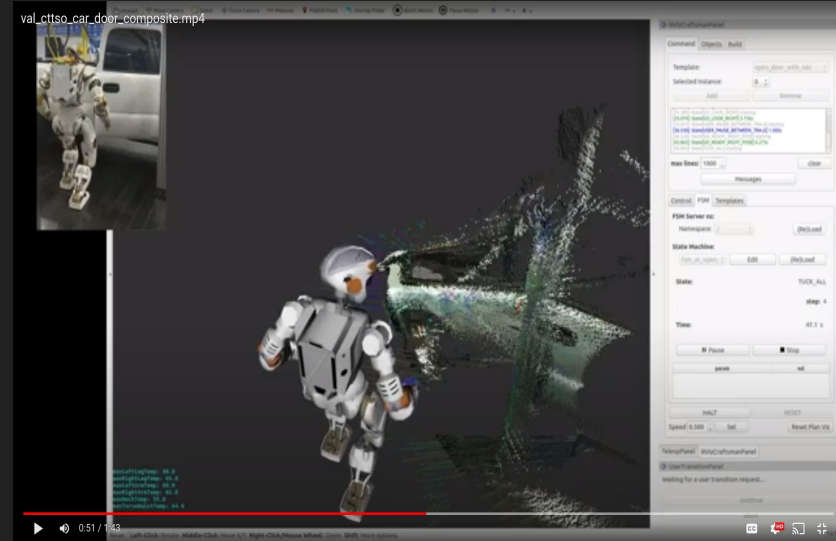
- Dexterous manipulation.
- Perception.
- Mobile robot navigation.
- Human-Machine interfaces.

TRAC Labs

Darpa Robotics Challenge - 2015



CTTSO Demo - 2019



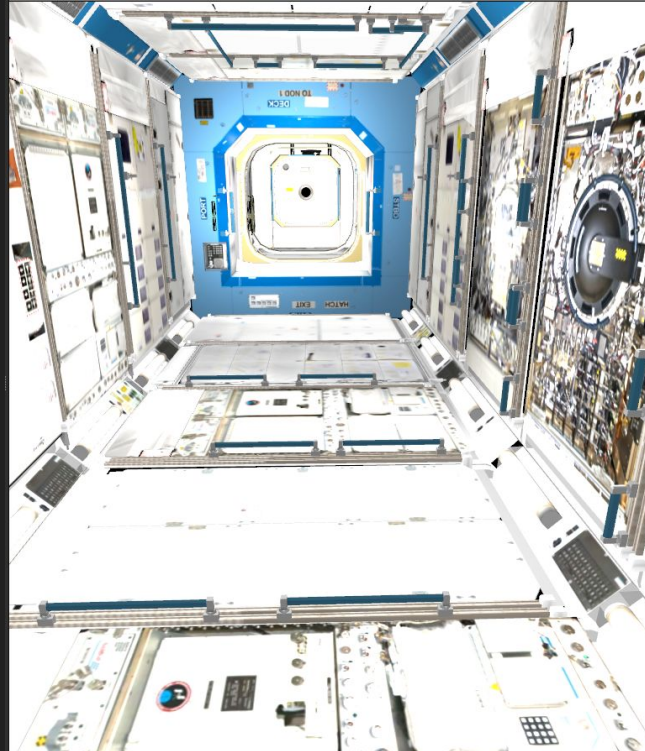
Industrial automation work, mostly with robot arms

Summary

1. Project motivation
2. Project description
3. Hardware validation

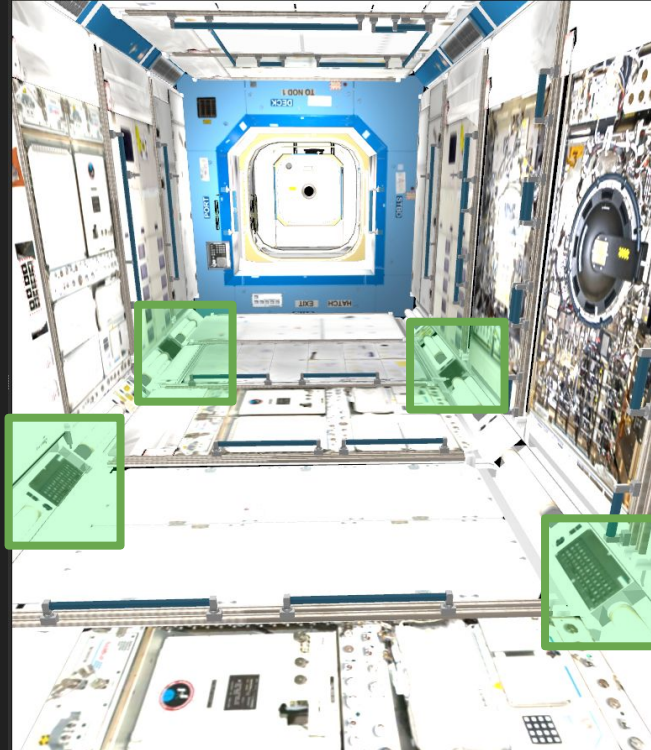
1. Motivation

Somewhere far, far away...



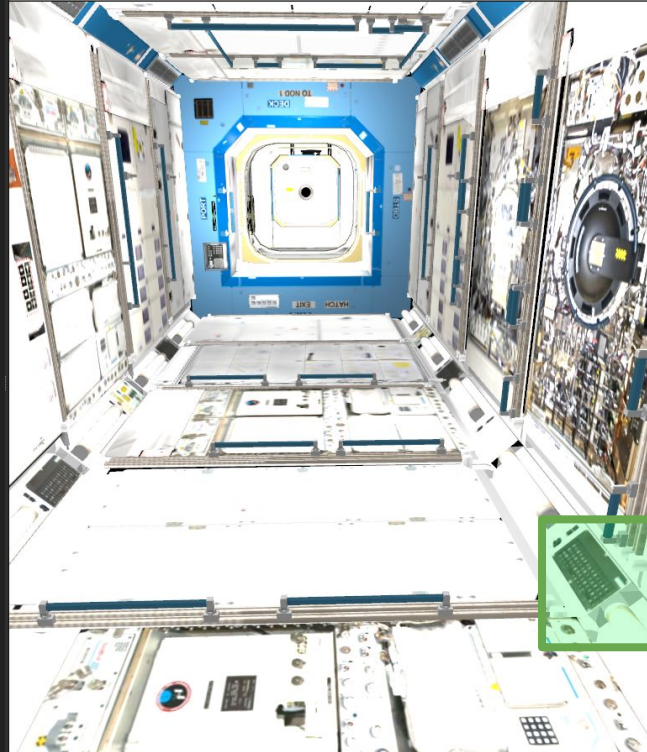
1. Motivation

Inspect air vents



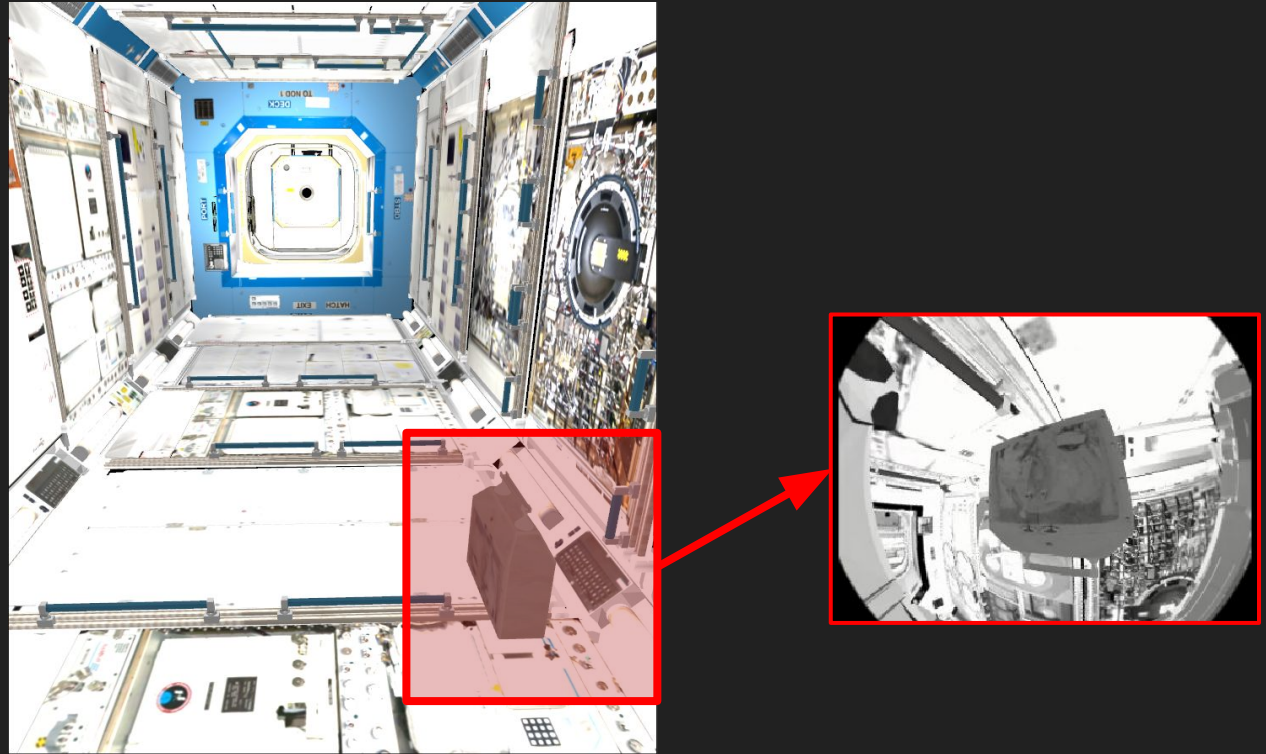
1. Motivation

Ideally, clear of obstructions



1. Motivation

If an object is detected...

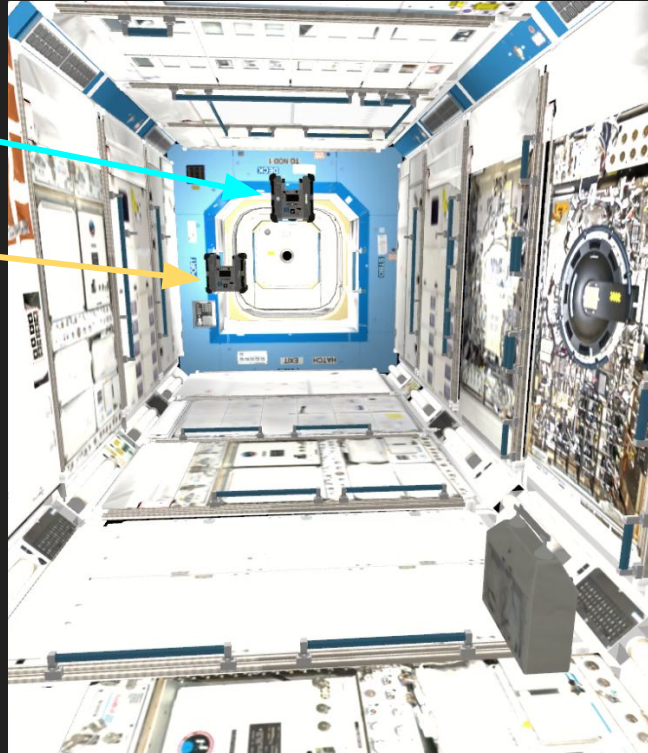


1. Motivation

Look who we got!

Bumble

Honey

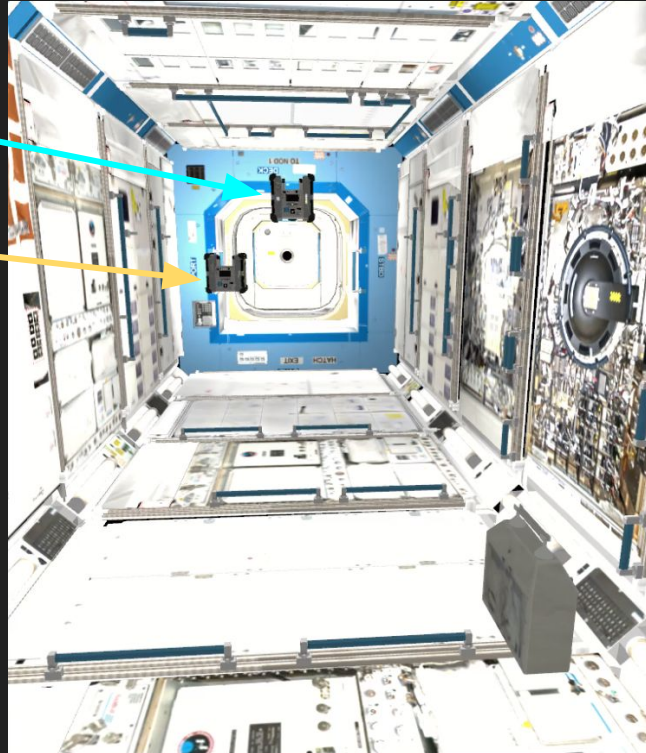


1. Motivation

Inspect Task: Divide and conquer

Bumble

Honey



Bumble:

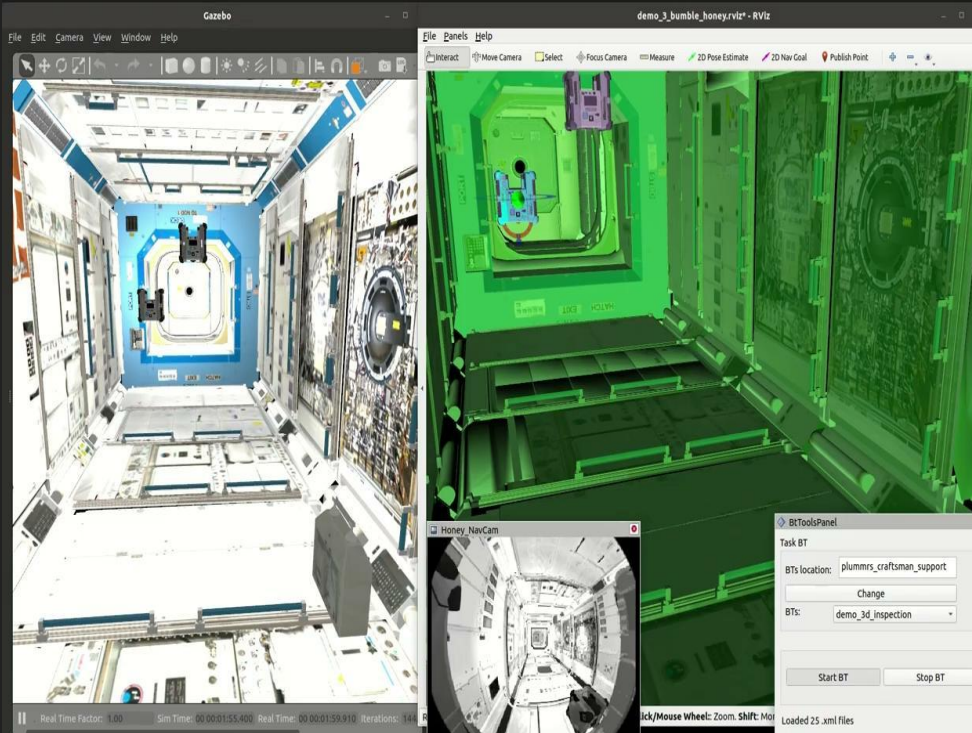
Navigates and visually
inspects each air vent.

Honey:

If an object is detected,
go and move it out of
the way.

1. Motivation

Inspect Task: Divide and conquer



Bumble:

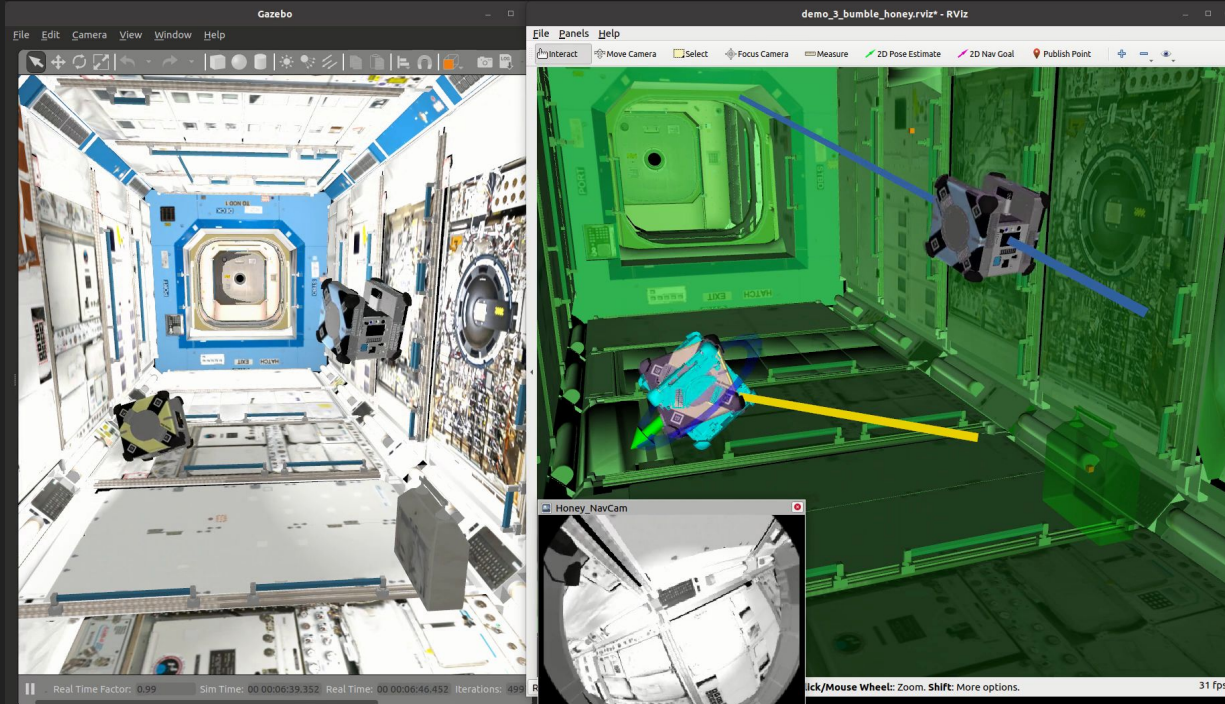
Navigates and visually
Inspects each air vent.

Honey:

If an object is detected,
go and move it out of
the way.

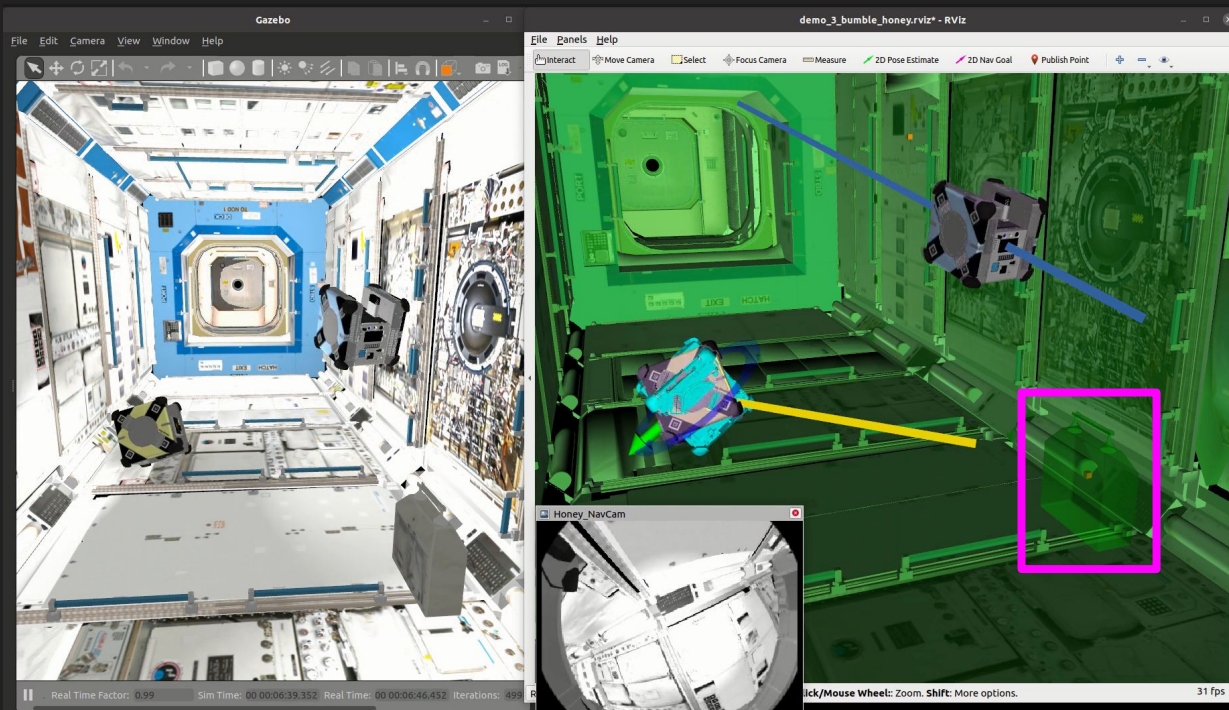
1. Motivation

What do we need to accomplish this?



1. Motivation

What do we need to accomplish this?

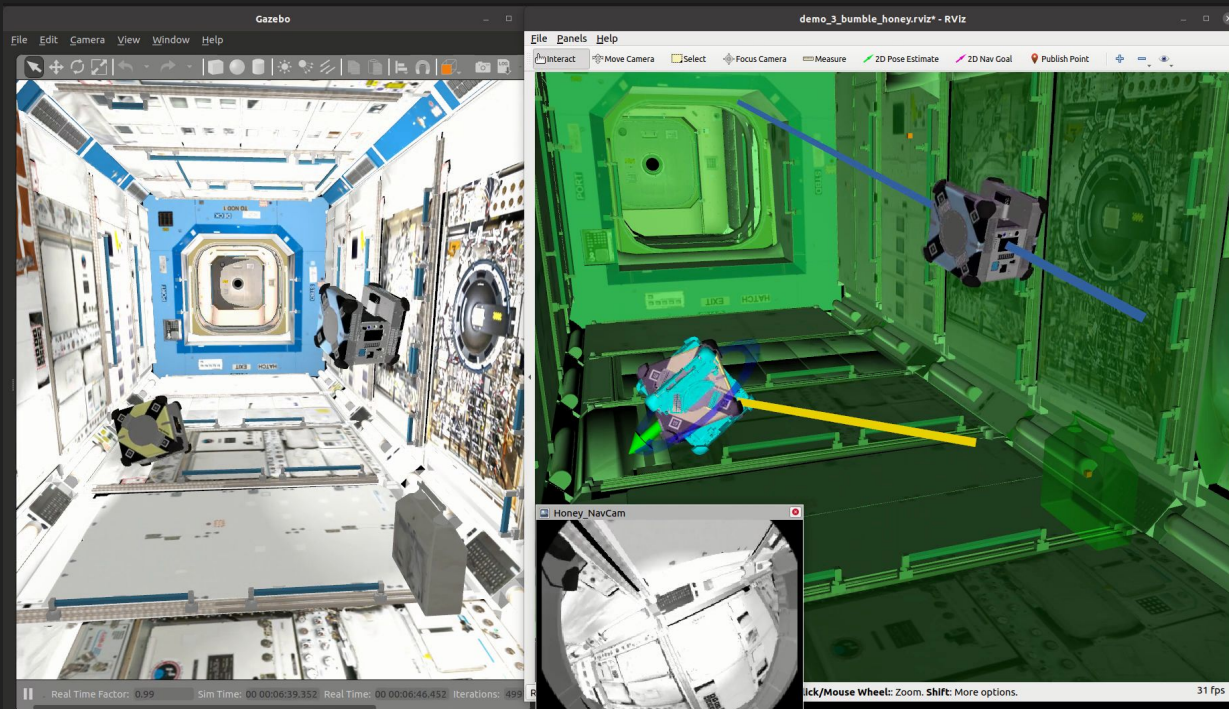


1. A centralized place to share information

- Object geometry.
- Images taken.
- 3D Poses.
- Robot states.

1. Motivation

What do we need to accomplish this?

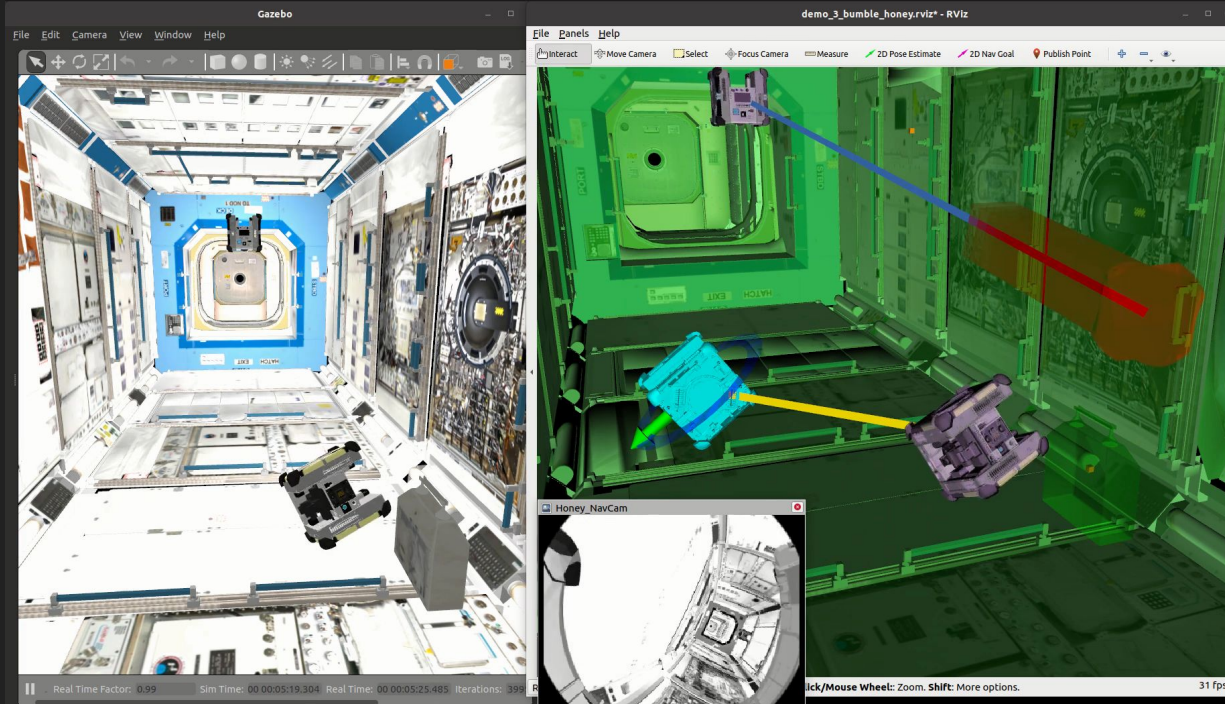


2. A way to register the robot's intentions

- Robot motion plans (navigation trajectories, robot arm joint trajectories)

1. Motivation

What do we need to accomplish this?



3. A way to prevent unsafe situations

- Constant check for collisions between robot's expected motions.

1. Motivation

What do we need to accomplish this?

1. A centralized place to share information

- Object geometry.
- Images taken.
- 3D Poses.
- Robot states.

2. A way to register the robot's intentions

- Robot motion plans (navigation trajectories, robot arm joint trajectories)

3. A way to prevent unsafe situations

- Constant check for collisions between robot's expected motions.

1. Motivation

What do we need to accomplish this?

1. Unified Representation

- Object geometry.
- Images taken.
- 3D Poses.
- Robot states.

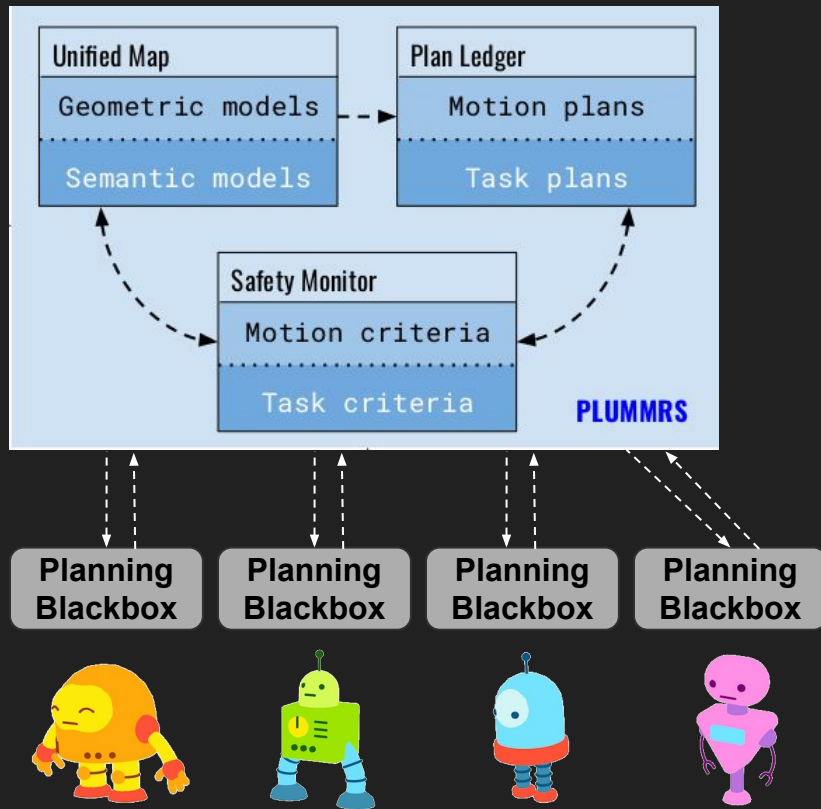
2. Plan Ledger

- Robot motion plans (navigation trajectories, robot arm joint trajectories)

3. Safety Monitor

- Constant check for collisions between robot's expected motions.

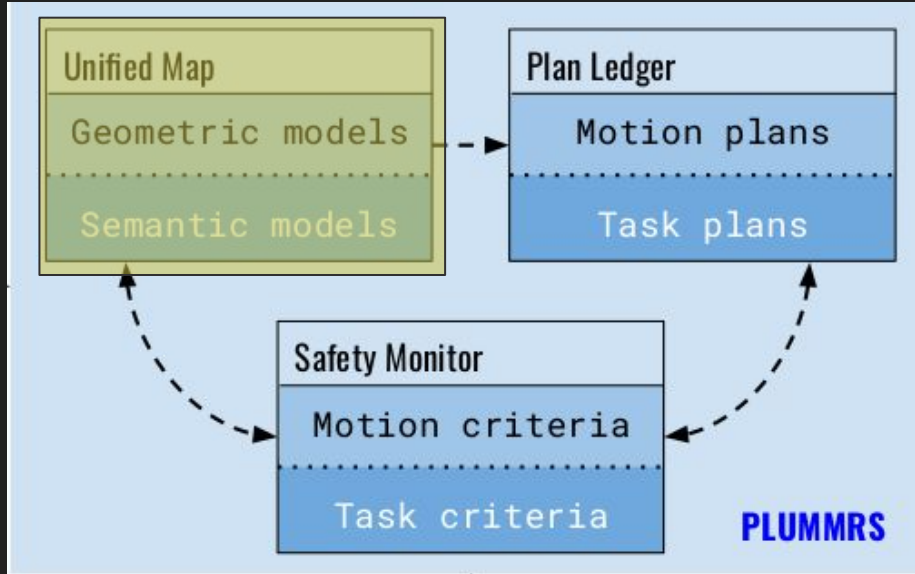
2. PLUMMRS



PLUMMRS consists of 3 elements that will allow it to:

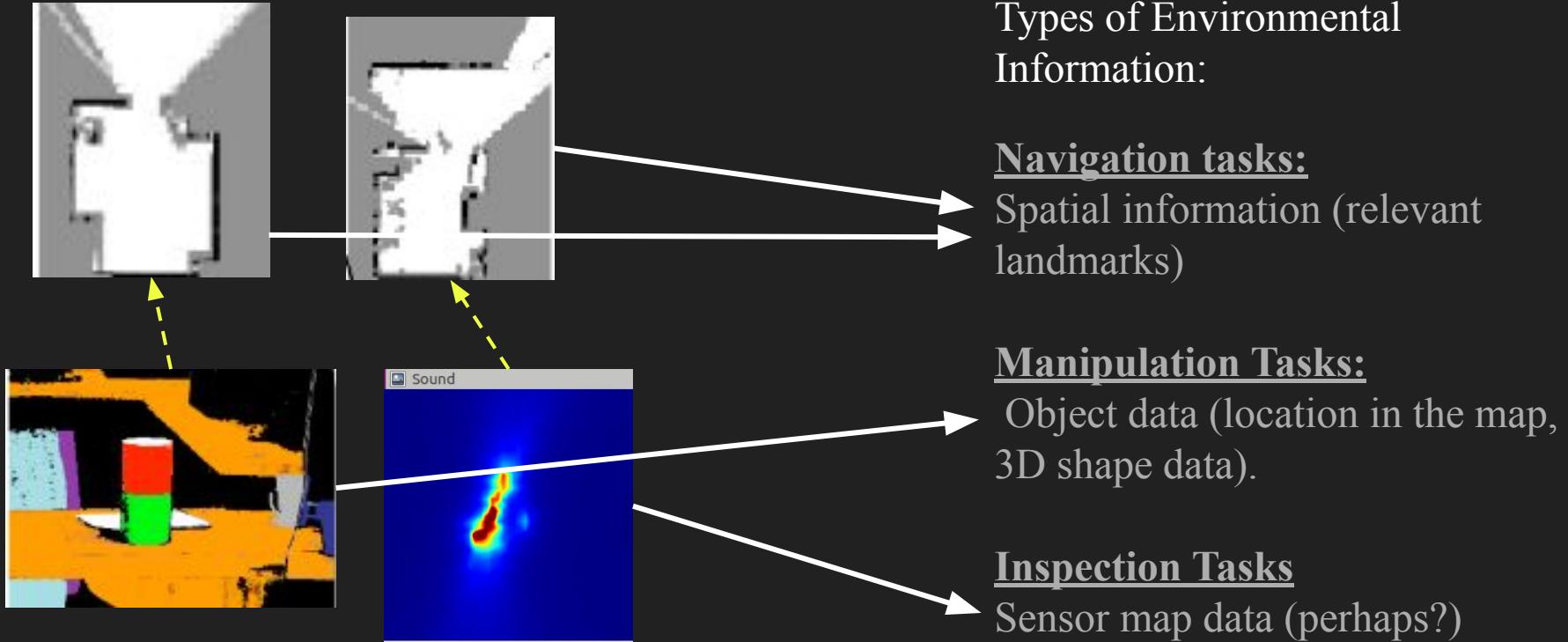
- (1) Unify perceptual information, spatial knowledge and state information.
- (2) Track proposed and currently execution motion trajectories along with task-level symbolic plans that require shared resources.
- (3) Monitor state, validate safety and broadcast concerns at a fast loop rate.

2.1. Unified Representation

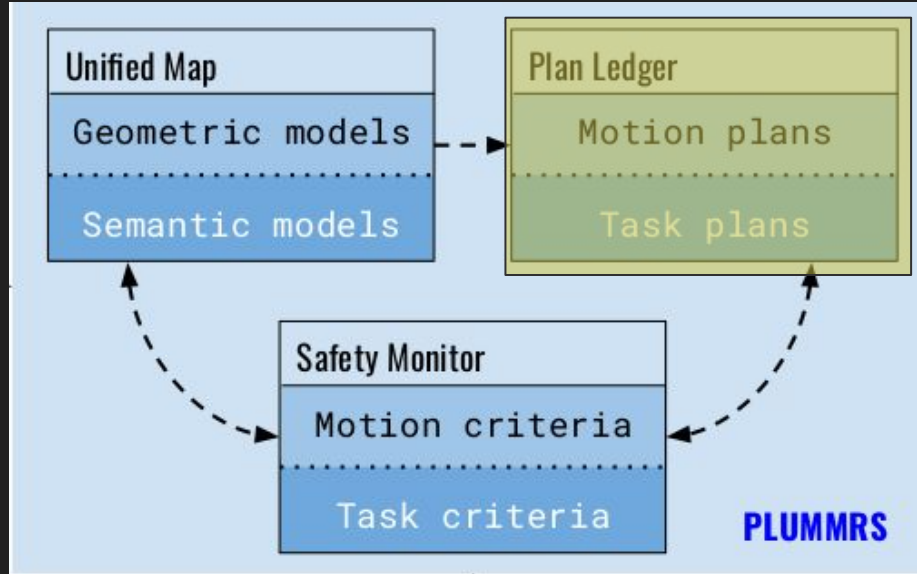


- Single-source database of perception and spatial knowledge.
- Each robot can contribute and query the Unified Map.
- Depending on the application, different types of environmental information will be needed.

2.1. Unified Map

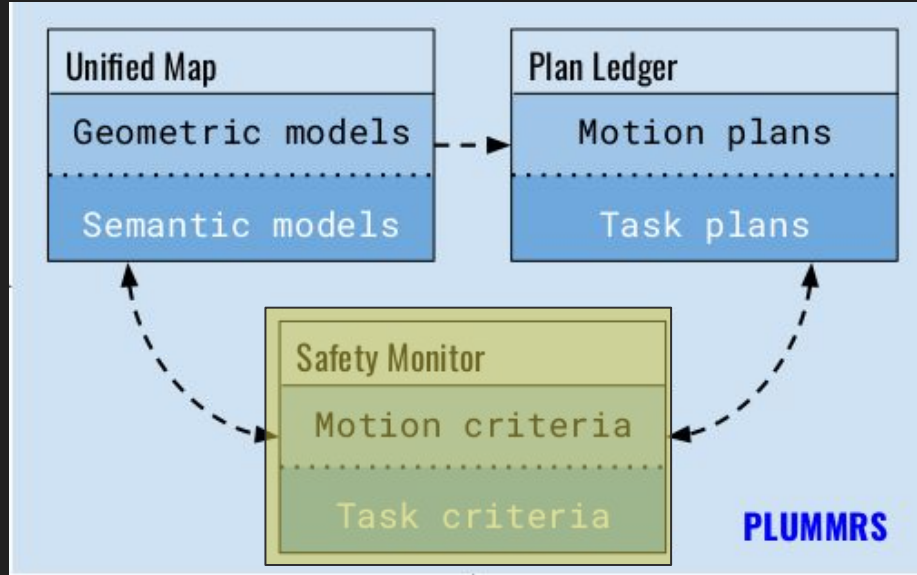


2.2. Plan Ledger



The Plan Ledger exists to ensure that each robot's proposed plans are validated for safety against the rest of the robots. It also stores the proposed plans.

2.3. Safety Monitor

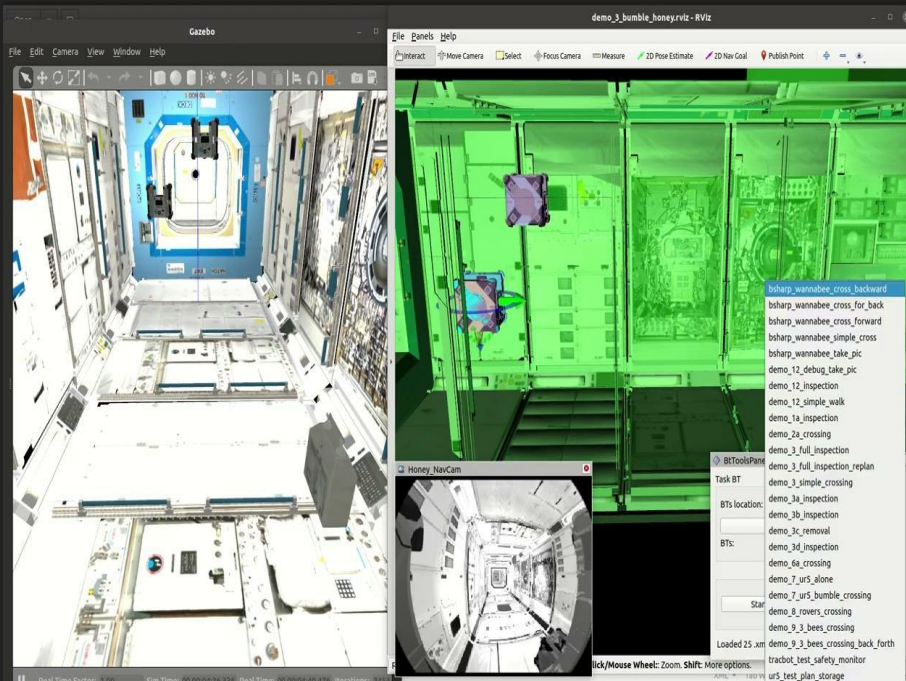


Even the best-laid plans of robots can go awry.

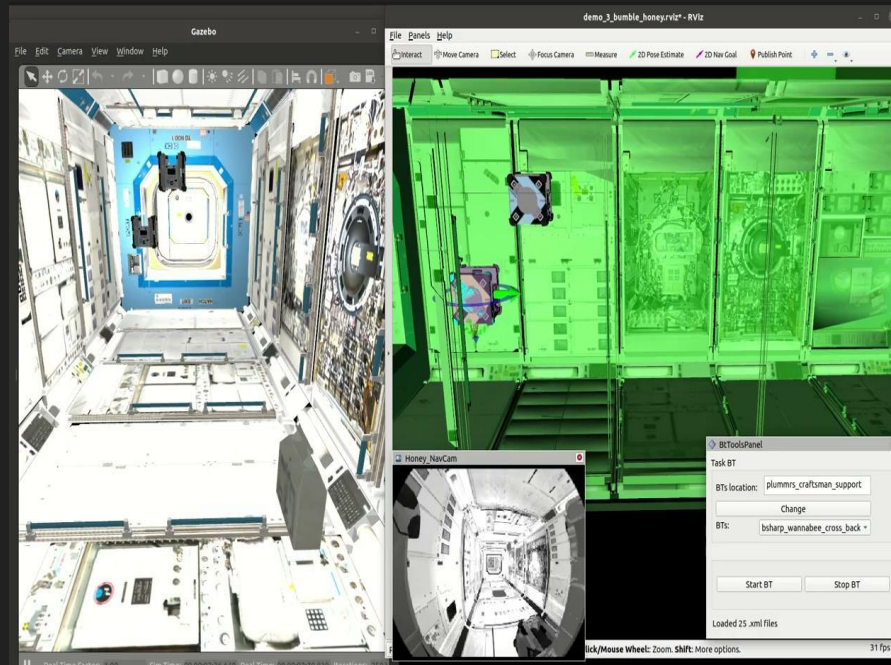
Possible issues:

- Potential collisions on new-sensor data.
- A robot not adequately following its plan, which is an issue for the robot itself and nearby robots.
- A human moving too close to the robot.

2.3. Safety Monitor



With padding = 0.15m



With no padding

3. Hardware Validation

Simple Inspection Task Demo

August 17th, 2023

4. Extra! On Task Description

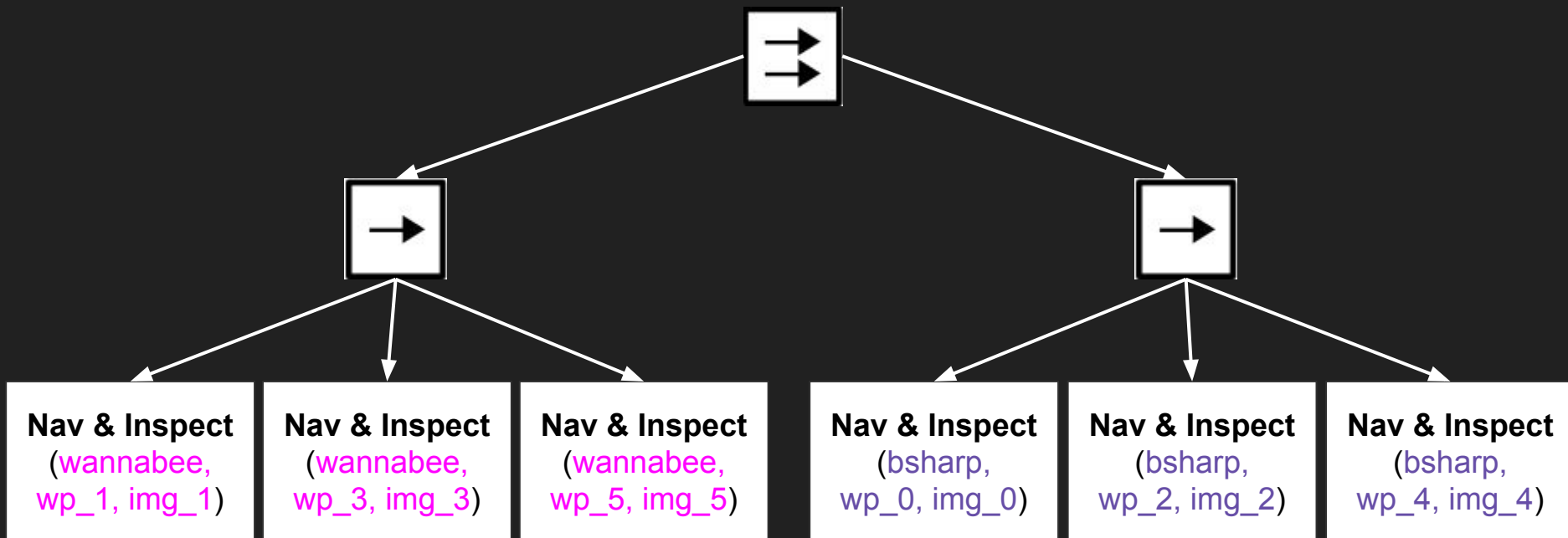
We encode tasks in 2 ways:

1. **Behavior Trees**
2. PDDL tasks, where actions are encoded as Behavior Trees Action node

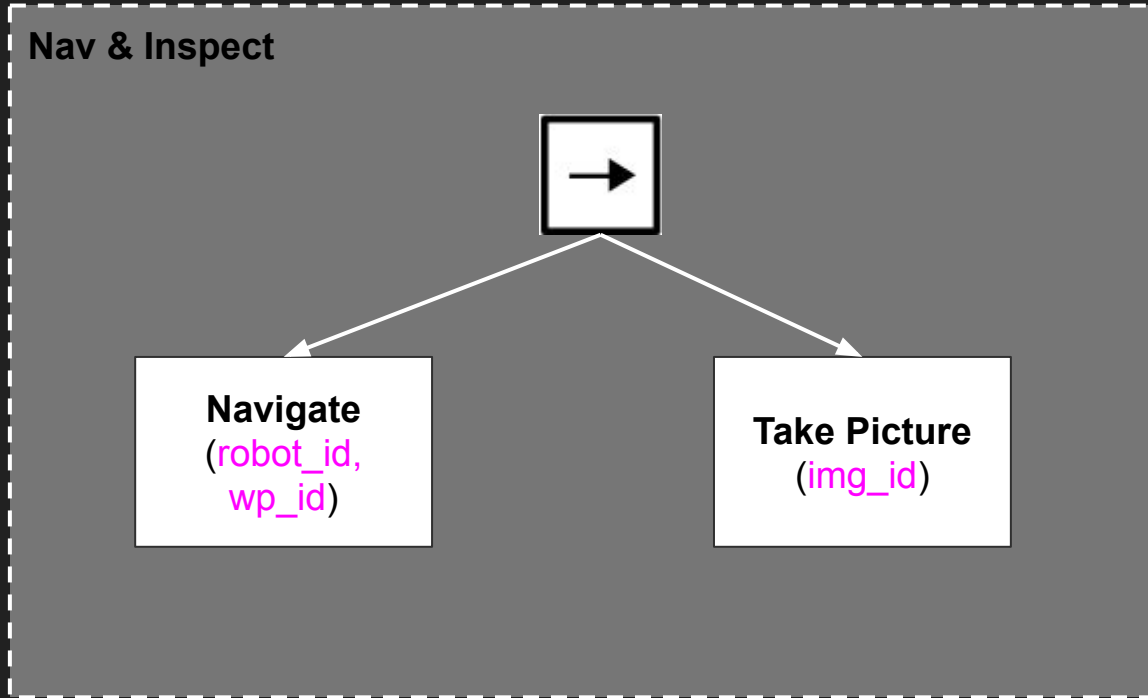


PDDL

4. On Task Description: ARC Demo



4. On Task Description: ARC Demo



4. On Task Description: ARC Demo

```
<root BTCPP_format="4" main_tree_to_execute = "MainTree">

  <!-- ***** MainTree ***** -->
  <BehaviorTree ID="MainTree">

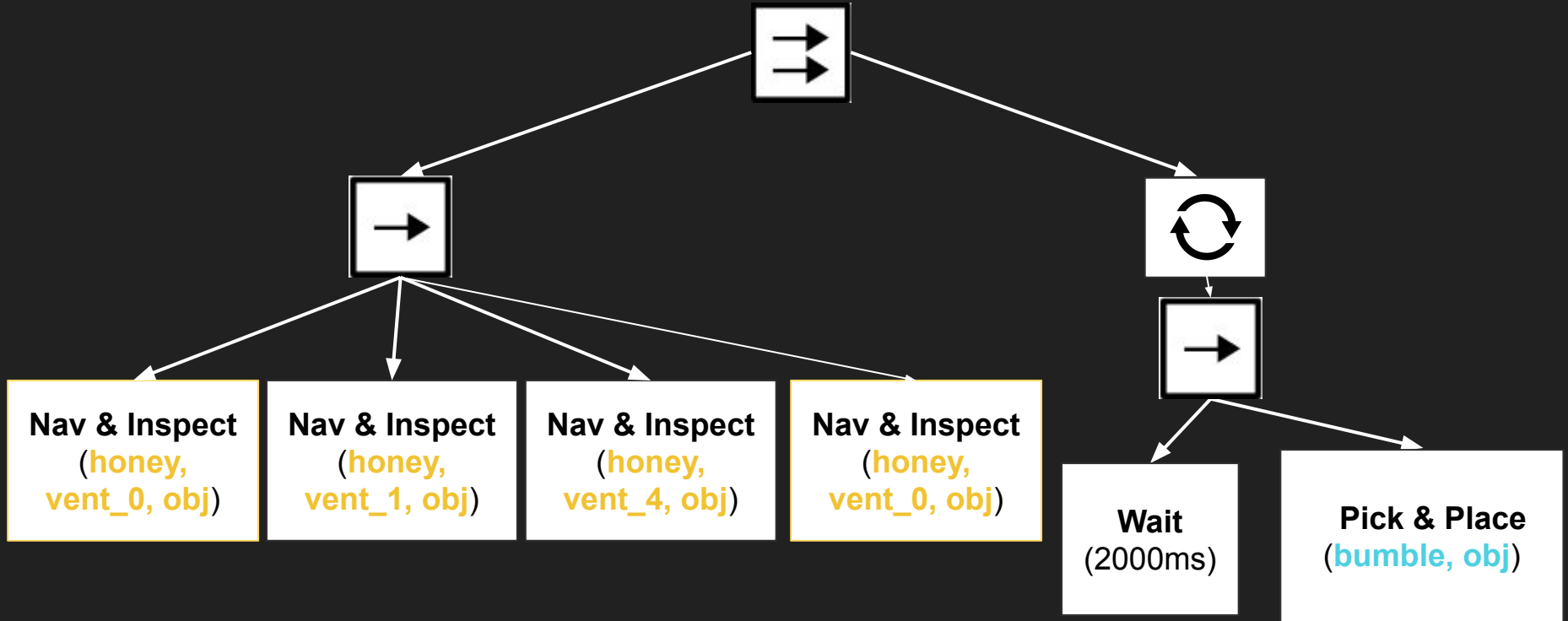
    <Parallel success_count="2">
      <Sequence>
        <SubTree ID="NavAndInspect" robot_key="wannabee" pose_key="wp_1" image_key="img_1"/>
        <SubTree ID="NavAndInspect" robot_key="wannabee" pose_key="wp_3" image_key="img_3"/>
        <SubTree ID="NavAndInspect" robot_key="wannabee" pose_key="wp_5" image_key="img_5"/>
      </Sequence>

      <Sequence>
        <SubTree ID="NavAndInspect" robot_key="bsharp" pose_key="wp_0" image_key="img_0"/>
        <SubTree ID="NavAndInspect" robot_key="bsharp" pose_key="wp_2" image_key="img_2"/>
        <SubTree ID="NavAndInspect" robot_key="bsharp" pose_key="wp_4" image_key="img_4"/>
      </Sequence>
    </Parallel>
  </BehaviorTree>

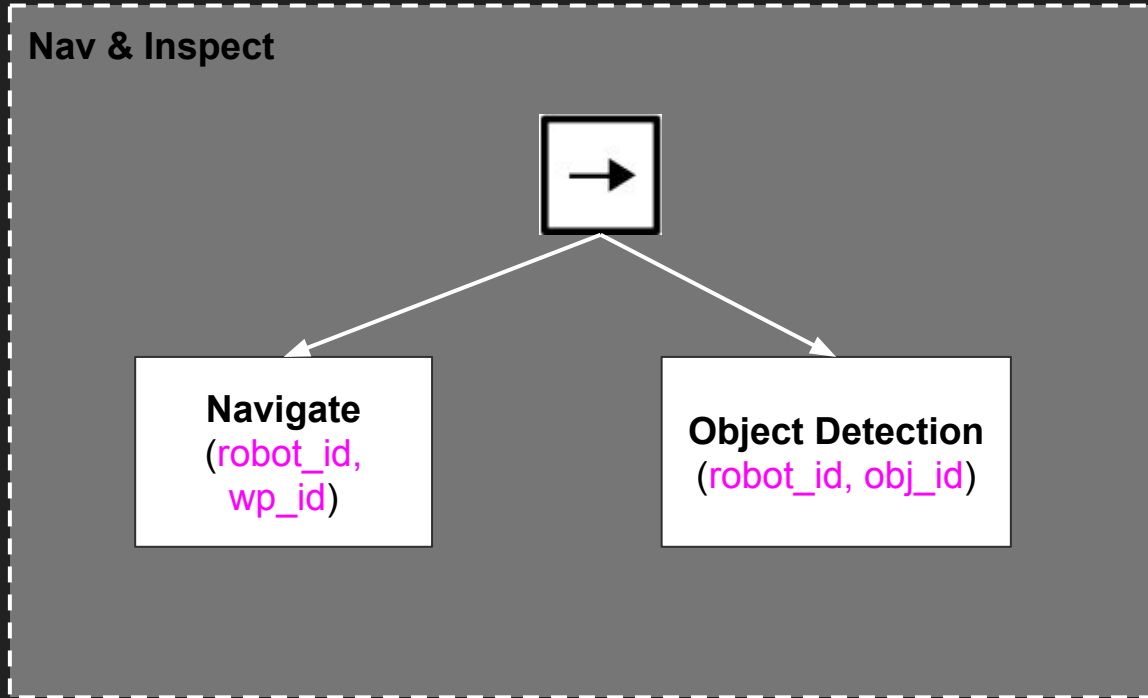
  <!-- ***** NavAndInspect SubTree ***** -->
  <BehaviorTree ID="NavAndInspect">
    <Sequence>
      <AwareNavigatePlum robot="{robot_key}" pose_id="{pose_key}"/>
      <TakePictureNavCam robot="{robot_key}" image_id="{image_key}"/>
      <ShowImage image_id="{image_key}"/>
    </Sequence>
  </BehaviorTree>

</root>
```

4. On Task Description: Inspection and disposal



4. On Task Description: ARC Demo



4. On Task Description: Inspection and disposal

```
<root BTCPP_format="4" main_tree_to_execute = "MainTree">

  <!-- ***** MainTree ***** -->
  <BehaviorTree ID="MainTree">

    <Parallel success_count="2">

      <!-- Honey inspects -->
      <Sequence>
        <SubTree ID="NavAndInspect" robot_key="honey" pose_key="vent_0" object_key="object_detected"/>
        <SubTree ID="NavAndInspect" robot_key="honey" pose_key="vent_1" object_key="object_detected"/>
        <SubTree ID="NavAndInspect" robot_key="honey" pose_key="vent_4" object_key="object_detected"/>
        <SubTree ID="NavAndInspect" robot_key="honey" pose_key="vent_5" object_key="object_detected"/>
      </Sequence>

      <!-- Bumble waits for object and then picks it up. Keeps repeating till succeeds -->
      <RetryUntilSuccessful num_attempts="1">
        <Sequence>
          <Wait msec="2000"/>
          <AwareManipulatorPickPlacePlumNode robot="bumble"
                                             goal="obstacle_disposal"
                                             object_id="object_detected"/>
        </Sequence>
      </RetryUntilSuccessful>
    </Parallel>

  </BehaviorTree>

  <!-- ***** Nav & Inspect ***** -->
  <BehaviorTree ID="NavAndInspect">
    <Sequence>
      <AwareNavigatePlum robot="{robot_key}" pose_id="{pose_key}"/>
      <ObjectDetection robot="{robot_key}" object_id="{object_key}"/>
    </Sequence>
  </BehaviorTree>

</root>
```

Thanks!

