

PHARAOH: An English-language procedure-based application framework for full or semi-autonomous robot operations

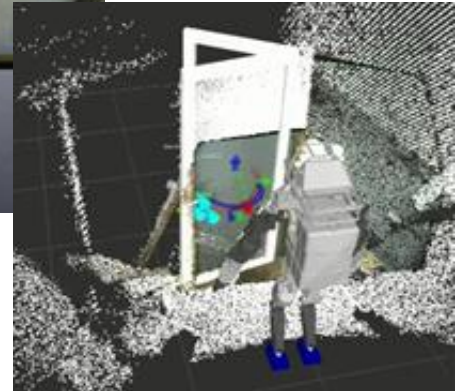
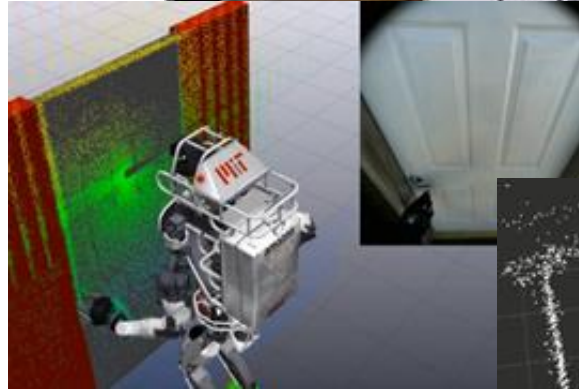
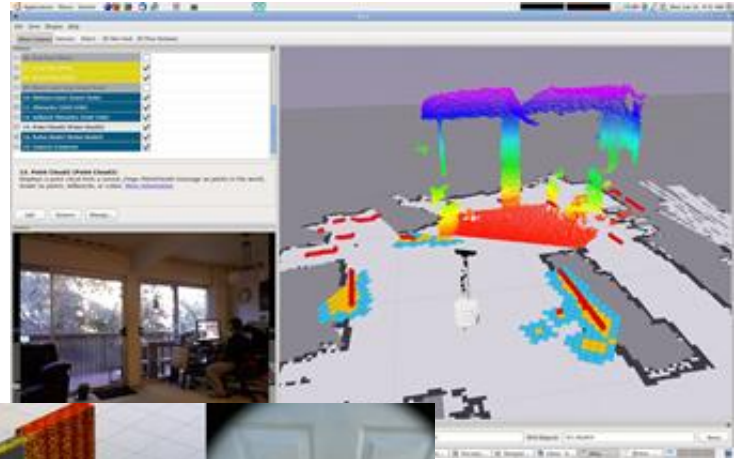
ROS-I Americas 2021

Stephen Hart (swhart@traclabs.com)

Rob Burrige presenting
TRAC Labs, Inc.

Motivation

- Recent advances in interactive 3D user interfaces have revolutionized how humans program and work with robots to perform tasks.



Motivation

- Recent advances in interactive 3D user interfaces have revolutionized how humans program and work with robots to perform tasks.
- Unfortunately, these “state-of-the-art” systems typically require a level of specialization that has kept these tools in the laboratory and research community.



vs.



Motivation

- Recent advances in interactive 3D user interfaces have revolutionized how humans program and work with robots to perform tasks.
- Unfortunately, these “state-of-the-art” systems typically require a level of specialization that has kept these tools in the laboratory and research community.
- Further, there is a vast mismatch between these tools and the traditional tools that direct humans to perform tasks or operate equipment during field operations.

GROUND COMMUNICATIONS AVAILABLE

NOTE

- The TEPC alarm is locally silenced using a switch on the Spectrometer. TEPC dose rate information may be read from the TEPC display.
- The alarm LED will flash as long as the dose rate exceeds the setpoint. When the dose rate goes below the setpoint, the LED will go dark. There is no audible indication that the alarm condition has cleared.

- Acknowledge Alarm**
Verify alarm LED – Flashing Red
‘Audio Alarm sw – ‘OFF’
Verify dose rate reading > 5 mrad/min.
- Notify MCC-H**
Time of alarm.
Initial alarm reading.
LED status during alarm.

NOTE

- The radiation ground support team is the primary means of assessing the radiation environment. A need to take special actions will be confirmed.
- Note the time when the alarm clears. Not condition will clear in less than 15 minutes.

- Follow ground instructions.
- When alarm clears**
Note the time when the alarm clears.
Reset alarm.
Verify alarm LED – Dark
‘Audio alarm sw – ‘On’

NOTE

Transit through high radiation zones resulting solar proton events may occur at ~45-minute intervals. Alarm notification may result from any subject until the event subsides.

- Acknowledge and clear alarms as they occur. Note timing and trends as directed in steps 1 a

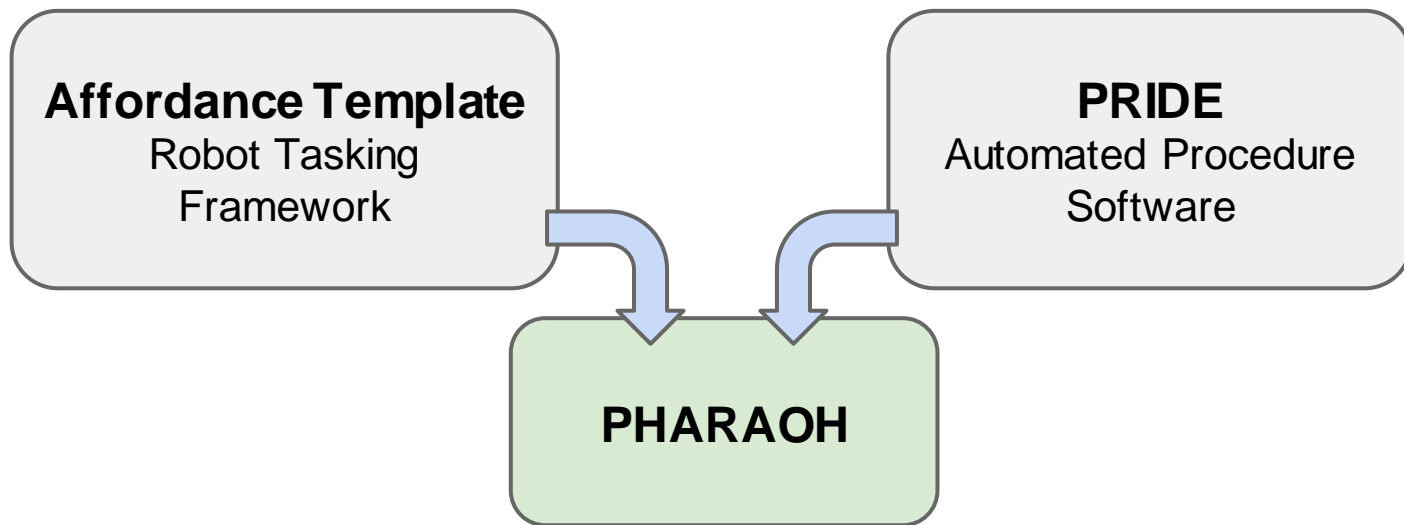
23 AUG 00

175

RIGGING EQUIPMENT		SECTION 1B
SAFETY		SHACKLES
<input type="checkbox"/> Slings & Rigging inspected PRIOR TO USE	<input type="checkbox"/> Shackles are at 100% of capacity when slings are from 0° to 90° of vertical	<input type="checkbox"/> Shackles are at 75% of capacity when the included sling angles are from 12° to 90° of vertical
<input type="checkbox"/> Objects in systems of synthetic slings DO NOT exceed 1/3 the length of the system.	<input type="checkbox"/> Shackles are at 50% of capacity when the included sling angles are from 91° to 180°	Number of Shackles: _____
<input type="checkbox"/> Objects in systems of wire rope slings DO NOT exceed 1/2 the length of the system.	Shackle Capacity: _____	
<input type="checkbox"/> The End Rate of any attached rigging equipment is in the Work Place as 2011		
<input type="checkbox"/> The included angle of the slings attaching to the hook DOES NOT exceed 90°		EYEBOLTS
<input type="checkbox"/> Sharp Edge Protection	<input type="checkbox"/> Non-shrouded eyebolts are at 100% of capacity when the slings are from 0° to 90° of vertical	<input type="checkbox"/> Non-shrouded eyebolts are NOT recommended with sling angles more than 90° from vertical
<input type="checkbox"/> Shield and protect slings and rigging against chemical exposure and extreme heat	<input type="checkbox"/> Shrouded eyebolts are at 100% of capacity when the slings are from 0° to 90° of vertical	<input type="checkbox"/> Shrouded eyebolts are at 75% of capacity when the sling angles are from 91° to 180° of vertical
<input type="checkbox"/> Safety notified 5 days prior to lift	<input type="checkbox"/> Shrouded eyebolts are at 50% of capacity when the sling angles are from 90° to 180° of vertical	
SLINGS		
<input type="checkbox"/> SYNTHETIC	<input type="checkbox"/> WIRE ROPE	<input type="checkbox"/> Non-shrouded
<input type="checkbox"/> CHAIN	<input type="checkbox"/> WIRE MESH	Eyebolt Capacity: _____ (lbs.)
SLING CONFIGURATION & CAPACITY		HOOK
<input type="checkbox"/> Vertical	<input type="checkbox"/> Vertical is other 0° to 9° to be 100%	Included angle to the hook DOES NOT exceed 90°
<input type="checkbox"/> Basket	(Wire Rope Slings require End of 25:1)	<input type="checkbox"/> Latch required
<input type="checkbox"/> Choker	(Shackles @ approximately 75% of Vertical)	<input type="checkbox"/> Latch required in the open position
Number of Slings: _____		<input type="checkbox"/> Latch required in the closed position
<input type="checkbox"/> 2 Leg Bride		MASTERLINK
<input type="checkbox"/> 3 Leg Bride		Masterlink Capacity: _____
<input type="checkbox"/> 4 Leg Bride		SWIVEL HOIST RING
<input type="checkbox"/> More than 4 Leg Bride		Torque value: _____
<input type="checkbox"/> Tag Line		Rated load ring capacity: _____

Purpose of this work

We aim to bridge the gap between state-of-the-art **robot application tools** and mission-appropriate **procedure execution software** to enable the use of complex robot systems in real-world field operations.



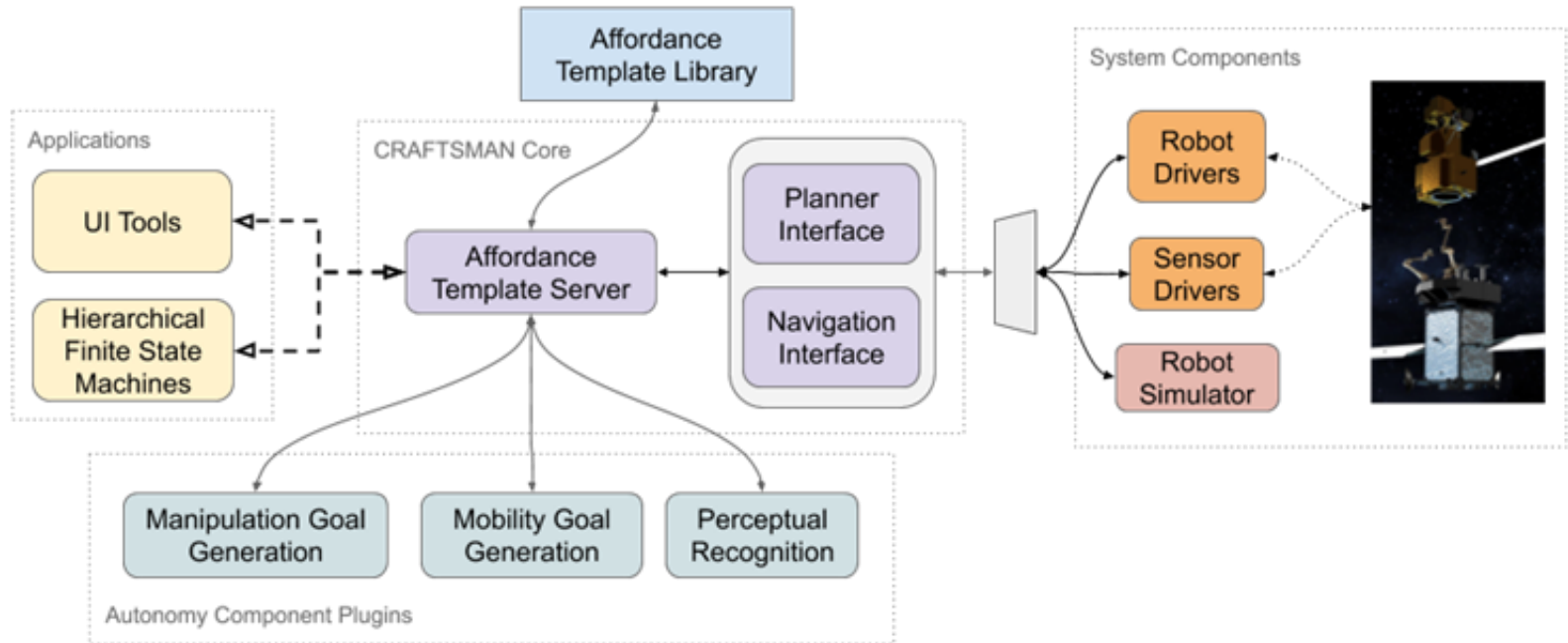
Purpose of this work

We aim to bridge the gap between state-of-the-art **robot application tools** and mission-appropriate **procedure execution software** to enable the use of complex robot systems in real-world field operations.



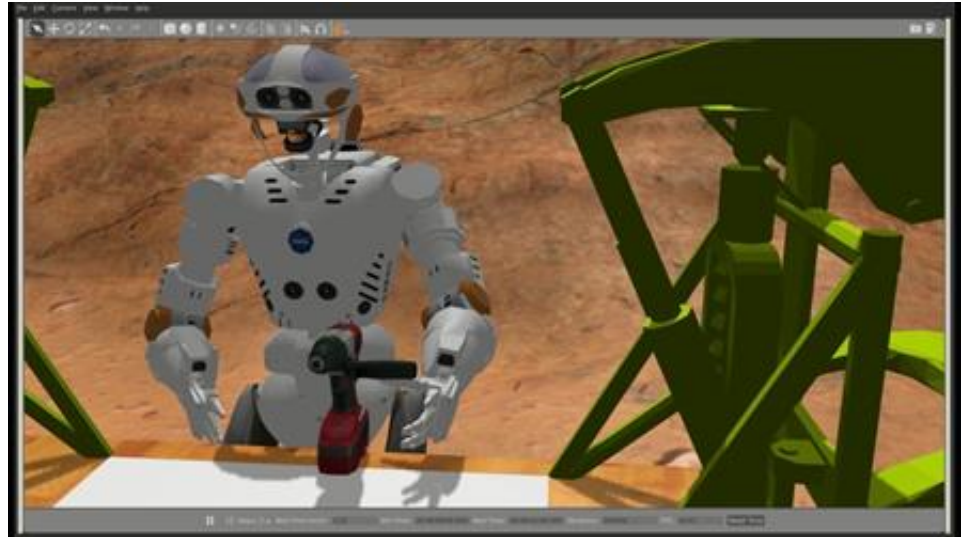
The CRAFTSMAN Software Tool-Suite

CRAFTSMAN is a **ROS-based** framework for developing advanced, sensor-driven, robotic applications for use with a wide variety of robots.



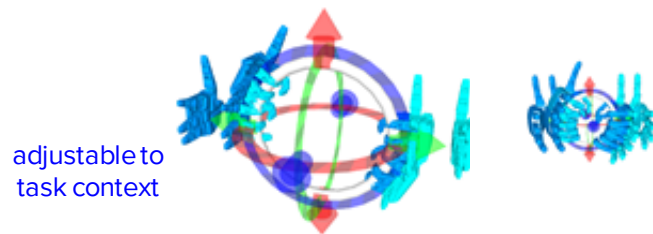
CRAFTSMAN Capabilities

- Suitable for teleoperation, shared autonomy, and full automation
- Matured through multiple deployments in industrial and NASA contexts
- Supports a wide variety of advanced motion planning techniques and features
 - Cartesian and Joint motion
 - collision free paths (OMPL)
 - tolerance/constraint based parameterization
 - plan optimization
 - multiple levels of safety checking
 - coordinated motion planning among multiple manipulators and robots
 - integrated mobile manipulation
 - grasp planning

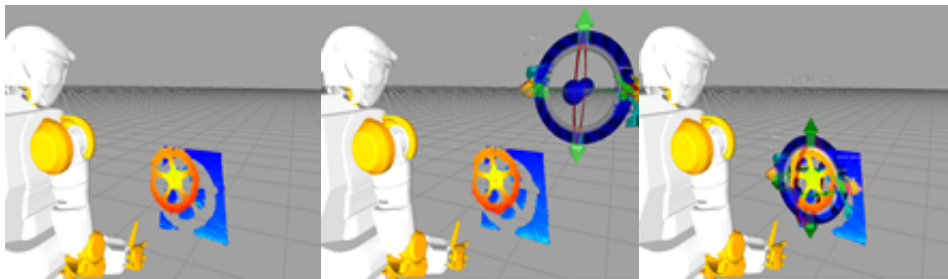


Affordance Templates

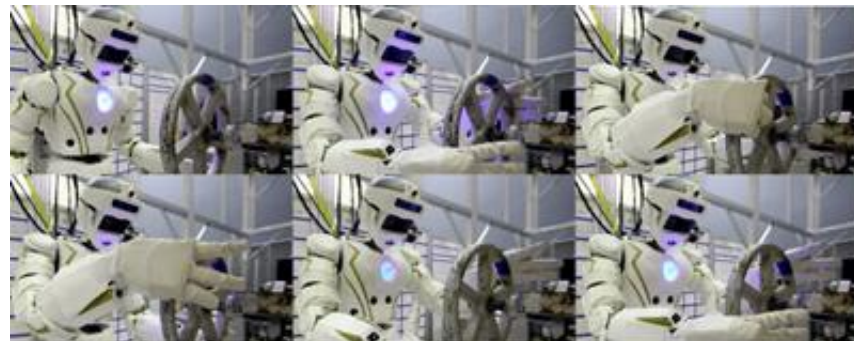
- Interactive 3D task programming and execution framework
 - “Object”-oriented task encoding
 - Inherent generalization across *robots*, *tasks*, and *environments*
 - Used on Robonaut 2, Valkyrie, Atlas, Industrial Robots, etc.
- Supports different levels of *shared* and *full* autonomy
 - Authoring, monitoring, and interacting/supervising
- Intuitive graphical tools to set or adjust task parameters and get feedback from the robot



ROS

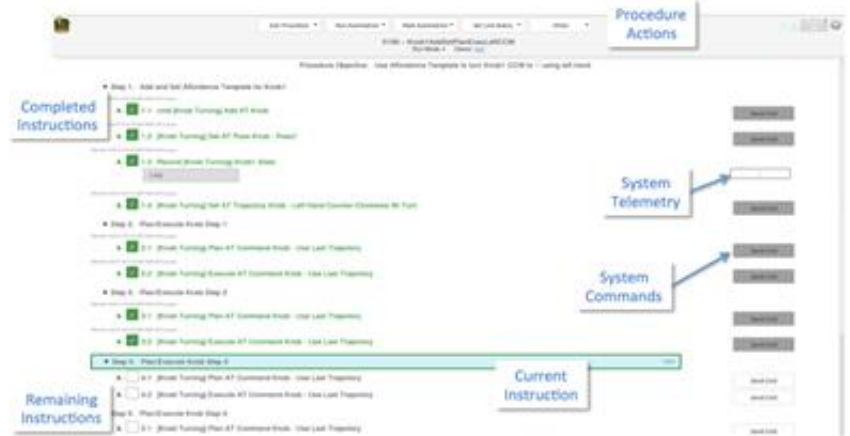
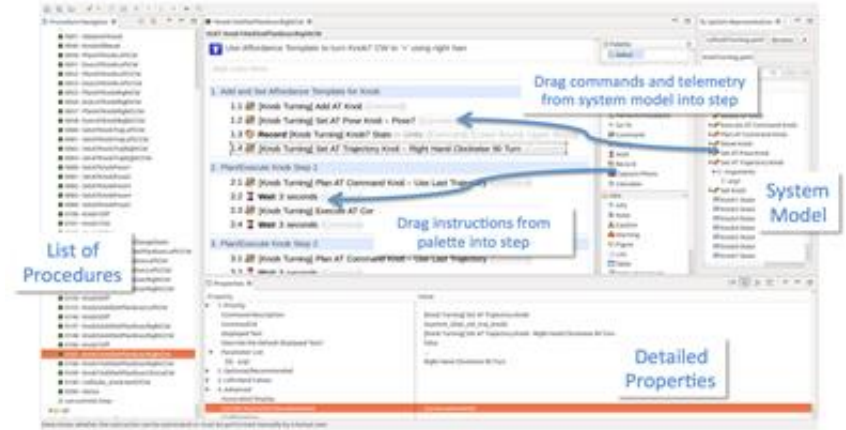


operator registers templates to run-time
sensor data



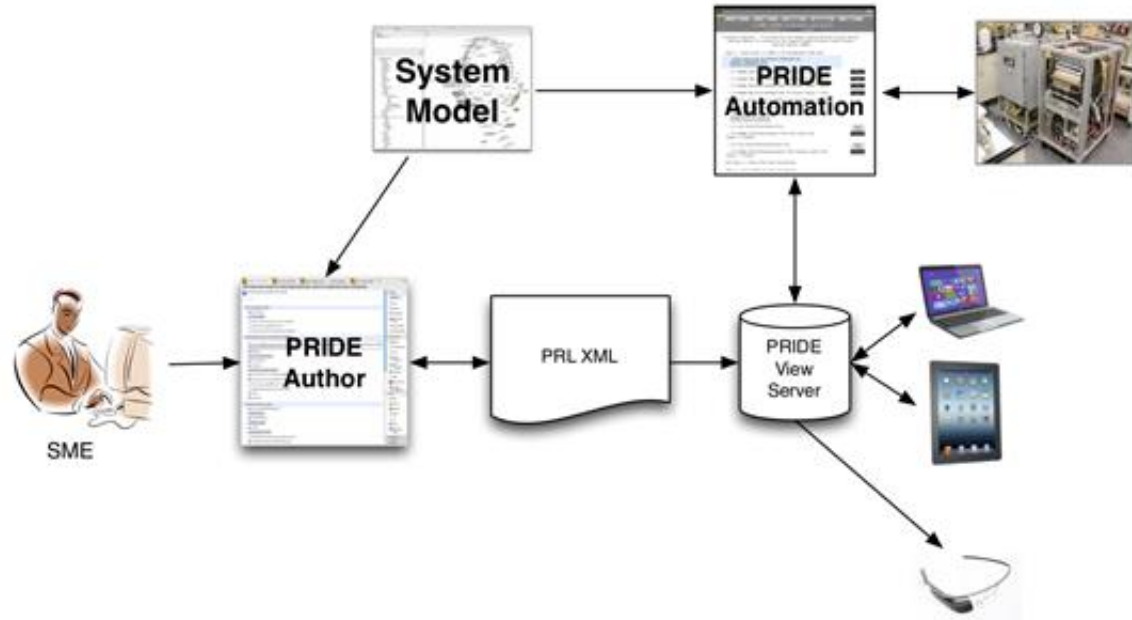
PRIDE - Electronic Procedure Automation Software

- Procedure **authoring** and **execution** IDE designed for NASA and service-sector *human* field operations
 - Sub-procedures can be composed hierarchically
 - Text-based, cross-platform, web front-end supports intuitive usage by trained operators



PRIDE - Electronic Procedure Automation Software

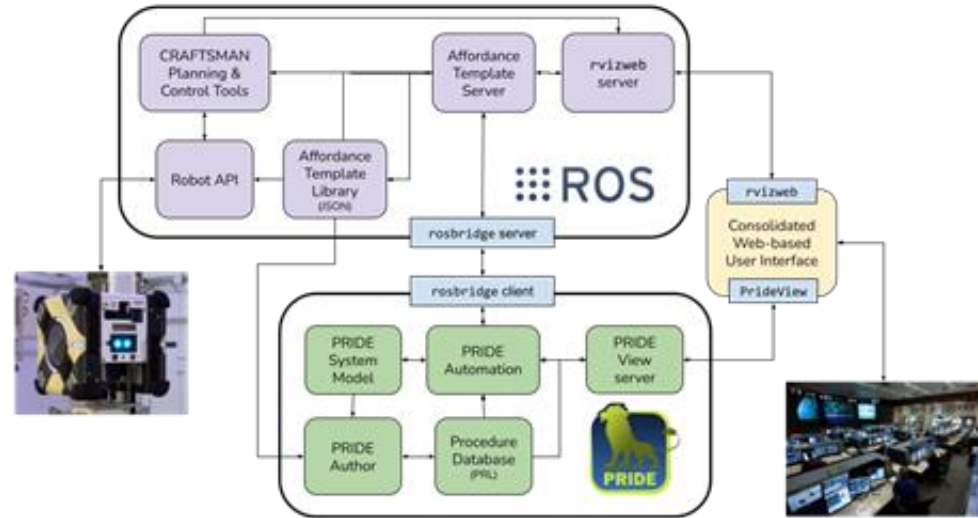
- Procedure **authoring** and **execution** IDE designed for NASA and service-sector *human* field operations
 - Sub-procedures can be composed hierarchically
 - Text-based, cross-platform, web front-end supports intuitive usage by trained operators
- **Telemetry** and **control** can be integrated into procedures via custom System Models
 - Increased automation and gathering of performance metrics

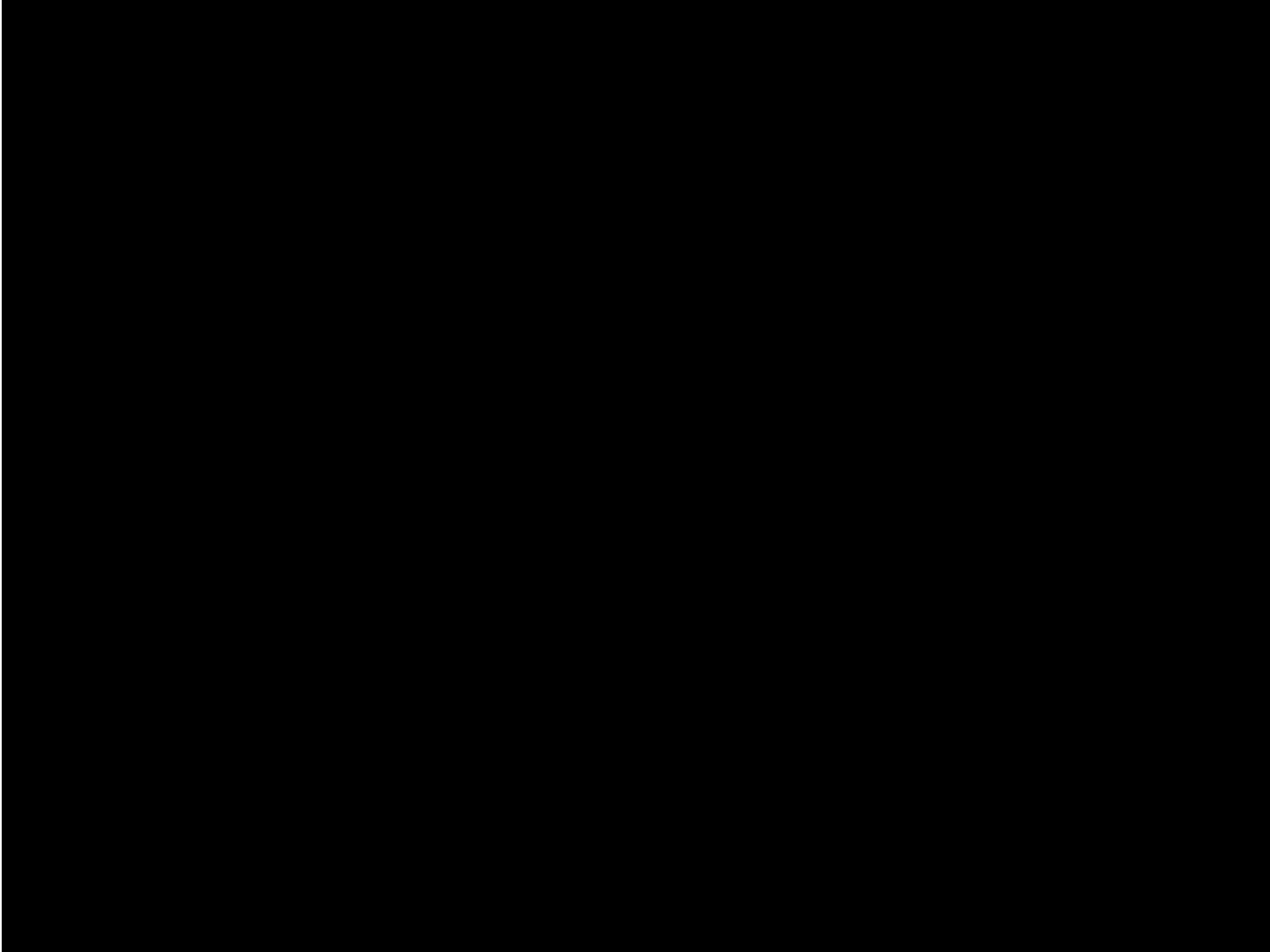


PHARAOH: Using PRIDE to Control Robots

- Procedure-Handling Architecture for Robots And/Or Humans

- Create a PRIDE **System Model** to interface with the Affordance Template ROS API
- Automate AT program flow:
 - Adding/Deleting
 - Planning/Executing
 - Setting strategy
- Walk an operator through adjusting AT task goals in 3D interface
 - Example: “Use 6-DOF arrows to register the virtual knob model to the 3D sensor data”
- Gather performance statistics during operation
 - System data
 - Timing
 - Success/failure rates
 - Manual vs. automated steps





PRIDE Precision

1.1.2.2 - Work/Play Schedule - View job

Step 1: Prepare to run Knobs

1.1.1 Record knob setting

Step 2: Run knob

2.1.1 (optional) Record Knob Data

2.1.2 Record knob settings

2.1.3 Confirm knob setting

Step 3: Complete Knob setting

3.1 Record knob setting

1.1.1 - Work/Play Schedule - View job

Step 3: Part Execution, Step 2

3.1.1 Plan to grab work into new delivery

3.1.2 Wait for part used

3.1.3 Wait for part complete

3.1.4 Grab work with Peeper

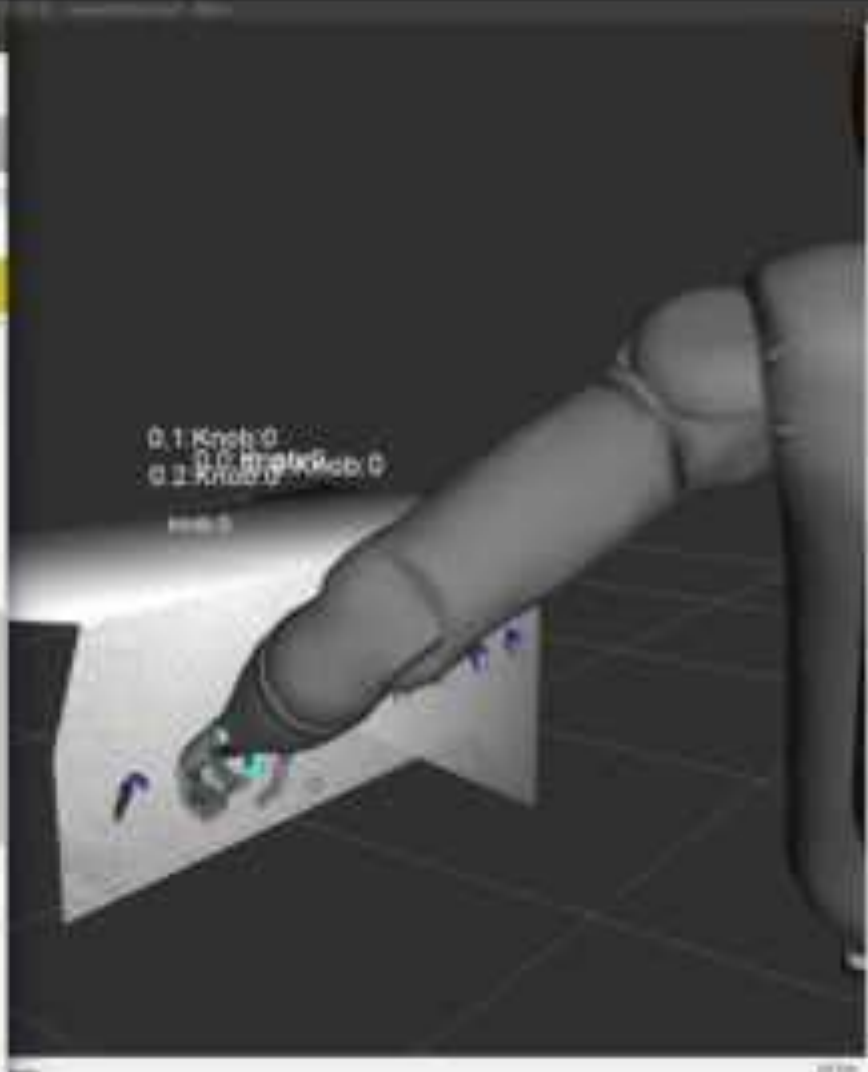
3.1.5 Wait for more work

3.1.6 Wait for work complete

Step 4: Part Execution, Step 2

4.1 Plan to put work into new delivery

4.2 Wait for part used



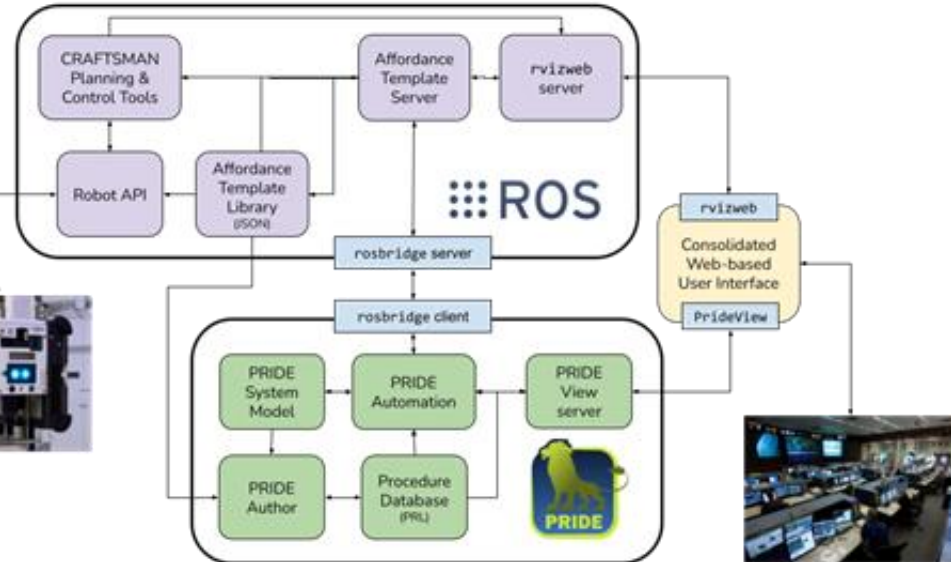
Affordance Template PRIDE Integration

- To accommodate Affordance Templates in PHARAOH, it is now possible to load AT JSON files **directly** as PRIDE System Representations.



- CRAFTSMAN capabilities needed in PRIDE:

1. Locate an item of interest (a display object).
2. Choose a trajectory that specifies the desired display objects, navigation, and manipulation task.
3. Navigate to a position that facilitates the chosen trajectory's manipulation task.
4. Do the chosen manipulation task.
5. Desiderata that are required by the software for AT usage but can possibly be hidden from the user.



Affordance Template PRIDE Integration

- A simplified ROS Service API was created for the AT server code that allows a **single** point-of-entry between PAX and CRAFTSMAN.

```
uint8 LOAD=0
uint8 UNLOAD=1
uint8 PLAN=2
uint8 EXECUTE=3
uint8 CLEAR=4
uint8 LOCALIZE=5

string template_name
string trajectory_name
uint8 command_type # One of the constants above
bool navigation
string identifier # Used for planning, executing, and clearing plans
string precondition # Only needed for planning
string display_object
---
bool success
string result_message
string identifier
```

Affordance Template PRIDE Integration

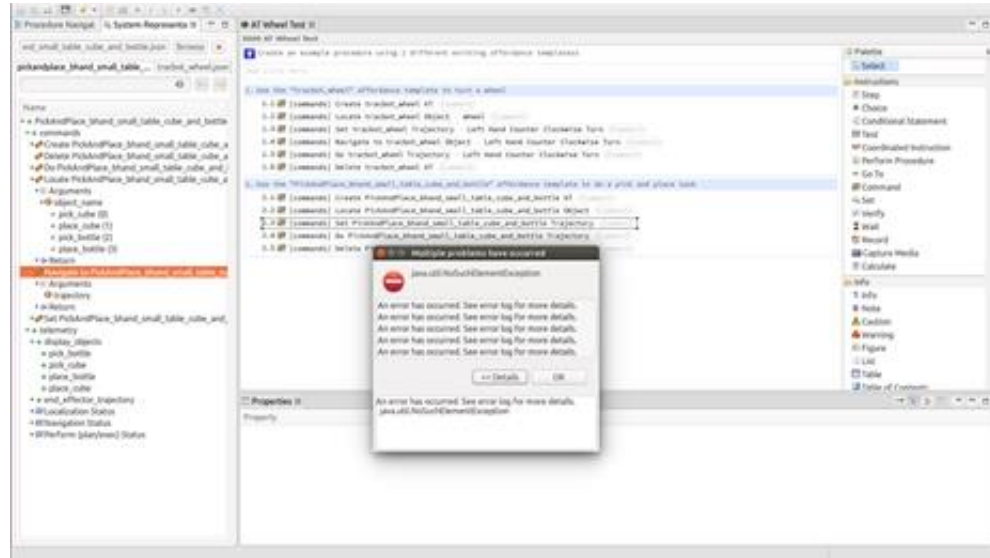
- A simplified ROS Service API was created for the AT server code that allows a **single** point-of-entry between PAX and CRAFTSMAN.
- New PAX Telemetry and Control Java components were added to communicate with this API.

```
public CommandItem sendCommandItem(String commandID, List<CommandArgument> arguments,
                                  CommandItem result, ATStatus status)
    throws IllegalStateException, Exception {
    CommandItem retval = null;
    String cmdid = commandID;
    if (poolCU == null) {
        logger.warn("sendCommandItem: '"+ commandID + "', refused ...missing connection ...utility object");
        return result;
    } else if (!poolCU.isCommandItem(cmdid)) {
        logger.error("sendCommandItem: unknown command: '"+ cmdid + "'");
        return result;
    } else if (!poolCU.isWell() {
        logger.error("sendCommandItem: '"+ commandID + "', refused ...no Service ...attempt ...creation");
        status.set(InstructionConnectionUtil.CALLCONNECT_TIMEOUT);
        return result;
    }
}

try {
    String s1Try = null;
    switch (cmdid) {
        case "at locate":
            if (arguments.size() > 0) {
                s1Try = SpringUtilUtil.getProperty(arguments.get(0).getValueDefinition());
            }
            try {
                retval = locateAT(status, cmdid, s1Try, result);
            } catch (ConnectionPendingException spe) {
                logger.warn("sendCommandItem: '"+ cmdid + "', connection ...pending");
                retval = locateAT(status, cmdid, s1Try, result);
            }
            break;
        case "at add":
            case "at delete":
            try {
                retval = sendAT(status, cmdid, arguments);
            } catch (ConnectionPendingException spe) {
                logger.warn("sendCommandItem: '"+ cmdid + "', connection ...pending");
                retval = sendAT(status, cmdid, arguments);
            }
            break;
        case "at position":
            if (arguments.size() > 0) {
                s1Try = SpringUtilUtil.getProperty(arguments.get(0).getValueDefinition());
            }
            try {
                retval = doAT(status, cmdid, s1Try, (cmdid.equals("at.navigate"));
            } catch (ConnectionPendingException spe) {
                logger.warn("sendCommandItem: '"+ cmdid + "', connection ...pending");
                retval = doAT(status, cmdid, s1Try, (cmdid.equals("at.navigate"));
            }
            break;
        case "at.navigate":
            if (arguments.size() > 0) {
                s1Try = SpringUtilUtil.getProperty(arguments.get(0).getValueDefinition());
            }
            try {
                retval = doAT(status, cmdid, s1Try, (cmdid.equals("at.navigate"));
            } catch (ConnectionPendingException spe) {
                logger.warn("sendCommandItem: '"+ cmdid + "', connection ...pending");
                retval = doAT(status, cmdid, s1Try, (cmdid.equals("at.navigate"));
            }
            break;
        default:
            logger.info("sendCommandItem - generic ...handling ...for '"+ cmdid + "'");
            break;
    }
} catch (ConnectionPendingException spe) {
    String msg = "sendCommandItem: Connection ...failed ...possibly ...no-try";
    logger.warn(msg + "...s spe.getProperty(Message()");
} catch (NullPointerException spe) {
    String msg = "sendCommandItem: ...for ...object ...null ...no-try ...PAX";
    logger.warn(msg + "...s spe.getProperty(Message()");
    throw new IllegalStateException(msg, spe);
} catch (NullPointerException spe) {
    String msg = "sendCommandItem: Connection ...failed ...no-try ...only";
    logger.warn(msg + "...s spe.getProperty(Message()");
    throw new IllegalStateException(msg, spe);
}
return result;
}
```

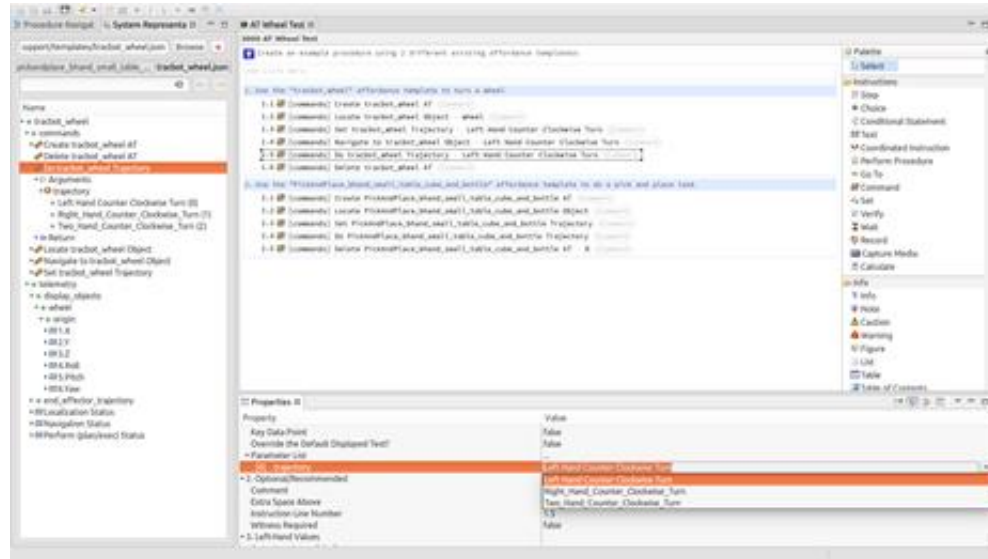
Affordance Template PRIDE Integration

- A simplified ROS Service API was created for the AT server code that allows a **single** point-of-entry between PAX and CRAFTSMAN.
- New PAX Telemetry and Control Java components were added to communicate with this API.
- Safety and helper guides are provided in PRIDE Author to ensure syntactically correct usage in procedures.



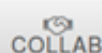
Affordance Template PRIDE Integration

- A simplified ROS Service API was created for the AT server code that allows a **single** point-of-entry between PAX and CRAFTSMAN.
- New PAX Telemetry and Control Java components were added to communicate with this API.
- Safety and helper guides are provided in PRIDE Author to ensure syntactically correct usage in procedures.





PRIDE



TEST USER

3000 - AT Wheel Test Rev:1.0

Static View

Procedure Objective: Create an example procedure using 2 different existing affordance templates

Step 1. Use the "tracbot_wheel" affordance template to turn a wheel

- 1.1 [commands] Create tracbot_wheel AT
- 1.2 [commands] Locate tracbot_wheel Object
- 1.3 [commands] Set tracbot_wheel Trajectory
- 1.4 [commands] Navigate to tracbot_wheel Object
- 1.5 [commands] Do tracbot_wheel Trajectory
- 1.6 [commands] Delete tracbot_wheel AT

Send

Send

Send

Send

Send

Send

Step 2. Use the "PickAndPlace_bhand_small_table_cube_and_bottle" affordance template to do a pick and place task

- 2.1 [commands] Create PickAndPlace_bhand_small_table_cube_and_bottle AT
- 2.2 [commands] Locate PickAndPlace_bhand_small_table_cube_and_bottle Object
- 2.3 [commands] Set PickAndPlace_bhand_small_table_cube_and_bottle Trajectory
- 2.4 [commands] Do PickAndPlace_bhand_small_table_cube_and_bottle Trajectory
- 2.5 [commands] Delete PickAndPlace_bhand_small_table_cube_and_bottle AT

Send

Send

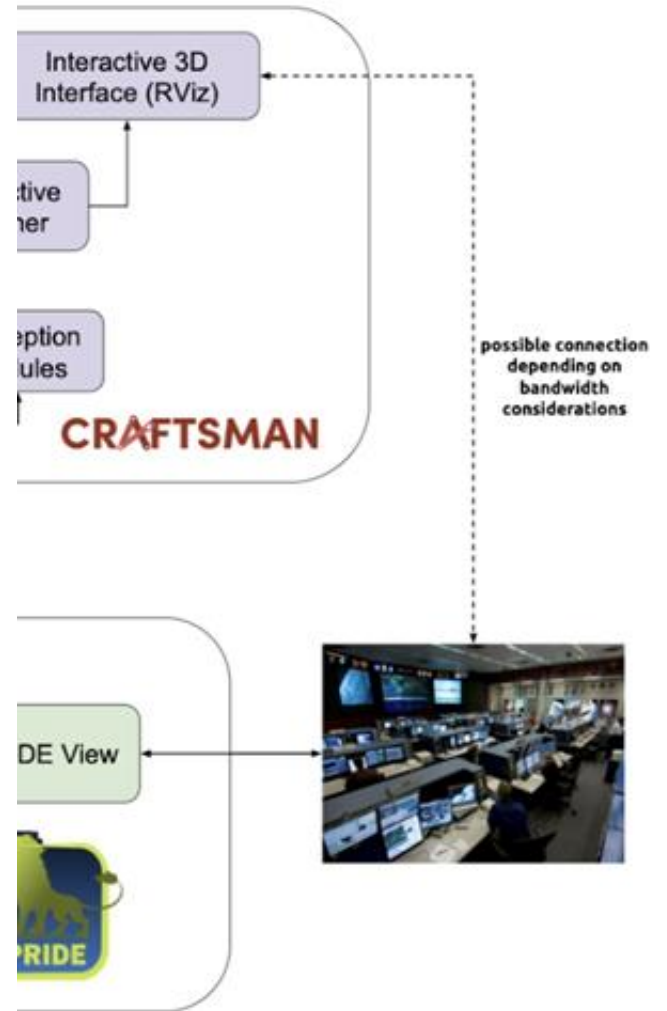
Send

Send

Send

Consolidated User Interface

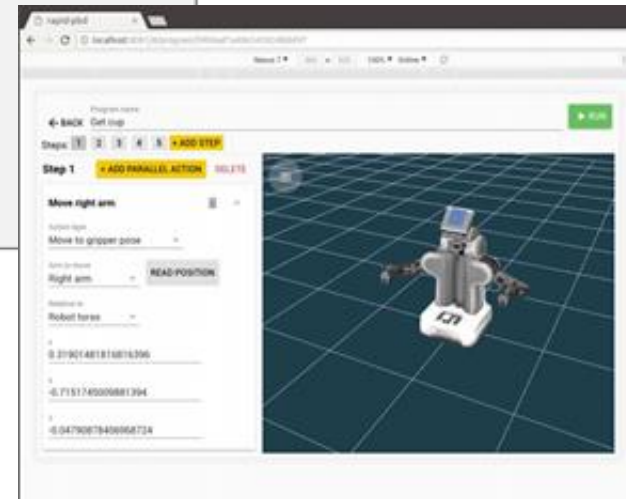
- In general, we can not presume robot operators will have a ROS-enabled Ubuntu computer for communication with their robots.
- A goal of the PHARAOH work has been to enable a more platform-independent and lightweight User Interface that provides the necessary components without sacrificing capabilities.
- Although Unreal and Unity based approaches were initially pursued, the complexity of ROS integration and multi-platform compatibility proved intractable from a maintenance and development standpoint.



Consolidated User Interface

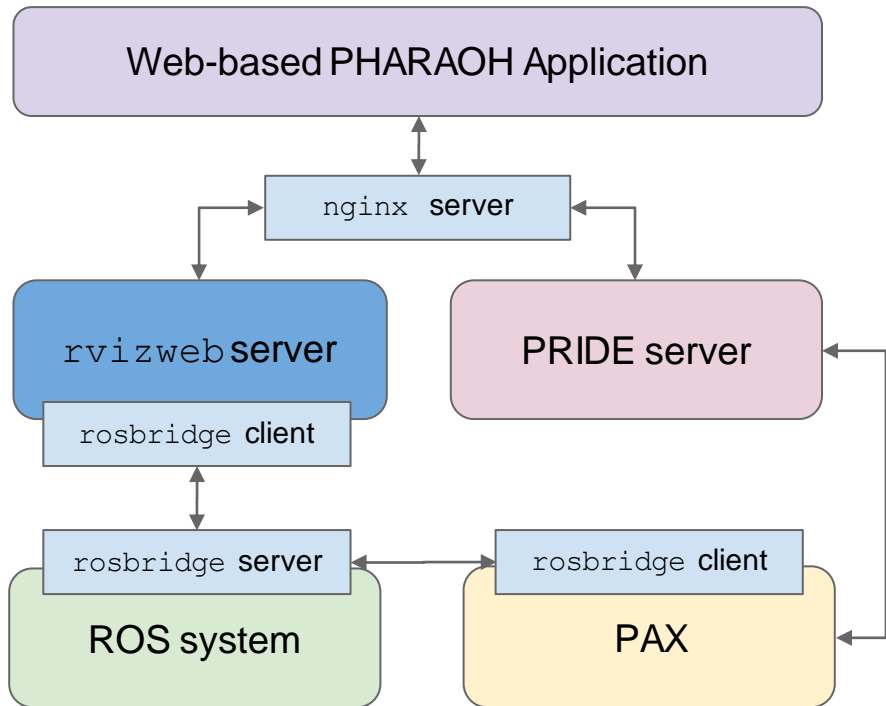
- We ultimately settled on consolidating the `rvizweb` utility with a web server that “consolidated” `PrideView` and the web-based `Rviz`.

```
{
  "globalOptions": {
    "background": "#111111",
    "colladaLoader": "collada1",
    "colladaServer": "https://localhost:443/rvizweb",
    "fixedFrame": "/rviz",
    "url": "ws://localhost:9090",
    "videoServer": "https://localhost:443/rvizweb/video"
  },
  "sidebarOpened": true,
  "displays": [
    {
      "isShown": true,
      "name": "Robot model",
      "options": {
        "param": "robot_description"
      },
      "type": "urdf"
    },
    {
      "isShown": true,
      "name": "Robot model",
      "options": {
        "param": "iss/robot_description"
      },
      "type": "urdf"
    },
    {
      "isShown": true,
      "name": "Image",
      "options": {
        "topic": "/hw/can_nav/rvizweb"
      },
      "type": "image"
    }
  ]
}
```



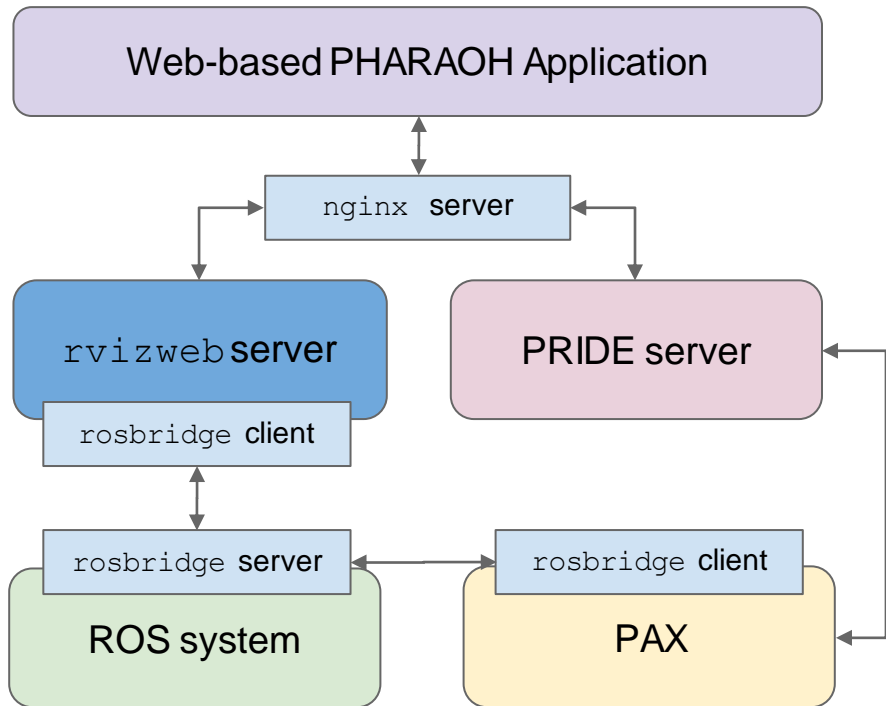
Consolidated User Interface

- We ultimately settled on consolidating the `rvizweb` utility with a web server that “consolidated” PrideView and the web-based Rviz.
- `rvizweb` uses `rosbridge` to communicate with web-based applications and provides a JSON-configurable RViz-like UI with the same core situational awareness capabilities.



Consolidated User Interface

- We ultimately settled on consolidating the `rvizweb` utility with a web server that “consolidated” PrideView and the web-based Rviz.
- `rvizweb` uses `rosbridge` to communicate with web-based applications and provides a JSON-configurable RViz-like UI with the same core situational awareness capabilities.
- Using `nginx`, we created a web-based front end that integrates, side-by-side, Pride View and `rvizweb` to provide a single PHARAOH front end.



Consolidated User Interface

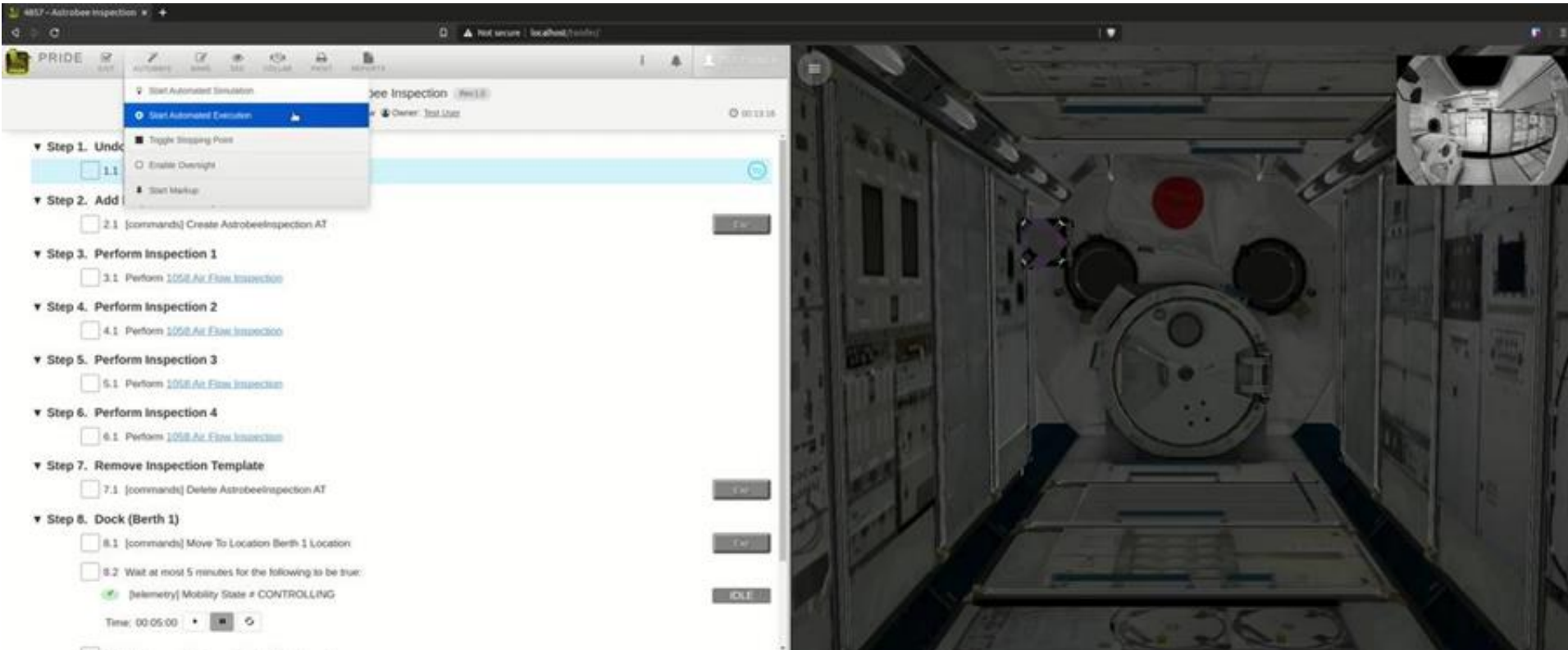
The screenshot displays a web browser window with a URL of `localhost:7000/`. The page title is "4857 - Astrobee Inspection" and it shows a "Run View" for "Owner: Test User" with a timer at "00:04:25".

The left sidebar contains a checklist of 10 steps:

- ▼ Step 1. Undock Astrobee
 - 1.1 Perform [0252 Undock](#)
- ▼ Step 2. Add Inspection Template
 - 2.1 [commands] Create AstrobeeInspection AT
- ▼ Step 3. Perform Inspection 1
 - 3.1 Perform [1058 Air Flow Inspection](#)
- ▼ Step 4. Perform Inspection 2
 - 4.1 Perform [1058 Air Flow Inspection](#)
- ▼ Step 5. Perform Inspection 3
 - 5.1 Perform [1058 Air Flow Inspection](#)
- ▼ Step 6. Perform Inspection 4
 - 6.1 Perform [1058 Air Flow Inspection](#)
- ▼ Step 7. Perform Inspection 5
 - 7.1 Perform [1058 Air Flow Inspection](#)
- ▼ Step 8. Perform Inspection 6
 - 8.1 Perform [1058 Air Flow Inspection](#)
- ▼ Step 9. Remove Inspection Template
 - 9.1 [commands] Delete AstrobeeInspection AT
- ▼ Step 10. Dock (Berth 1)
 -

The main area shows a 3D simulation of a robot (Astrobee) in a space station corridor. The robot is positioned in the center of the corridor, facing a blue door. The corridor has a dark floor with white markings and white walls with various panels. A circular inset in the top right corner shows a close-up of the robot's camera view.

PHARAOH Demonstrations



The image displays a software interface for the PHARAOH system, split into two main sections. The left section is a control panel titled "PRIDE" with a menu and a list of tasks. The right section is a 3D simulation of a spacecraft's interior, showing a central circular hatch and various equipment racks.

PRIDE Interface:

- Menu: Start Automated Simulation, Start Automated Execution (highlighted), Toggle Stopping Point, Enable Oversight, Start Marker.
- Task List:
 - Step 1. Undo
 - 1.1 [command] Undo
 - Step 2. Add
 - 2.1 [command] Create AstrobeeInspection AT
 - Step 3. Perform Inspection 1
 - 3.1 Perform 1058.Air.Flow.Inspection
 - Step 4. Perform Inspection 2
 - 4.1 Perform 1058.Air.Flow.Inspection
 - Step 5. Perform Inspection 3
 - 5.1 Perform 1058.Air.Flow.Inspection
 - Step 6. Perform Inspection 4
 - 6.1 Perform 1058.Air.Flow.Inspection
 - Step 7. Remove Inspection Template
 - 7.1 [command] Delete AstrobeeInspection AT
 - Step 8. Dock (Berth 1)
 - 8.1 [command] Move To Location Berth 1 Location
 - 8.2 Wait at most 5 minutes for the following to be true:
 - [x] telemetry Mobility State = CONTROLLING

Time: 00:05:00

3D Simulation:

The 3D simulation shows a perspective view of a spacecraft's interior. A large circular hatch is the central focus, surrounded by equipment racks and structural elements. A red circle is visible on the wall above the hatch. An inset window in the top right corner provides a different view of the interior.

← → ↻ Not secure localhost:8000/viewer/cdb0a0f9-3bc7-4a92-89cd-970bd92b74f6?mode=... ☆ 🌐 🏠

Apps Timesheets time Webex-TRAC... astrobee

PRIDE AUTOMATE SIMS OPS COLLABOR CHAT ASSISTANTS

3206 – ISS Ventilation Maintenance (Rev.13) 00:00:30

Guided View Owner: TestUser

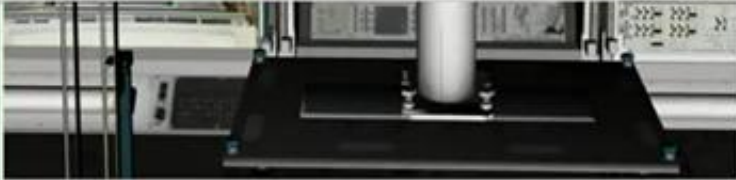


Figure 4. R2 Right Vent Scraping AT

▼ 9.5.4 Clean Up After Removal

CAUTION

Return Robonaut to its ready position

9.5.4.1 [commands] Delete sock_removal AT Done

9.5.4.2 [commands] Locate r2_helper Object backward Pending

Command Settings

Arguments


++ object_name Add a comment to this line ++: backward

Results

▼ Step 10. Go to beginning of procedure

10.1. Go To Step 4. Monitor CO2 Levels

Garbo

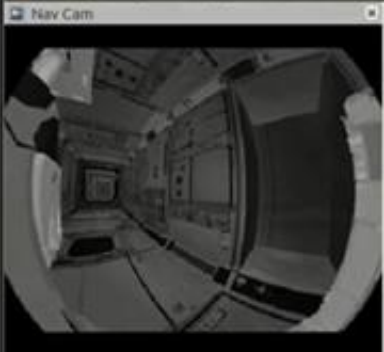


Real Time Factor: 0.93 Sim Time: 00:00:18:27.292 Real Time: 00:00:19:37.852 Iterations: 138474 FPS:

performed for: 13 h Time

pharaoh_astrobee_r2.rviz* - RViz

Nav Cam



Reset 31 fps

The End