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 Burridge presenting
TRACLabs, 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.
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.
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.
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
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

http://traclabs.com/projects/pride/
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

http://traclabs.com/projects/pride/
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
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.

```c
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.
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.
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

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
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.
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.
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
PHARAOH Demonstrations
9.5.4 Clean Up After Removal

CAUTION

Return Robonaut to its ready position

- 9.5.4.1 [commands] Delete sock_removal AT
- 9.5.4.2 [commands] Locate r2_helper Object backward

Comment Settings

Arguments

+ object_name Add a comment to this line + : backward

Results

Step 10. Go to beginning of procedure
The End