Providing Open-Source Solutions for Industry

2022 Annual Meeting
ROS-I Americas Tech Vision

- The “killer” app?

![Diagram showing process: Model → A5 → ... → Scan 'N Plan Workshop → ???]

- Why so many apps that look the same?
  - Similar approach, similar components, different requirements
  - Focus on modularity
  - Vision
    - Create a toolbox of intuitive and useful robotics modules that are easy to build into an application
Modularity at the Application Level

- Scan ‘N Plan capabilities curated into more modular repos over time
- Greater flexibility/reusability
- Greater cohesion
Modularity at the Repository

- Scan ‘N Plan Workshop/Blending M5
  - Revised architecture
  - How to create a richer, more customizable workflow?
    - 10000+ line case statement?
    - State machine?
    - Behavior Tree?
  - BT
    - 17 Custom nodes, 4 sub-trees
    - Blackboard for data transfer
    - Virtually unlimited expression capability
  - Modularity unlocks easy customization
    - Blending M5: 150 lines of new application code
Modularity at the Repository Level

- Experience with BT
  - Clear and cohesive concepts/interfaces
  - Easy to build and customize
  - Decent documentation
  - My application did not exist out of the box with BT, but it was not difficult to create
- Similar experience with several other robotics projects (Open3D, Drake, ROS2 control)
- Let’s do the same for our ROS-I tools
Modularity at the Repository

- **Tesseract (Motion Planning)**
  - Abstract interfaces for core capabilities (collision checking, IK, visualization, planning)
- **Planning Pipelines**
  - Planning capabilities as modules with inputs/outputs
  - Plugins for customization
  - Construct complex pipelines for complex tasks
  - Meta-planning
- **Visualization of results after each pipeline stage**
Modularity at the Repository

- Noether (Tool Path Planning)
  - Clear and cohesive interfaces
  - Many out-of-the-box capability modules
    - Mesh modification (primitive fitting, ROI selection)
    - Tool path planners
    - Tool path modifiers (approaches, departures, smoothing, organization)
- GUI
  - Quick pipeline configuration
  - Plugins for customization
  - Present custom front-ends to specific users
Modularity at the Repository

- Calibration
  - Improved reorganization
  - Provided utilities for
    - Common sensor calibration types
    - Validation and accuracy checks
    - Detecting targets
- Hooks for customizability
- GUI (in progress)
- Example data collection pipeline applications
Modularity at the Repository Level

- **SWORD**
  - CAD-based sandbox for experimenting with ROS-I tools
  - GUI front-end for reduced development iteration time
Modularity for Deployment

- Build on multiple platforms (Ubuntu, Windows)
- Common CI practices across repositories
- Containerization
  - Build containers per repository
    - Host-agnostic
    - Configured for network and device connection via docker-compose
  - Layer containers
  - Full system bring-up for applications with docker-compose
Takeaways

- Maybe we don’t have one “killer app” but many killer* building blocks
- *Still much work to be done
  - Clear and cohesive concepts/interfaces
  - Easy to build and customize
  - Decent documentation
- Contributions always welcome
  - Code, feedback, testing, etc
  - Consortium input needed for design effort
Thank You

Michael Ripperger

ROS-I Americas Technical Lead
Southwest Research Institute

michael.ripperger@swri.org