PACKML2: STATE MACHINE BASED SYSTEM PROGRAMMING, MONITORING AND CONTROL IN ROS2

Presenter: Dejanira Araiza-Illan, PhD
Motivation

• State machines allow...
  – Modelling sequential and concurrent processes and systems
  – Composability
  – Simplified implementation of robot control code
  – (Formal) analysis
  – IP protection through abstraction

• The PackML standard
  – State machines to model, program and control packing processes
  – Standard template with states, transitions and triggering events
Background

• PackML ROS package
  – Released in 2016-2017
  – RViz plugin (Indigo)
  – Qt GUI (Kinetic)
  – State machine simulator in C++

• Existing state machine libraries previously adopted in ROS
  – SMACH (Python) [http://wiki.ros.org/smach](http://wiki.ros.org/smach)
  – Lifecycle (C++, ROS 2 Crystal and newer) [https://github.com/ros2/demos/tree/master/lifecycle](https://github.com/ros2/demos/tree/master/lifecycle)
Methodology for Implementation

1. Porting PackML (Kinetic) packages

2. PackML use case

3. PackML ROS 2 *lifecycle* package (standard PackML state machine)

4. Comparison of PackML ROS 2 and first ROS implementation
1. Porting PackML Kinetic Packages to Dashing

- packml_msgs
  - Transition.srv
  - AllStatus.srv
  - ...

- packml_sm
  - Continuous Execute state
  - Time limited Execute state

- packml_plugin
  - Transition service server
  - Status service server

- packml_ros
1. Porting PackML Kinetic Packages to Dashing

**Issues**

- Lack of documentation and examples in ROS 2
- Syntax changes in ROS API, CMakeLists.txt and package.xml
- Intertwined ROS code
- Tests that fail and only for the state machine library

**Lessons Learned**

- Documenting and sharing the code for the future
- Separation of libraries without ROS content vs. ROS nodes
- Modularity, functions, classes, package structure
- New unit testing implemented, for all the code
2. PackML Use Case

PackML state machine in ladder code

Siemens PLC

ROS 2 Dashing

OPCUA over TCP/IP

packml_plc

ROS 2 OPCUA client in Python 3
3. PackML ROS 2 lifecycle package

- packml_lifecycle_msgs
  - State.msg
  - Transition.msg
  - ...

- packml_lifecycle
  - Node with PackML SM structure in C++

- packml_rcl_lifecycle
  - PackML SM skeleton structure in C
4. PackML ROS 2 vs PackML ROS

- No more *roscore* overhead

  *In Melodic*: 327564K (Core) + 833636K (Master) + 344528K (Logger) + 504492K (Node) + 1481356K (Qt GUI) ~ 3.5GB

  *In Dashing*: 617824K (Node) + 1864192K (RViz plugin) ~ 2.5GB

- No more topics, only services

- Visualization of state machine state and elapsed time per state

- More code unit testing (>80% LOC)
THANK YOU

PackML2 will be released open-source soon!

For more information, please contact me:
Dejanira Araiza-lllan
dejanira_araiza@artc.a-star.edu.sg