

# Internship Presentation

Brian Chen

Open Robotics

August 18, 2022

## 1 What I did

- Service Introspection
  - Demo
- Other
- Things I will get to :)

## 2 Retrospective

## 3 Questions

# What I did



Figure: well, sort of

# Service Introspection (ros2/ros2#1285, REP2012)

- Or: finally being able to take a peek into what services are doing.
- Context: This has *not* been possible since the introduction of ROS1 services in ~2011

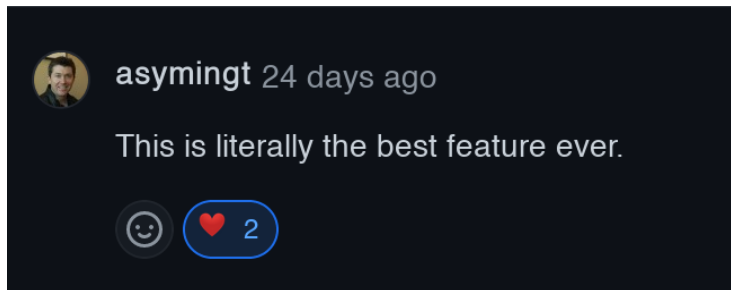


Figure: It appears that service introspection is useful. <sup>1</sup>

---

<sup>1</sup>Personal note: Lack of this feature a major reason why we don't use services at aUToronto (university design team I'm a part of)

# Features

- Service events are published to `<service_name>/_service_event`
  - ▶ service event messages contain
    - ★ ServiceEventInfo: metadata about the event
    - ★ The request/response message
- `ros2 service echo`
- `ros2 bag record --services`
- Runtime configuration of service introspection via ROS parameters
  - ▶ `<service_name>.publish_service_events`
  - ▶ `<service_name>.publish_service_content`
  - ▶ ...
- Naturally supports actions (!)

# How do I opt in?

```
auto node = rclcpp::Node::make_shared("my_client",  
    rclcpp::NodeOptions().enable_service_introspection(true));
```

Figure: cpp

```
class MyNode(Node):  
    def __init__(self):  
        super().__init__('my_node', enable_service_introspection=True)
```

Figure: python

— Demo —

# Implementation

- REP2012
- rcl support
  - ▶ Hook into services/clients in rcl to publish service events
- rosidl
  - ▶ Generate definitions for new `service_msgs`
  - ▶ Offloading `service_msg` instantiation to typesupport libraries
    - ★ Can't fill out message structs with non-POD members in rcl
  - ▶ Fixing some dependency loops
    - ★ cycle: `rosidl <-> service_msgs`
    - ★ bonus: no longer need to `<depend>action_msgs</depend>`
- Client library integration
  - ▶ Opt-in/opt-out (opt-out default)
  - ▶ Expose ROS parameters for runtime configuration of service introspection behaviour
- Small changes in `rmw` and some downstream packages
- `ros2 service echo`
- `ros2 bag record --services`



# Challenges

- Serialize vs non-serialized?
- `rosidl`
- Dependency cycles

# What's next?

- PRs need iteration and review
- Documentation/demos/tutorials
- Security & sros2
- Give actions some love

## Other things I did

- `rclpy AsyncParameterClient`
  - ▶ `ros2param` refactor to use `AsyncParameterClient`
  - ▶ added a demo
- Miscellaneous VIPER tasks: ROS2/Gazebo maintenance
  - ▶ Gazebo sdf for stereo camera projection matrix (calculating P matrix WIP)
  - ▶ Fix deadlock in `rclpy.shutdown` when being called from a callback
  - ▶ Various `ros2cli` changes
    - ★ Setting multiple parameters
- Opt-in QOS overrides for `image_common` publisher <sup>2</sup>
- launch fixes in `stereo_image_proc`, `gzserver`
- reduce `rosviz` message spam
- ...

---

<sup>2</sup>See: Figure 1

# Things I will get to :)

- vim syntax highlighting for em files

```
@# emSimpleExpression
@simple_expression
(
| @event_type
)

@# emReprExpression
@'repr_expression'

@# emExpressionWithDummy
@:2 + 2:this will get replaced with 4:

@# emStatement
@{", ".join([i for i in range(100)])}@
@{", ".join([i for i in range(100)])}

@# emSignificator
@%keywords 'physics', 'gravity', 'Einstein', 'relativity'
```

- Fixing type annotations in rclpy
- colcon printing dependency loops

# Retrospective

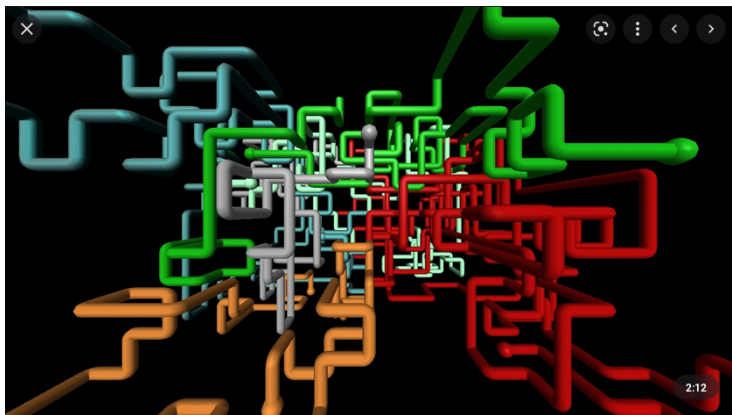


Figure: Plumbing

# Retrospective

- Should've thought through the serialized/non-serialized decision a little more
- Communication is important
- Working with non-research-grade code
- Rubber ducky
- I had a good time<sup>3</sup>

---

<sup>3</sup>And I hope I (excuse the corporate-speak) produced some value while doing so

# Thanks

Thanks to Jacob & Open Robotics for the opportunity

# Questions

## Presentation TLDR:

- Service introspection
  - ▶ (`ros2 service echo`, `ros2 bag record --services`)
  - ▶ Implementation details and challenges
  - ▶ demo
- various PRs & things I will get to
- AsyncParameterClient, maintenance tasks, small features
- Retrospective: plumbing, serialize/non-serialize, communication, non-research grade, rubber ducky, had a good time