

An Experience Report on Challenges in Learning the Robot Operating System

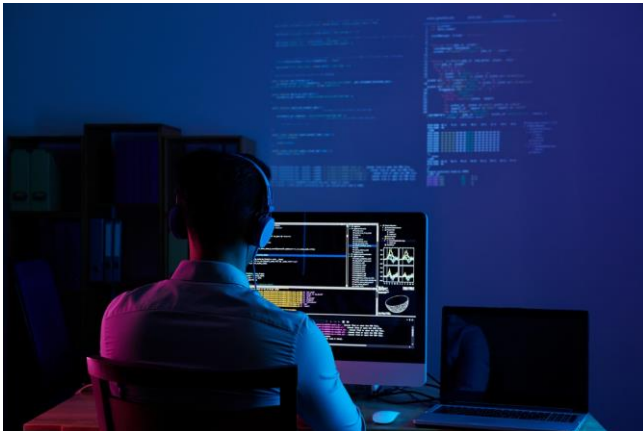
Paulo Canelas^{1,2}, Miguel Tavares¹, Ricardo Cordeiro¹, Alcides Fonseca¹, Christopher S. Timperley²

¹LASIGE, Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa

²Institute for Software Research, School of Computer Science, Carnegie Mellon University

The Robot Operating System

ROS



CMU Robotics Institute Summer Scholars (RISS) program

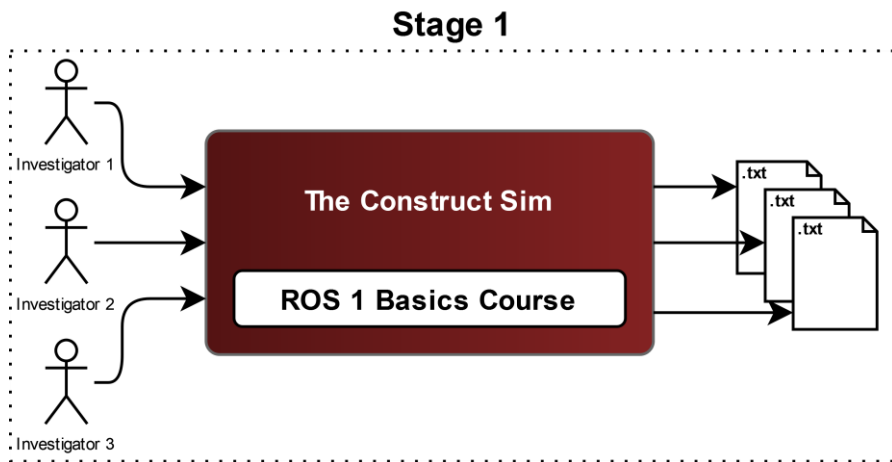


“ROS offers a standard software platform to **developers across industries** that will carry them from **research and prototyping** all the way through to deployment and production.”

Understand the **challenges** of newcomers when learning the Robot Operating System

Methodology

The Investigators



Paulo Canelas

Ph.D. Student

No previous experience with robotic systems.



Miguel Tavares

MSc. Student

Experience with Thymio [1].



Ricardo Cordeiro

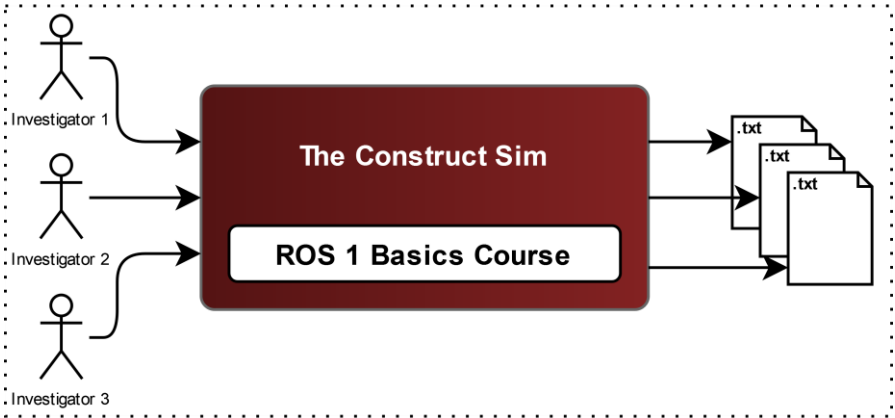
MSc. Student

No previous experience with robotic systems.

[1] Fanny Riedo. 2015. Thymio a holistic approach to designing accessible educational robots. (2015).

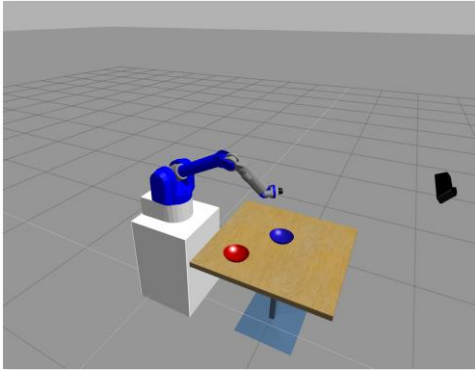
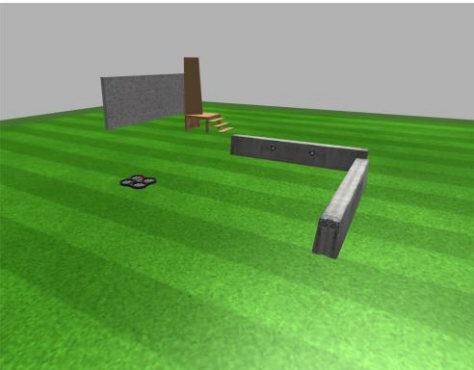
ROS 1 Basics Course from The Construct Sim

Stage 1



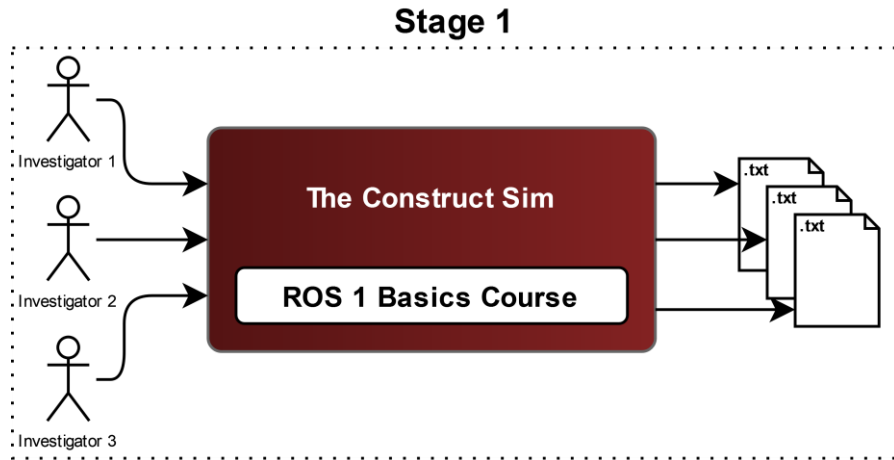
Course Summary

Introduction	📄
ROS Deconstruction	📄
ROS Basics	📄
Understanding ROS Topics - Publishers	📄
Understanding ROS Topics - Subscribers & Messages	📄
Understanding ROS Services - Clients	📄
Understanding ROS Services - Server	📄
Using Python Classes in ROS	📄
Understanding ROS Actions - Clients	📄
Understanding ROS Actions - Servers	📄
How to Debug ROS Programs	📄
Appendix	📄

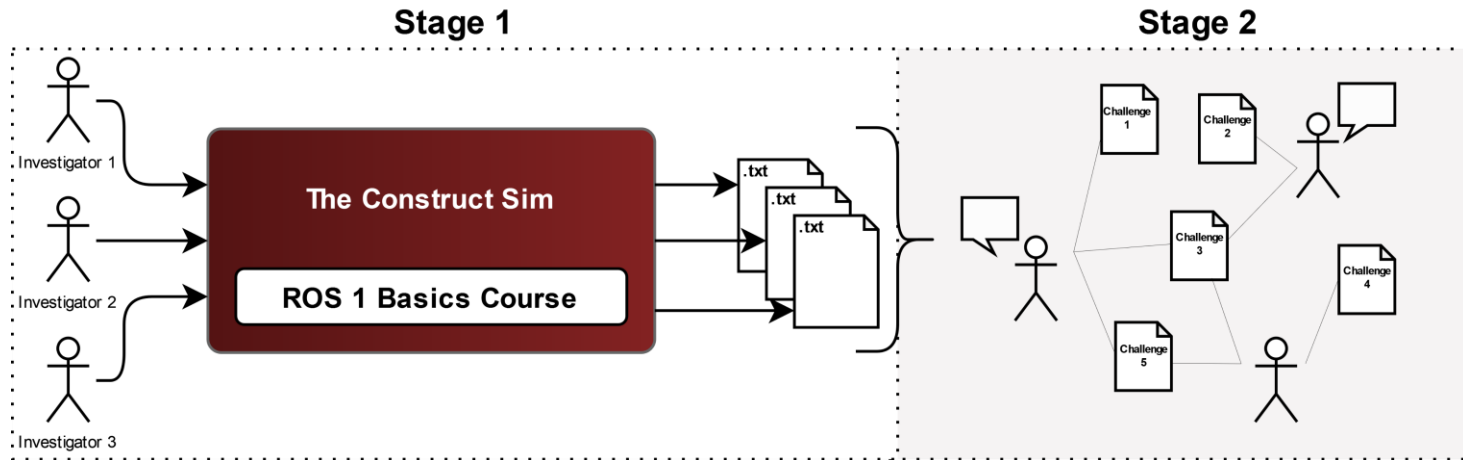


```
File Edit View Help
...
1 #! /usr/bin/env python
2
3 import rospy
4 import rospy
5
6 from trajectory_msgs.msg import Trajectory, TrajectoryRequest, TrajectoryResponse
7
8 # Initialize the node
9 rospy.init_node('service_client_trajectory')
10
11 # Create the client object
12 client = rospy.ServiceClient('rospy_get_service', 'rospy_get_service')
13
14 # Create the service we are sending a request to
15 service_name = 'rospy_get_service'
16
17 # Create the request object
18 request = TrajectoryRequest()
19
20 # Send the request and process the answer
21 while not rospy.is_shutdown():
22     request = client.call(service_name, request)
23     print(request)
```

Annotation of the Challenges

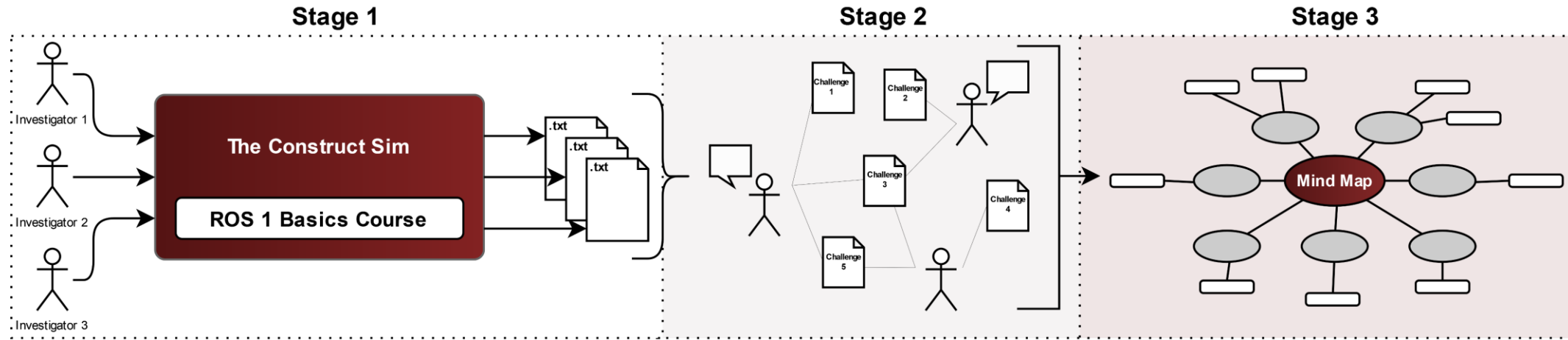


Adjudication and Discussion

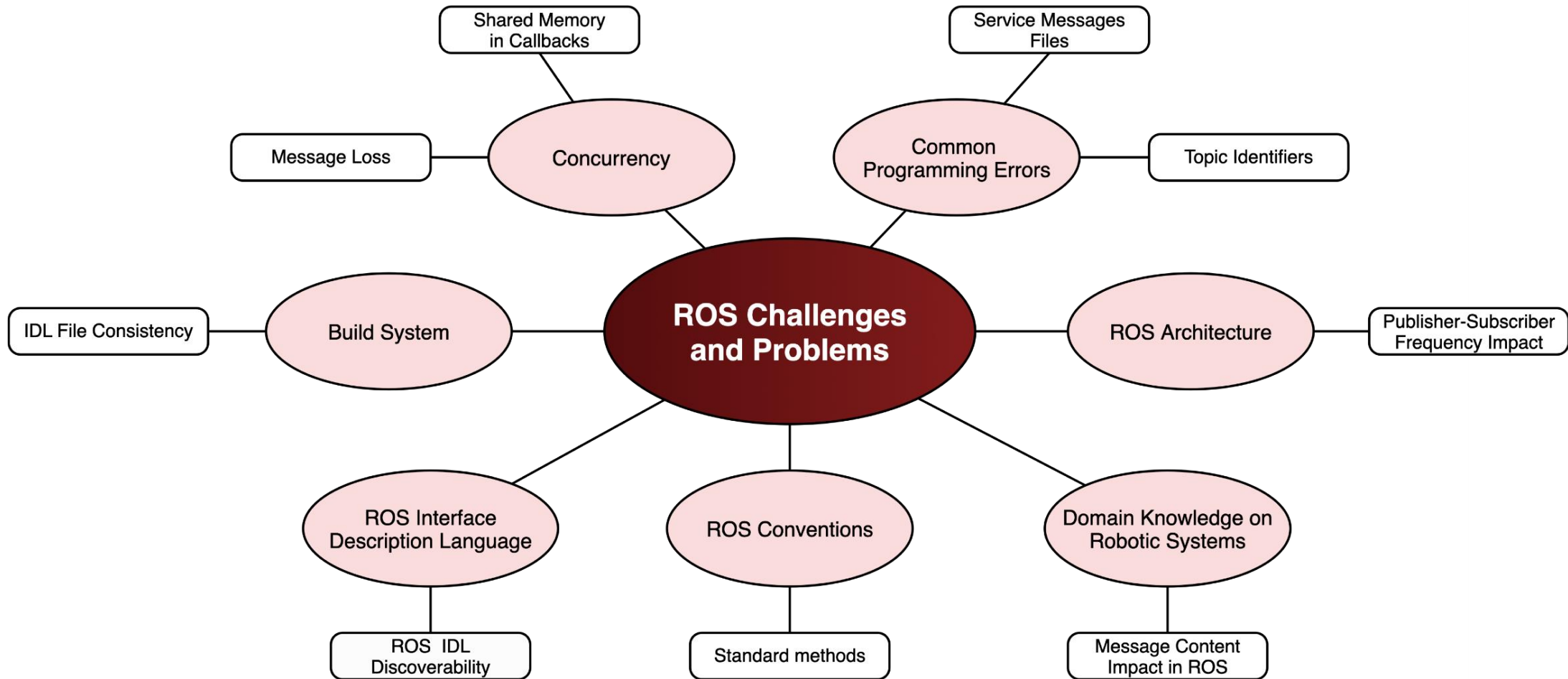


The unorganized notes are categorized and the investigators discuss the shared challenges.

Creation of the Mind Map



We have identified seven high-level challenges



The Challenges

1. IDL File Consistency

Build System

Consistency is required between the multiple configuration and implementation files.

```
<launch>
  <node pkg="module5_6_pkg"
        type="client.py"
        name="service_client"
        output="screen" />
</launch>
```

```
add_service_files(
  FILES
  DurationServiceMessage.srv)
```

```
import rospy
from module5_6_pkg.srv import DurationServiceMessage,
                             DurationServiceMessageResponse

rospy.init_node('service_client')
service = rospy.Service('/myservice', DurationServiceMessage,
                        my_callback)
```

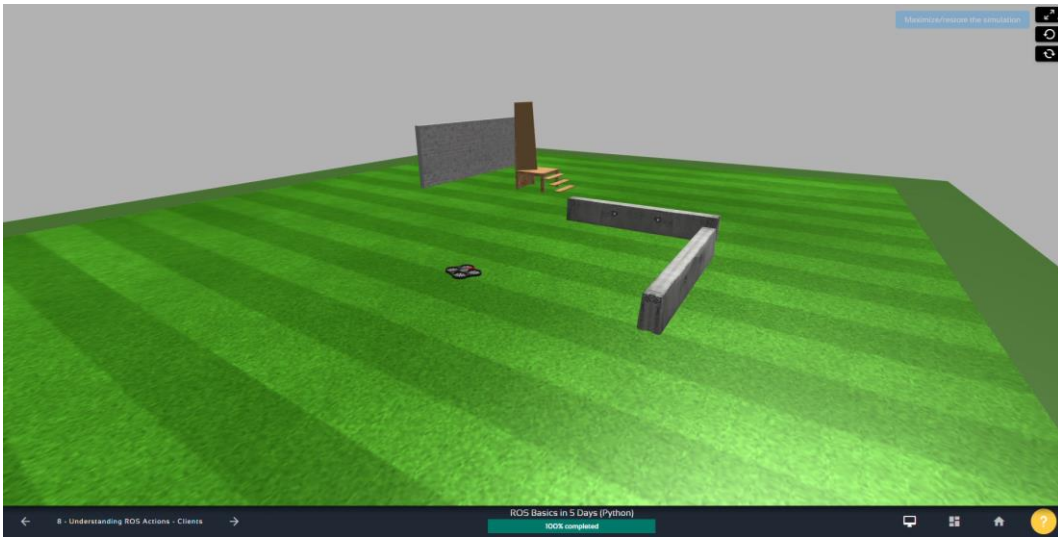


Common not to include dependencies and mistype configurations

2. ROS IDL Discoverability

ROS Interface Description Language

Description of the available topics and their content.



Which topic is responsible for the drone position?

```
user:~/catkin_ws$ rostopic list
/camera_info
/clock
/cmd_vel
/drone/down_camera/image_raw
...
/drone/gt_acc
/drone/gt_pose
/drone/gt_vel
/drone/imu
/drone/land
/drone/posctrl
/drone/reset
/drone/sonar
/drone/takeoff
/drone/vel_mode
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
/rosout
/rosout_agg
```



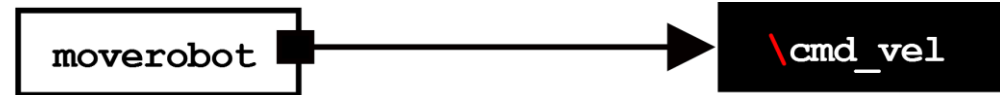
Lack of information about the topics and their content lead to trial and error searching for the wanted information

3. Topic Identifiers

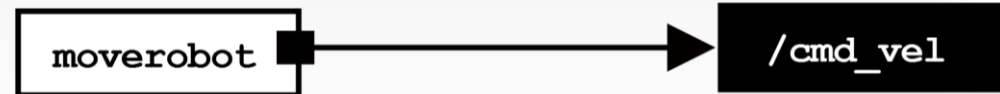
Common Programming Errors

The most common error by the investigators was the **mistyping of topic names**.

```
rospy.init_node('moverobot')  
pub = rospy.Publisher('\cmd_vel', Twist, queue_size=1)
```



```
rospy.init_node('moverobot')  
pub = rospy.Publisher('/cmd_vel', Twist, queue_size=1)
```



Time-consuming identifying what and where the problem is

4. Publisher-Subscriber Frequency Impact

ROS Architecture

Define the adequate **publishing rate** and **queue size**.

Challenge 1

What is the proper queue size?

Challenge 2

What is the proper publishing rate?

- ❖ Dependency between the queue size and the publishing rate.
- ❖ The wrong configuration combination can lead to unintended robot behavior.

```
# Create the publisher
pub = rospy.Publisher('/cmd_vel', Twist, queue_size=1)

# Create the message
message = Twist()
message.linear = Vector3(0.5, 0, 0)

# Define the rate
rate = rospy.Rate(10)

# Publish the speed at fixed rate of 10 Hz
while not rospy.is_shutdown():
    message.linear.x += 0.01
    pub.publish(message)
    rate.sleep()
```

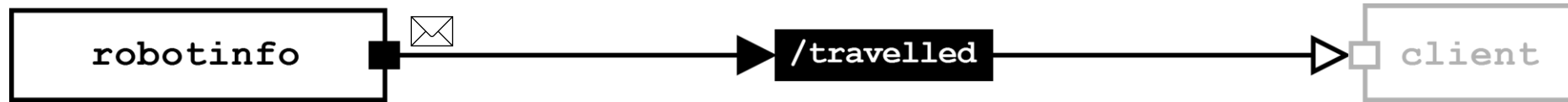


No perception on the impact queue sizes, and rates can have in the robot

5. Message Loss

Concurrency

Loss of messages when publishing before the subscriber is listening.



If the connection is not *latched*, the order in which the subscriber and publisher are initiated matters.

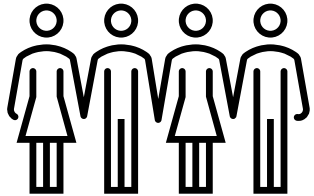


Time-consuming task pinpointing the origin of the issues

What is next?

What is next?

Usability Studies



- ❖ Help design in-depth usability studies with larger groups.

Documentation Improvement

- ❖ Encourage the improvement of the documentation:
 - ❖ Component's interface;
 - ❖ Intended communication model;
 - ❖ Frequency;
 - ❖ Bounds on values of messages.

Improvement of Verification Techniques

ROSDISCOVER



Architectural Robot and System Verification

- ❖ Analysis of the robot's system and architecture [3].
- ❖ Architectural specification of the system [4].

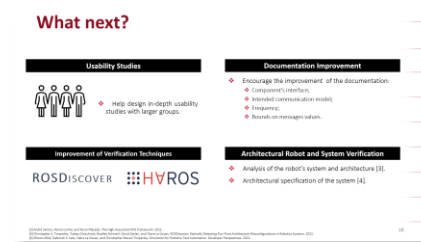
[2] André Santos, Alcino Cunha, and Nuno Macedo. The High-Assurance ROS Framework. 2021.

[3] Christopher S. Timperley, Tobias Dürschmid, Bradley Schmerl, David Garlan, and Claire Le Goues. ROSDiscover: Statically Detecting Run-Time Architecture Misconfigurations in Robotics Systems. 2022.

[4] Afsoon Afzal, Deborah S. Katz, Claire Le Goues, and Christopher Steven Timperley. Simulation for Robotics Test Automation: Developer Perspectives. 2021.

An Experience Report on Challenges in Learning the Robot Operating System

Understand the **experience** of newcomers when learning the Robot Operating System



Contact me: pacsantos@fc.ul.pt

This work was supported by *Fundação para a Ciência e Tecnologia* (FCT) in the LASIGE Research Unit under the ref. (UIDB/00408/2020 and UIDP/00408/2020), and the CMU-Portugal Dual Degree PhD Program (SFRH/BD/151469/2021), by the CMU-Portugal project CAMELOT, (LISBOA-01-0247-FEDER-045915), the RAP project under the reference (EXPL/CCI-COM/1306/2021), and the U.S. Air Force Research Laboratory (\OSR-4066).

The authors are grateful for their support. Any opinions, findings, or recommendations expressed are those of the authors and do not necessarily reflect those of the US Government.