An Experience Report on Challenges in Learning the Robot Operating System

Paulo Canelas ¹,², Miguel Tavares ¹, Ricardo Cordeiro ¹, Alcides Fonseca ¹, Christopher S. Timperley ²

¹ LASIGE, Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa
² Institute for Software Research, School of Computer Science, Carnegie Mellon University
“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.

ROS 1 Basics Course from The Construct Sim

Stage 1

The Construct Sim

ROS 1 Basics Course

Course Summary

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

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

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

```python
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?

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**.

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

---

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

Time-consuming identifying what and where the problem is
# 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()
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**

- **ROSDicover**
- **HAROS**

**Architectural Robot and System Verification**

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

---

An Experience Report on Challenges in Learning the Robot Operating System

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.

Contact me: pacsantos@fc.ul.pt