ROBOT OPERATING SYSTEM (ROS)

M. Sc. Mihaela Popescu
Dipl.-Inf. Andreas Bresser
Robotics Innovation Center
DFKI Bremen

Prof. Dr. Dr. h.c. Frank Kirchner
Arbeitsgruppe Robotik, Universität Bremen
`https://robotik.dfki-bremen.de/`
`robotics@dfki.de`

26th October, 2022 – Bremen, Deutschland

Overview

ROS in the Robot Design Lab

Universität Bremen

ROS Introduction

ROS Introduction

Storytime

comic by `https://phdcomics.com`

ROS Introduction

Storytime

comic by `https://phdcomics.com`

ROS Introduction

Storytime

comic by https://phdcomics.com

ROS Introduction

Storytime

comic by https://phdcomics.com

ROS Introduction

Storytime

comic by https://phdcomics.com

ROS Introduction

Storytime

comic by https://phdcomics.com

ROS Introduction

Storytime

German Research Center for Artificial Intelligence

comic by https://phdcomics.com

Universität Bremen

ROS Introduction

Storytime

comic by `https://phdcomics.com`

ROS Introduction

Storytime

comic by https://phdcomics.com

Universität Bremen

ROS Introduction

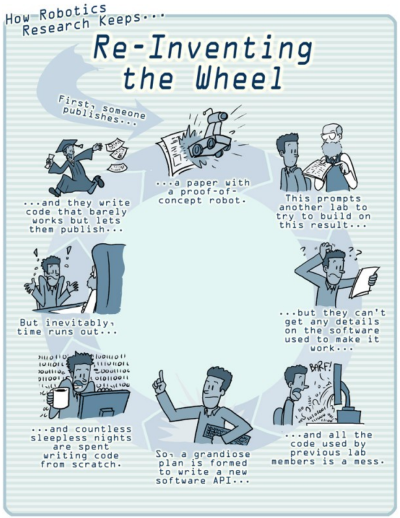
What's the problem?

German Research Center for Artificial Intelligence

Solutions?

⇒ How to break the circle?

comic by `https://phdcomics.com`


How Robotics Research Keeps...
Re-Inventing the Wheel
First, someone publishes...
...a paper with a proof-of-concept robot.
This prompts another lab to try to build on this result...
...but they can't get any details on the software used to make it work...
BARF!
...and all the code used by previous lab members is a mess.
So, a grandiose plan is formed to write a new software API...
...and countless sleepless nights are spent writing code from scratch.
But inevitably, time runs out...
...and they write code that barely works but lets them publish...

Universität Bremen

ROS Introduction
What's the problem?

German Research Center for Artificial Intelligence

Solutions!

- Standardize components (that can be documented),
- build generic tools,
- split the work to multiple people by building a community and
- make the industry part of this community.

comic by `https://phdcomics.com`

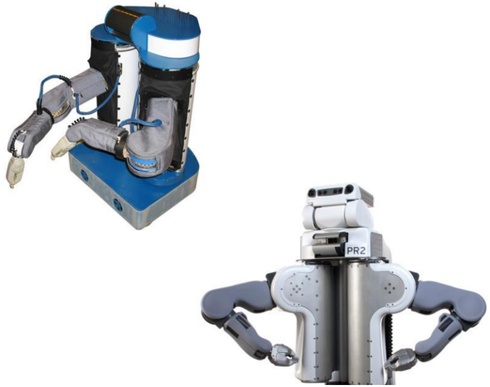
How Robotics Research Keeps...
Re-Inventing the Wheel
First, someone publishes...
...a paper with a proof-of-concept robot.
This prompts another lab to try to build on this result...
...but they can't get any details on the software used to make it work...
BARF!
...and all the code used by previous lab members is a mess.
So, a grandiose plan is formed to write a new software API...
...and countless sleepless nights are spent writing code from scratch.
But inevitably, time runs out...
...and they write code that barely works but lets them publish...

Universität Bremen

ROS Introduction
Standing on the shoulders of giants

Evolution of ROS

- Willow Garage (2008-2013)
  - Building libraries for the community
- OSRF/Open Robotics (2014+)
  - 2000+ packages, worldwide adaption
- ROS 2 (since 2016)
  - New middleware with deep changes



PR-1 (top) by Stanford University and PR-2 by Willow Garage, image source: `https://robots.ieee.org/robots/pr2/`

ROS Introduction

What is ROS?

Not an Operating System but a **Middleware**

ROS
=
Plumbing
+
Tools
+
Capabilities
+
Ecosystem

Image source: `https://www.ros.org/about-ros/`

ROS Introduction

Examples for very different systems using ROS

ROS 1 systems at DFKI-RIC

Cuttlefish — AUV with 2 arms

ARTER — Autonomous Rough Terrain Excavator Robot

Mobipick — Mobile Manipulator a MiR 100 base and UR5 arm

Source: `https://robotik.dfki-bremen.de/en/research/robot-systems.html`

ROS Introduction
DFKI RIC Active Systems

German Research Center for Artificial Intelligence

Aktives Zweiarm-Exoskelett

ARTEMIS

ARTER

ASGUARD II

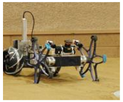
ASGUARD IV

AUV Cuttlefish

AUVx

Charlie

COMPI


Coyote II

Coyote III

CREX

DAGON

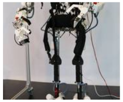
DeepLeng

EO smart connecting car 2

Exoskelett aktiv (Capio)

Exoskelett Passiv (CAPIO)

Flatfish


Ganzkörperexoskelett

Innok Heros

KUKA KR 60

Magnet Crawler II

MANTIS

Mitsubishi PA 10-7C

Mobipick

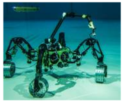
Orion

SeaBotix LBV 150

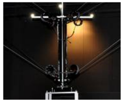
SeeGrip Manipulator

SeekurJr

SherpaTT


SherpaUW

SpaceClimber


SpiderCam

Teredo IceShuttle

TIAGo

YEMO 1.1

Source: `https://robotik.dfki-bremen.de/en/research/robot-systems.html`

ROS Introduction

ROS distributions (End-Of-Life dates in brackets)

German Research Center for Artificial Intelligence

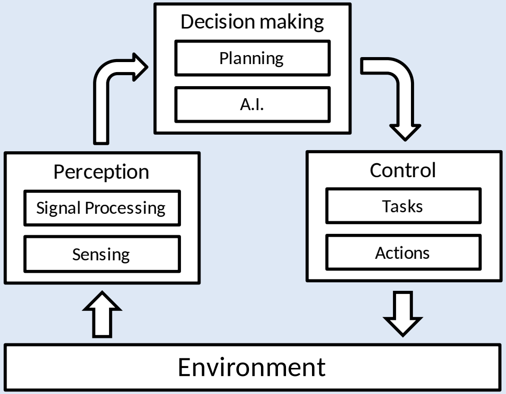
Ubuntu	ROS 1 distribution	ROS 2 distribution	Release
22.04 LTS		🟩 **Humble** Hawksbill (05/2027)	2022
20.04 LTS		**Galactic** Geochelone (11/2022)	2021
	🟩 **Noetic** Ninjemys (05/2025)	🟩 **Foxy** Fitzroy (05/2023)	2020
18.04 LTS		**Eloquent** Elusor	2019
		Dashing Diademata	
	🟩 **Melodic** Morenia (05/2023)	**Crystal** Clemmys	2018
		Bouncy Bolson	
17.04	**Lunar** Loggerhead		2017
16.04 LTS	**Kinetic** Kame	**Ardent** Apalone	2016

🟩 currently supported distributions

Topology of a Robot

German Research Center for Artificial Intelligence

The Robotics problem...



Distributed processes:

- Sensor data sampling
- Data/signal processing
- Decision making
- Control signals $\rightarrow$ actuators

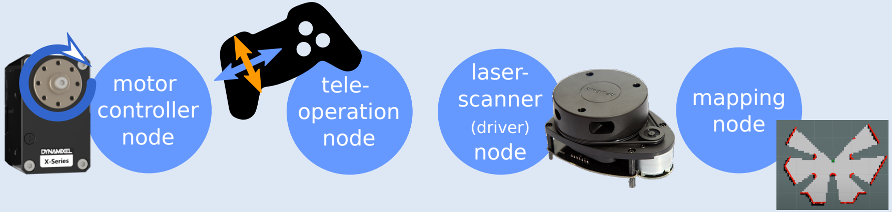
ROS Concepts

Nodes

What is a ROS Node?

Nodes are processes.

A node is a small Python or C++ program that should be responsible for a single, module purpose. Examples of nodes:

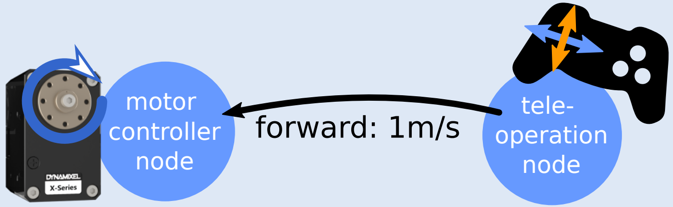


Communication

How do Robot components communicate?

Nodes exchange data → interprocess communication.

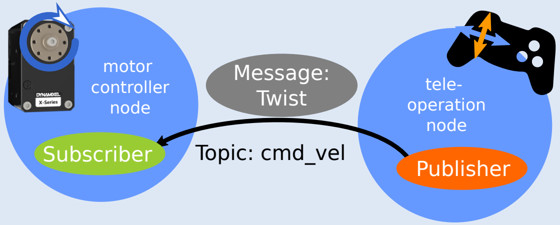
A simple example:



- teleop-node gets data from the joystick, calculates speed for the robot
- teleop-node sends calculated velocity, the motor controller-node receives it
- motor controller-node calculates how fast the motor has to turn to move the robot and sends the translated speed to the motor.

ROS Topics

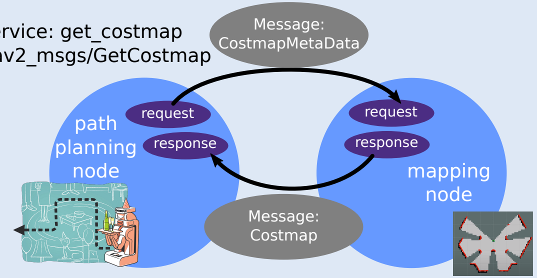
Nodes can use topics to communicate. Use topics to send fast and repeating data (for example sensor readings).



- the teleop-node publishes a Twist message on the `cmd_vel` topic.
- the motor controller-node subscribes to the `cmd_vel` topic and receives it.
- other nodes can also talk or listen to `cmd_vel`!

Communication

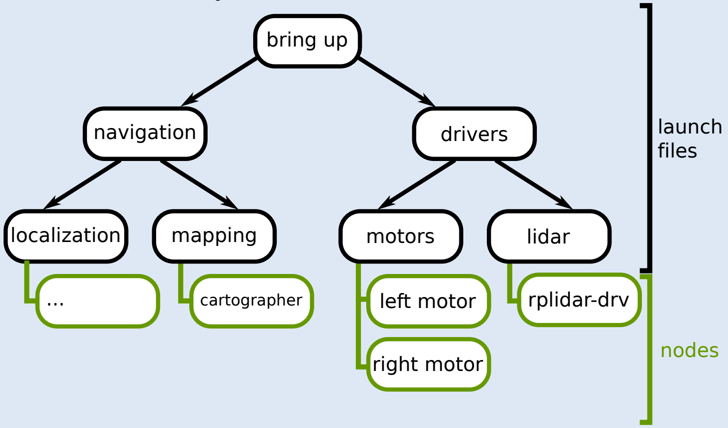
Services



- Topic-Problem: You don't know if your message got received.
- ROS services force you to send feedback with your message!
- Multiple nodes can call the same service, but only one node can provide one!

Launch Files

Launch files allow you to start other launch files and nodes.



ROS 2 Commands

Examples of ROS 2 commands that can be executed through the command line interface (CLI):

- `colcon build`: Build all packages in the workspace.

ROS 2 Commands

Examples of ROS 2 commands that can be executed through the command line interface (CLI):

- `colcon build`: Build all packages in the workspace.
- `ros2 node list`: Outputs a list of available nodes.

ROS 2 Commands

Examples of ROS 2 commands that can be executed through the command line interface (CLI):

- `colcon build`: Build all packages in the workspace.
- `ros2 node list`: Outputs a list of available nodes.
- `ros2 topic list`: Outputs a list of active topics.

ROS 2 Commands

Examples of ROS 2 commands that can be executed through the command line interface (CLI):

- `colcon build`: Build all packages in the workspace.
- `ros2 node list`: Outputs a list of available nodes.
- `ros2 topic list`: Outputs a list of active topics.
- `ros2 interface show <message-type>`: Outputs the interface definition.

ROS 2 Commands

Examples of ROS 2 commands that can be executed through the command line interface (CLI):

- `colcon build`: Build all packages in the workspace.
- `ros2 node list`: Outputs a list of available nodes.
- `ros2 topic list`: Outputs a list of active topics.
- `ros2 interface show <message-type>`: Outputs the interface definition.
- `ros2 run <package> <executable>`: Runs an executable (node).

ROS 2 Commands

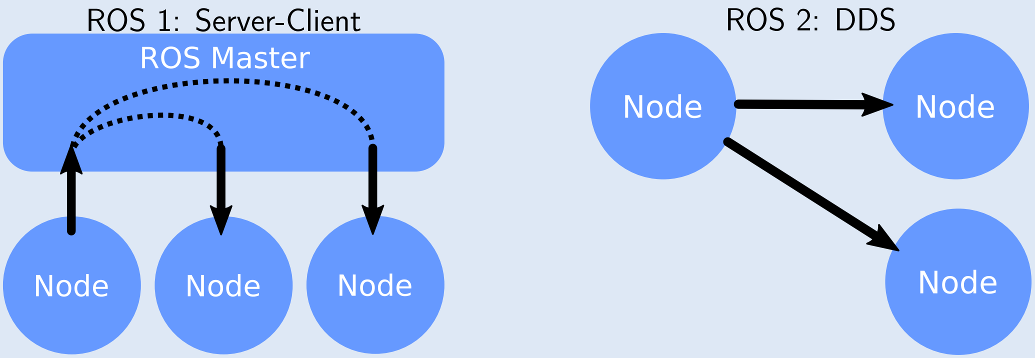
Examples of ROS 2 commands that can be executed through the command line interface (CLI):

- `colcon build`: Build all packages in the workspace.
- `ros2 node list`: Outputs a list of available nodes.
- `ros2 topic list`: Outputs a list of active topics.
- `ros2 interface show <message-type>`: Outputs the interface definition.
- `ros2 run <package> <executable>`: Runs an executable (node).
- `ros2 launch <package> <launch-file>`: Runs a launch file.

ROS 1 and ROS 2

ROS 1 and ROS 2

Differences in Communication



Data flow in ROS 1 and ROS 2

ROS 1 and ROS 2
Different tools and commands

Build tools

- ROS 1 uses `catkin_make` or `catkin build`
- ROS 2 uses **`colcon`**

Command line interfaces (CLI)

- ROS 1 has separated commands like `roslaunch`, `rostopic`, ...
- ROS 2 commands are run with `ros2` followed by a space, like **`ros2 launch`** or **`ros2 topic`**.

ROS 1 and ROS 2
Differences in programming

Starting nodes
- ROS 1 launch files are written in XML.
- ROS 2 launch files are written in **Python**. (you can still force XML)

Developing nodes
- The Python 2 API for ROS 1 is `rospy`.
- The Python **3** API for ROS 2 is **`rclpy`**.

ROS 1 and ROS 2

Getting help

ROS Wiki

- ROS 1 Wiki: `https://wiki.ros.org/`
- ROS 2 Documentation: `https://docs.ros.org/en/humble/`

Conclusions and Further Reading

Takeaways
What do we know about ROS now?

ROS is awesome

- ROS solves basic problems in robotics.
- It helps you to get data from a node to another easily.
- ROS has a lot of graphical and command line tools.
- ROS has a very useful command line interface.
- Be aware that ROS 1 is still huge!

$\Rightarrow$ You will get hands-on-experience in the tutorials!

Further learning

Play around with ROS!

German Research Center for Artificial Intelligence

Web

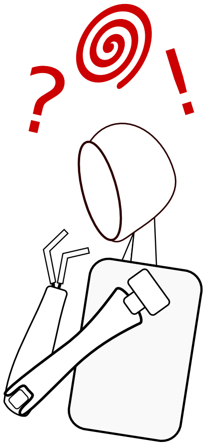
- ROS 2 docs: `https://docs.ros.org/en/humble/`
- Books about ROS at `https://wiki.ros.org/Books`
- ROS forum: `https://answers.ros.org/`

Final Notes

German Research Center for Artificial Intelligence

Don't Panic!

- Ubuntu, ROS, Python, LaTeX, git . . .
  So much new Software! All this in the 1st semester?!
- Robotics is a huge field with a lot of active research.
- This course can only give a very superficial overview.
- Ask questions! We are happy to help!



Universität Bremen

Next: Programming in Python