# Ros Basic - Concepts

Federico Sarrocco

May 2021

## 1 Introduction

**Robot Operating System(ROS)** is not a full fledged operating system, it is a "meta operating system". It is built on top of a full operating system. It is called an OS because it also provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management.

ROS does not replace, but instead works alongside a traditional operating system

- Distributed computation
- Software reuse
- Rapid testing

## 2 FileSystem

ROS packages are saved in a catkin "workspace" folder. The package folders are saved in the "src" folder in the catkin "workspace".

## 3 Packages

**Ros Package** - contains libraries, executables, scripts, and other artifacts for a specific ROS program. Packages are used for structuring specific programs. Files in a package also have a specific structure. A ROS package folder could contain:

- **launch folder** - it contains the launch files(launch files are used to run multiple nodes).
- **src folder** - it contains the source files for instance python or c++ files.
- **package.xml** - also called manifest file, contains package metadata, dependencies, and other metadata related to the package.

- **CMakeLists.txt** -It is a script for cross-platform build system. It contains a list of build instructions includ- ing what executables should be created, what source files to use to build each of them, and where to find the include files and libraries needed for those executables.

A ROS package must be in the parent "catkin_ws/src" folder, its folder, and must contain package.xml and CmakeList.txt.

# 4 The Master

One of the basic goals of ROS is to enable to design software as a collection of small, mostly independent programs called nodes that all run at the same time. For this to work, those nodes must be able to communicate with one another. The part of ROS that facilitates this communication is called the **ROS master**.

Provides naming and registration services to the rest of the nodes in the ROS system. Publishers and Subscribers register to the master, then ROS Master tracks ROS topics being published by the publisher and ROS Topics being subscribed to by the subscribers. It also provides the Parameter Server.

# 5 Nodes

A ROS node is an executable that uses ROS to communicate with other nodes. Each node does not depend on another node. Nodes can publish or subscribe to a Topic.

# 6 Topics

Ros Topics are the buses used by ROS nodes to exchange messages. Imagine a ROS Topic as a water pipe and ROS Message as the water, the two ends of the pipe are where the nodes are located. Topic transport message between a publisher node and a subscriber node.

Nodes that generate message/data publish to a specific topic, and nodes that consume or need data subscribed to a specific topic. The relationship between publishers and subscribers is many to many.

# 7 Messages

Nodes communicate by sending ROS messages to each other using ROS Topic. A message can be of primitive type integer, floating-point, boolean, etc. A publisher and subscriber should communicate using the same topic type. The topic type is determined by the message type.

## 8 Service

ROS service is one to one two way transport, it is suitable for request/reply interactions. A ROS node(server) offers a service, while another ROS node(client) requests for the service. The server sends a response back to the client. Services are defined using srv files. srv files are just like msg files, except they contain two parts: a request and a response. Services also have types like topics.

## 9 Publisher and Subscriber

Publisher and subscriber is many to many but one way transport. A node sends out a message by publishing it to a given topic. The topic is a name that is used to identify the content of the message. A node that is interested in a certain kind of data will subscribe to the appropriate topic.

## 10 Tools

- `rqt_console`
- `rqt_graph`
- `rqt_plot`
- `rqt`
- `rviz`