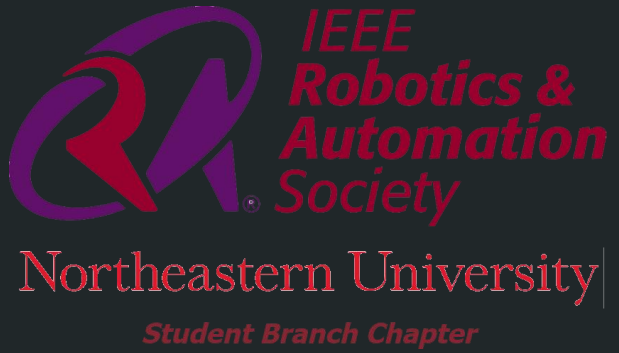
ROS Workshop

IEEE Robotics & Automation Society
Northeastern University
Anas Abou Allaban (abouallaban.a@husky.neu.edu)



Overview

- ROS Framework
- Nodes & Topics
- Services
- Actions
- Publishers
- Subscribers
- Interactive Portion:
  - Build a publisher/subscriber
  - Turtlebot Teleop

Robots

- Definition
- Uses
- Components

What is a Robot?

Definition: A machine (especially one programmable by a computer) capable of carrying out a complex series of actions automatically.

What is a robot (usually) made out of?

- Actuators
  - Parts that directly interact with the environment
- Sensors
  - Parts that measure the environment
- Power System
- Processor

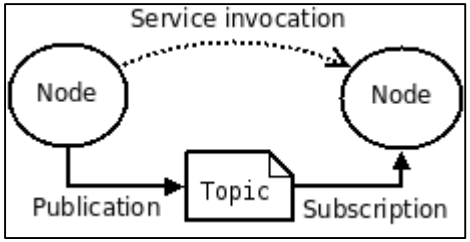
Where are robots commonly used?

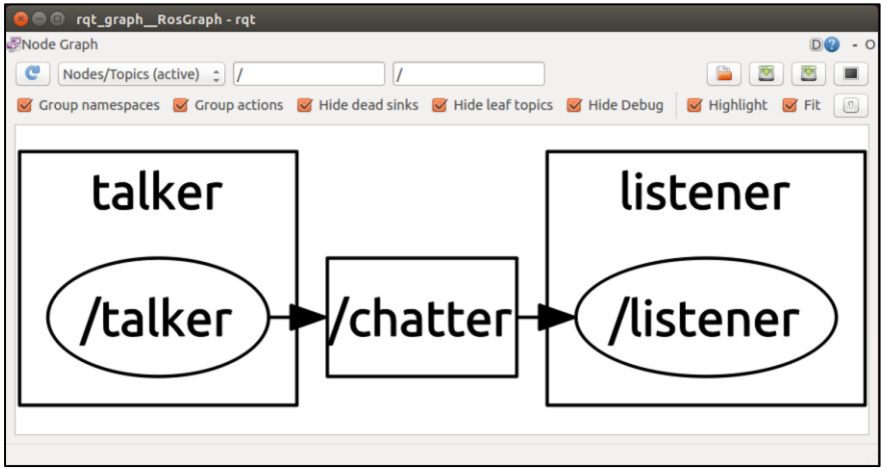
- For tasks that are
  - Dull
  - Dirty
  - Dangerous

- Framework
  - Topics
  - Services
  - Actions
- Publisher/Subscriber Python Node

ROS Framework

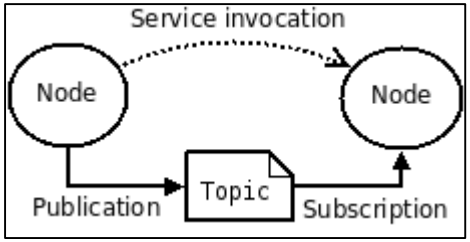
- “Middleware”: Interface between robots and computers/other robots.
- Communication framework using TCP
- Master (ROS Master) & Slave (Nodes)
- RQT Graph





Nodes & Topics

- Nodes:
  - Similar to graph vertex
  - Nodes are what run your program
- Topics:
  - Similar to graph edge
  - Channel in which data is transmitted
  - Nodes subscribe to a topic to get data
  - Node publish to a topic to send data
- Messages:
  - Data that is transmitted on a topic
  - Booleans, numbers, strings, custom structures



std_msgs/String Message

File: std_msgs/String.msg

Raw Message Definition

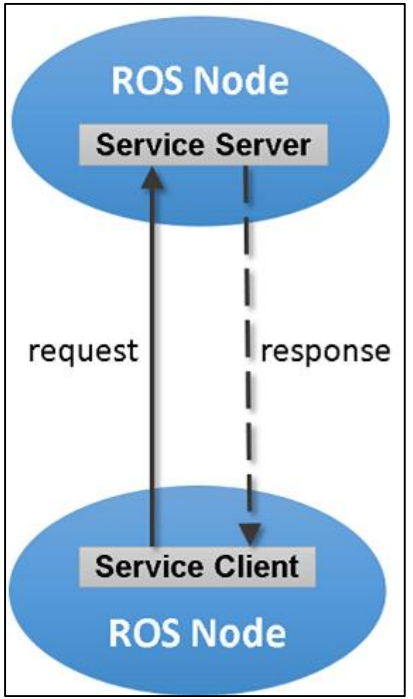
```
string data
```

Compact Message Definition

```
string data
```

Services

- Type of node the computes a ‘remote procedure call’ (RPC).
  - Quick computation or query.
- Used to check status or get updates from the robot.
- Example: Service node that gets number of obstacles from a sensor.
- Service client sends a service request to a server.
- Service servers receive request, process, return a response.

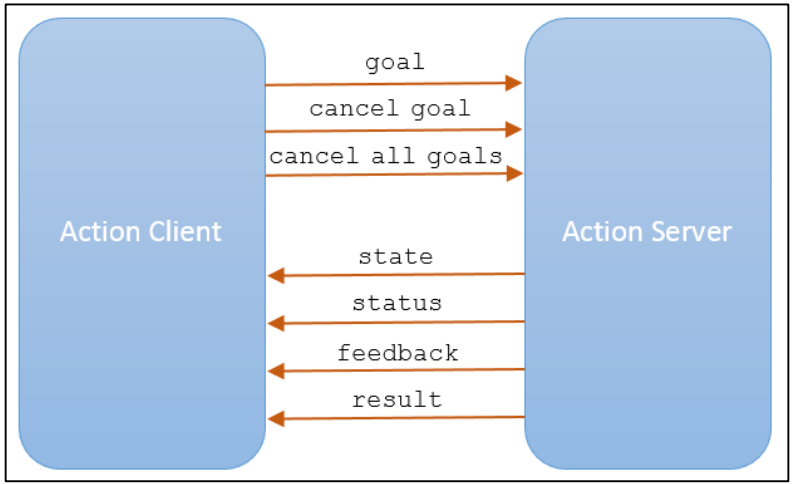


Actions

- More complex non-blocking background processing nodes.
- Used for longer tasks like execution of robot actions.
- Can provide feedback during execution.
- Similar structure to service:

  Server and Client.

  - Client requests an action to be done. Can request status or stop pre-emptively.
  - Server executes responds with status of action.



ROS Tools

- rosrun: Run an executable script or compiled file
  - `rosrun <package_name> <file_name>`
- rostopic: View topic information
  - `rostopic echo <topic_name>`: Print out topic data
  - `rostopic info <topic_name>:` Print out topic info
- roscd: like cd but for ROS packages.
- roslaunch: like rosrun but for launch files.

Robot Ignite Academy

- Navigating interface and creating script

Turtlebot teleop

- Refresh Terminal: `source ~/.bashrc`
- Build packages (must be in the workspace directory): `catkin build`
- Run as executable: `chmod +x <file_name>`

Helpful Links

- Book: Robot Operating System (ROS) for Absolute Beginners
- Tutorials: https://wiki.ros.org/ROS/Tutorials
- Command line cheat sheet: https://www.cheatography.com/davechild/cheat-sheets/linux-command-line/
    - Terminal Aliases: https://www.techradar.com/how-to/computing/apple/terminal-101-creating-aliases-for-commands-1305638
- AWS Robomaker: Watch out for fees!
- Not for the faint of heart (myself included): https://index.ros.org/doc/ros2/