1.4

ROS Services – “request – response” communication

Mukunda Bharatheesha

ROS Services – the basic idea

- ROS nodes communicating with Topics
  - “Many to many one-way communication” – ROS Wiki.
  - No acknowledgement.
  - Useful for monitoring.
- ROS Services
  - **Request – Response** communication.
  - Two message types – a **request**, and a **response**.
- “**Event-based**” execution for ROS applications.

http://wiki.ros.org/Services

ROS Services – on the filesystem, utility cmds

- ROS services are also defined in the ROS messages package
  - hrwros_msgs/**srv** folder.

list all available ROS services

```
$ rosservice list
```

call a ROS service

```
$ rosservice call <service_name> <arguments_req>
```

ROS Services – an example service type

- Requirement: convert measurement in SI units to Imperial Units
  - request message – measurement in SI units (m).
  - response message – measurement in Imperial Units (ft).

hrwros_msgs/srv/ConvertMetresToFeet.srv

```
float64 measurement_metres
---
float64 measurement_feet
Bool success
```

request field

response field

- A service request is processed in a **service callback**.

ROS Services – code nomenclature

- A ROS Node: **service server** “advertises” a service
  - makes the service available for other ROS nodes.

- A ROS Node: **service client** “calls” a service
  - sends a request message once a service is available.

ROS services – a few properties

Attention!

- ROS services block the program flow.
- Useful for designing sequential behaviors.
- Desirable to have quickly executing computations in service callback.
- No going back after a service call!