

Using ROS

Jeremiah Via

23 February 2011

Outline

1 What is ROS?

- Basics
- Vocabulary
- Architecture

2 Why ROS?

3 Code Example

4 Getting Involved

Outline

1 What is ROS?

■ Basics

■ Vocabulary

■ Architecture

2 Why ROS?

3 Code Example

4 Getting Involved

- Robot Operating System
- Meta-operating System
- Originally developed in 2007 at Stanford AI Lab
- Now developed at Willow Garage

- Robot Operating System
- Meta-operating System
- Originally developed in 2007 at Stanford AI Lab
- Now developed at Willow Garage

- Robot Operating System
- Meta-operating System
- Originally developed in 2007 at Stanford AI Lab
- Now developed at Willow Garage

- Robot Operating System
- Meta-operating System
- Originally developed in 2007 at Stanford AI Lab
- Now developed at Willow Garage

Outline

1 What is ROS?

- Basics

- **Vocabulary**

- Architecture

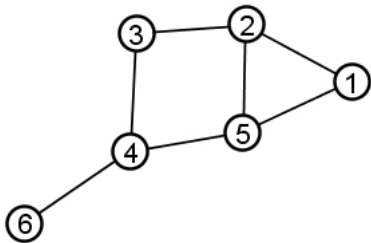
2 Why ROS?

3 Code Example

4 Getting Involved

Node

- A process that performs computation



Message

- The way nodes communicate to one another
- Strictly typed

geometry_msgs/Twist Message

File: geometry_msgs/Twist.msg

```
# This expresses velocity in free space broken into it's linear and angular parts.  
Vector3 linear  
Vector3 angular
```

Expanded Definition

```
Vector3 linear  
float64 x  
float64 y  
float64 z  
Vector3 angular  
float64 x  
float64 y  
float64 z
```

Topic

- String which labels a stream of data, *e.g.*, `cmd_vel` or `scan`
- Nodes can subscribe and publish to these topics



Service

- Synchronous transaction (like a webpage request)

```
#request contents
int8 FOO=1
int8 BAR=2
#request fields
int8 foobar
another_pkg/AnotherMessage msg
---
#response contents
uint32 SECRET=123456
#response fields
another_pkg/YetAnotherMessage val
uint32 an_integer
```

Outline

1 What is ROS?

- Basics
- Vocabulary
- **Architecture**

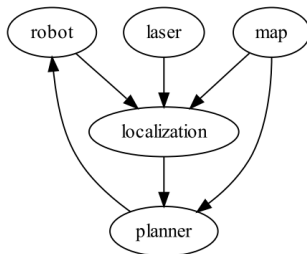
2 Why ROS?

3 Code Example

4 Getting Involved

Peer-to-Peer

- Designed to have many nodes running
- Communication is managed by the master node



<http://www.ros.org/wiki/ROS/Tutorials/UnderstandingNodes>

- └ What is ROS?
- └ Architecture

Language Agnostic



Tools-based

- Follows the UNIX philosophy of many small tools
- Less efficient but more stable and less complex
- Example tools
 - `roscd`
 - `rosls`
 - `roinstall`
 - `rostopic`
 - `roscrcat-pkg`

Thin

- Algorithm and driver development occur as standalone libraries
- Small executables expose library functionality



Outline

1 What is ROS?

- Basics
- Vocabulary
- Architecture

2 Why ROS?

3 Code Example

4 Getting Involved

Debugging

- Debugging is hard
- Debugging robots is really hard
- Modular design
 - Only need to restart the modified nodes
 - Graph structure modified silently
 - No need to take down the entire system

Debugging

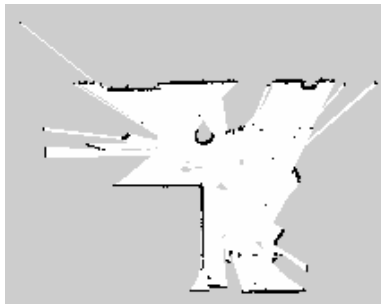
- Debugging is hard
- Debugging robots is really hard
- Modular design
 - Only need to restart the modified nodes
 - Graph structure modified silently
 - No need to take down the entire system

Debugging

- Debugging is hard
- Debugging robots is really hard
- Modular design
 - Only need to restart the modified nodes
 - Graph structure modified silently
 - No need to take down the entire system

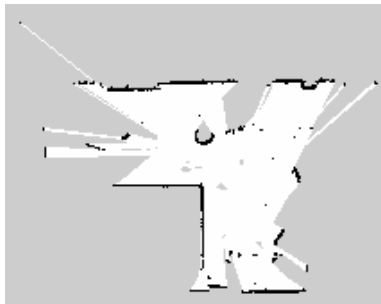
Logging & Playback

- ROS can log any data we want it to
- Logged data can be used to test various algorithms
- Data can even be used to create a map



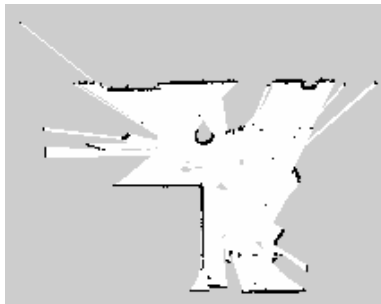
Logging & Playback

- ROS can log any data we want it to
- Logged data can be used to test various algorithms
- Data can even be used to create a map



Logging & Playback

- ROS can log any data we want it to
- Logged data can be used to test various algorithms
- Data can even be used to create a map

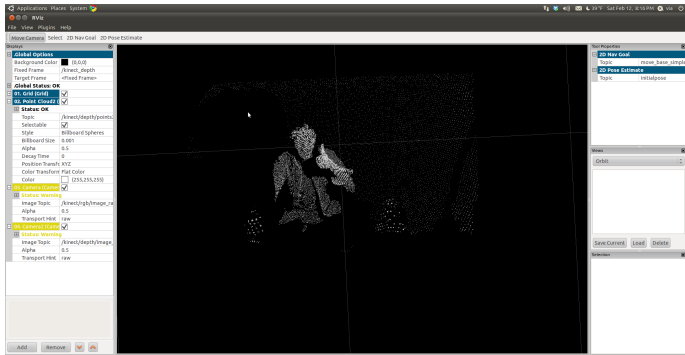


Packaged Subsystems

- Some areas of research are mature enough to use standard algorithms
- No sense in reimplementing a new SLAM system for each new robot
- ROS allows multiple packages to be run together:
`roslaunch`

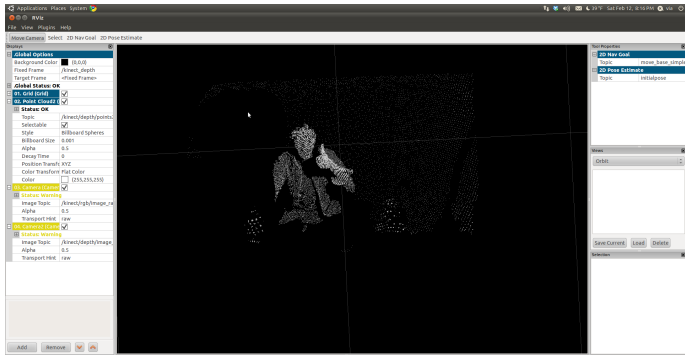
Visualizations & Monitoring

- Printed data can be hard to understand
- Printed data is impossible to understand when it's in 3D
- ROS has a built-in tool to help visualize sensor data called rviz



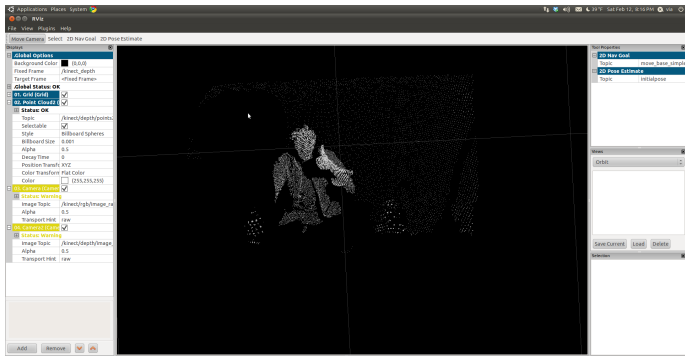
Visualizations & Monitoring

- Printed data can be hard to understand
- Printed data is impossible to understand when it's in 3D
- ROS has a built-in tool to help visualize sensor data called rviz



Visualizations & Monitoring

- Printed data can be hard to understand
- Printed data is impossible to understand when it's in 3D
- ROS has a built-in tool to help visualize sensor data called rviz



Outline

1 What is ROS?

- Basics
- Vocabulary
- Architecture

2 Why ROS?

3 Code Example

4 Getting Involved

Simple Python Example

Robot Programming Exercise 3 part 1:

Build a control strategy that keeps the robot a set distance away from an obstacle in front of it.

look at LaserListenter, Talker, etc.

Outline

1 What is ROS?

- Basics
- Vocabulary
- Architecture

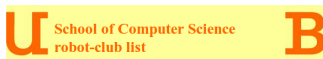
2 Why ROS?

3 Code Example

4 Getting Involved

Join the Mailing List

- The simplest way to get started is to join the mailing list
- `https://mailman.cs.bham.ac.uk/mailman/listinfo/robot-club`



robot-club --

robot-club List Archive

To see the collection of prior postings to the list, visit the [robot-club Archives](#). (The current archive is only available to the list members.)

About robot-club

To post a message to all the list members, send email to robot-club@cs.bham.ac.uk.

You can subscribe to the list, or change your existing subscription, in the sections below.

Subscribing to robot-club

Subscribe to the robot-club mailing list by filling out the following form.

You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a private list, which means that the list of members is not available to non-members.

Your email address:
Your name (optional):

You may enter a privacy password below. This provides only mild security, but should prevent others from changing your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext. **You should not** use your Windows or Unix password.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options. Once a month, your password will be emailed to you as a reminder.

Click a password:

Install ROS

- Go to

<http://www.ros.org/wiki/ROS/Installation> for
installation instructions



Check out the Code

- svn checkout
`https://codex.cs.bham.ac.uk/svn/nah/robotclub`

Summary

- ROS is powerful and useful
- Robot club is fun
- Join!