Introduction to ROS

Lorenz Mösenlechner

Technische Universität München

July 18th, 2012
Motivation

- Today’s robotic systems are complex.
- Many sensors.
- Highly distributed, many processes, many computers.
- Teams of engineers.
Motivation

• Today’s robotic systems are complex.
• Many sensors.
• Highly distributed, many processes, many computers.
• Teams of engineers.

⇒ ROS — The Robot Operating System.
Outline

Overview

ROS Communication Layer

ROS Build System

Programming with ROS

The TF Library
Outline

Overview

ROS Communication Layer

ROS Build System

Programming with ROS

The TF Library
What is ROS?
More than just a middleware

- A “meta” operating system for robots
- A collection of packaging, software building tools
- An architecture for distributed inter-process/inter-machine communication and configuration
- Development tools for system runtime and data analysis
- A language-independent architecture (c++, python, lisp, java, and more)
What is ROS not?
No confusion

- An *actual* operating system
- A programming language
- A programming environment / IDE
- A hard real-time architecture
What does ROS get you?
All levels of development
What does ROS get you?
All levels of development
What does ROS get you?
All levels of development

general tools for distributed computing

ROS

universe

main
What does ROS get you?
All levels of development
What does ROS get you?
All levels of development

- universe
- main
- algorithms
  - frameworks
  - hardware drivers
  - "robotic apps"
- general tools for distributed computing

Maintained by Willow Garage, inc and some external developers

Introduction to ROS
What does ROS get you?
All levels of development

Developed and maintained by the international ROS community

algorithms frameworks
hardware drivers
"robotic apps"

Maintained by Willow Garage, inc and some external developers

general tools for distributed computing

Overview | ROS Communication Layer | ROS Build System | Programming with ROS | The TF Library
---|---|---|---|---
Lorenz Mösenlechner
What does ROS get you?
All levels of development

packaging & build tools, communication infrastructure, ROS API language bindings, introspection tools...
What does ROS get you?
All levels of development

- applications
- capabilities
- libraries
- main

packaging & build tools, communication infrastructure, ROS API language bindings, introspection tools...

Lorenz Mösenlechner
What does ROS get you?
All levels of development

- applications
- capabilities
- libraries
- main

packaging & build tools, communication infrastructure, ROS API language bindings, introspection tools...

tf, opencv, pcl, kdl, cisst, simulation, drivers...

Overview      ROS Communication Layer      ROS Build System      Programming with ROS      The TF Library

Lorenz Mösenlechner
What does ROS get you?
All levels of development

- applications
- capabilities
- libraries
- main

grasping, control, execution, navigation...
tf, opencv, pcl, kdl, cisst, simulation, drivers...
packaging & build tools, communication infrastructure, ROS API language bindings, introspection tools...

Overview  ROS Communication Layer  ROS Build System  Programming with ROS  The TF Library

Lorenz Mösenlechner

Introduction to ROS
What does ROS get you?
All levels of development

- applications
- capabilities
- libraries
- main

fetching beer, scraping the seafloor, grasping, control, execution, navigation...
tf, opencv, pcl, kdl, cisst, simulation, drivers...

packaging & build tools, communication infrastructure, ROS API language bindings, introspection tools...

Overview  ROS Communication Layer  ROS Build System  Programming with ROS  The TF Library
Lorenz Mösenlechner
Introduction to ROS
The ROS Community
Researchers using common tools to enable collaboration

79 Institutional ROS Repositories, all over the world (July, 2011)
www.ros.org - The ROS Hub
A centralized location for ROS users and developers

ROS.org

Documentation  Browse Software  News  Download

Documentation
ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more.

ROS:
Install
Getting Started
Tutorials, technical overview, and links to getting help. Also, checkout the ROS cheat sheet PDF.
Contribute
How to contribute to the ROS community, such as submitting your own repository.

Software:
Core Libraries
APIs by-language and topic.
Common Tools
Common tools for developing and debugging ROS software.
Search Software
Search the 2000+ libraries available for ROS.

Robot Hardware:
Robots
Robots that you can use with ROS.
Sensors
Sensors drivers for ROS.
Driver Tutorials
Tutorials for supported hardware.

Publications, Courses, and Events
Papers
Published papers with open source implementations available.
Courses
Courses using or teaching ROS.
Events
Past events and materials based on ROS.

Except where otherwise noted, the ROS wiki is licensed under Creative Commons Attribution 3.0.
answers.ros.org - ROS Questions & Answers
Community-supported help for ROS users
ros mailing lists
Getting in touch with the developer community

- ROS Users - for general ROS-related discussions
  https://code.ros.org/mailman/listinfo/ros-users

- Other Lists & List Archives
  http://code.ros.org/lurker
Outline

Overview

ROS Communication Layer

ROS Build System

Programming with ROS

The TF Library
ROS Core
Where it all comes together

- **ROS Master**
  - A centralized XML-RPC server
  - Negotiates communication connections
  - Registers and looks up names for ROS graph resources

- **Parameter Server**
  Stores persistent configuration parameters and other arbitrary data

- **rosout**
  Essentially a network-based stdout for human-readable messages
ROS “Graph” Abstraction
Named network resources

ROS graph resources:

- **nodes**
  - processes
  - produce and consume data

- **parameters**
  - persistent data storage
  - configuration, initialization settings
  - stored on parameter server

- **topics**
  Asynchronous many-to-many communication streams.

- **services**
  Synchronous one-to-many network-based functions.
ROS “Graph”
rxgraph: communication network visualization
Creating and Running ROS Nodes
Distributing computation with ROS

Launch files

- XML files for launching nodes
- associate a set of parameters and nodes with a single file
- hierarchically compose collections of other launch files
- automatically re-spawn nodes if they crash
- change node names, namespaces, topics, and other resource names *without* recompiling
- easily distribute nodes across multiple machines
Example Launch File

Example *launch file*

```xml
<launch>
  <node name="my_node" pkg="foo" type="bar">
    <remap from="/base_laser/scan" to="scan" />
    <rosparam>
      use_foo: True
      frame_id: base_laser
    </rosparam>
  </node>
</launch>
```

- Launch with roslaunch package foo.launch
ROS Communication Protocols
Connecting nodes over the network

- **ROS Topics**
  - Asynchronous “stream-like” communication
  - Strongly-typed (ROS .msg spec)
  - Can have one or more *publishers*
  - Can have one or more *subscribers*

- **ROS Services**
  - Synchronous “function-call-like” communication
  - Strongly-typed (ROS .srv spec)
  - Can have only one *server*
  - Can have one or more *clients*

- **Actions**
  - Built on top of topics
  - Long running processes
  - Cancellation
Asynchronous Distributed Communication
ROS TCP Topics

- ros "master"
- camera
- viewer
Asynchronous Distributed Communication
ROS TCP Topics

```
advertise("images")
```

```
camera
```

```
ros "master"
```

```
viewer
```
Asynchronous Distributed Communication
ROS TCP Topics

camera

ros
"master"

viewer

topic:images
Asynchronous Distributed Communication
ROS TCP Topics

subscribe("images")

ros
"master"

-camera-
topic:images

viewer

Introduction to ROS
Asynchronous Distributed Communication

ROS TCP Topics

subscribe("images")

camera

ros
"master"
topic:images

viewer
Asynchronous Distributed Communication
ROS TCP Topics
Asynchronous Distributed Communication
ROS TCP Topics

Overview
ROS Communication Layer
ROS Build System
Programming with ROS
The TF Library

Introduction to ROS

Lorenz Mösenlechner
Asynchronous Distributed Communication
ROS TCP Topics

![Diagram of ROS TCP communication between camera and viewer]

- camera
  - ros
    - "master"
  - topic:images
  - publish(img)
- viewer
  - images(tcp)
Asynchronous Distributed Communication
ROS TCP Topics

```
ros
"master"

`topic:images`

`camera` 
`images(tcp)` 
`viewer`

`publish(img)` 

`viewer_too`
```
Asynchronous Distributed Communication

ROS TCP Topics

```
ros
"master"

subscribe("images")

topic:images

camera

images(tcp)

viewer

publish(img)

viewer_too
```
Asynchronous Distributed Communication

ROS TCP Topics

Diagram showing a ROS network with topics and nodes:
- **camera** publishing images
- **viewer** subscribing to images
- **viewer_too** subscribing to images (TCP)
- The network includes topics like "master", topic:images, images(tcp)

Overview
- ROS Communication Layer
- ROS Build System
- Programming with ROS
- The TF Library
ROS Graph Introspection
No more wireshark

ROS provides several tools for analyzing the data flowing over ROS communication resources:

- **rosnod**e
  Gives a user information about a node: publications, subscriptions, etc

- **rostopic**
  Gives datarate, actual data, publishers, subscribes

- **rosservice**
  Enables a user to call a ROS Service from the command line

- **roswtf** (wire trouble finder)
  Diagnoses problems with a ROS network
ROS GUI Tools
There are lots...
ROS GUI Tools
There are lots...
ROS GUI Tools
There are lots...
rviz - 3D Visualization
Modular state and sensor visualization

Overview
ROS Communication Layer
ROS Build System
Programming with ROS
The TF Library

Lorenz Mösenlechner
Introduction to ROS 49