ROS : Robot Operating System

Weipeng He
2he@informatik.uni-hamburg.de

5 November, 2012

Outline

Introduction

What is ROS?

- ROS is Robot Operating System.
- ROS is an open-source, meta-operating system for robots.
- It provides the services including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management.
- It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

Introduction

History

- ROS was originally developed in 2007 under the name switchyard by the Stanford Artificial Intelligence Laboratory in support of the Stanford AI Robot (STAIR) project.
- From 2008, development continues primarily at Willow Garage, with more than twenty institutions collaborating in a federated development model.
- April 23, 2012 the latest stable release : Fuerte.
- Now, ROS works on different operating systems (Linux, Mac OS, Windows, FreeBSD, etc.).
- ROS can be developed in C++, Python and Lisp.
- More than 80 robots(PR2, Care-O-bot 3, iRobot Create, Aldebaran Nao, etc.) are listed as supported by ROS.

Motivation

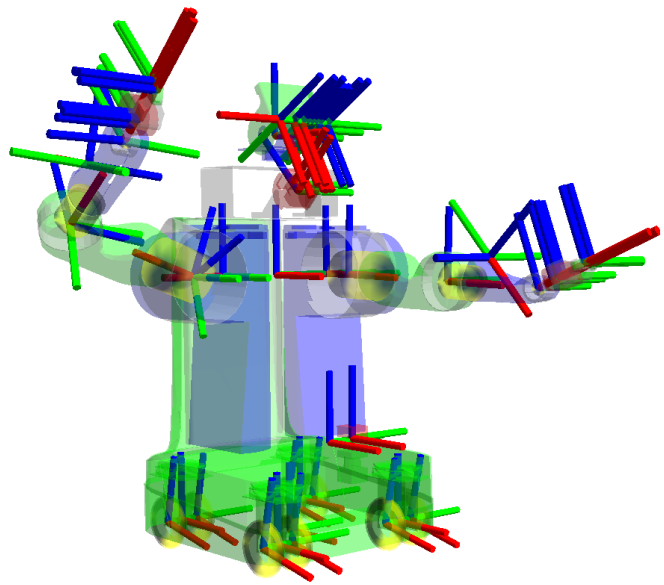
Some facts about research in robotics

- It's difficult to reproduce the experiment results from other's publication.
- Researchers use different robots. The hardware are very different.
- Sometimes, researchers from different groups just do the same coding work.
- What if they have a platform that they can use to collaborate?

Motivation

Pose tracking

- Suppose I need to develop a program to track the pose of an robot arm.



- tf in ROS : keep track of multiple coordinate frames over time.

Motivation

We can use ROS

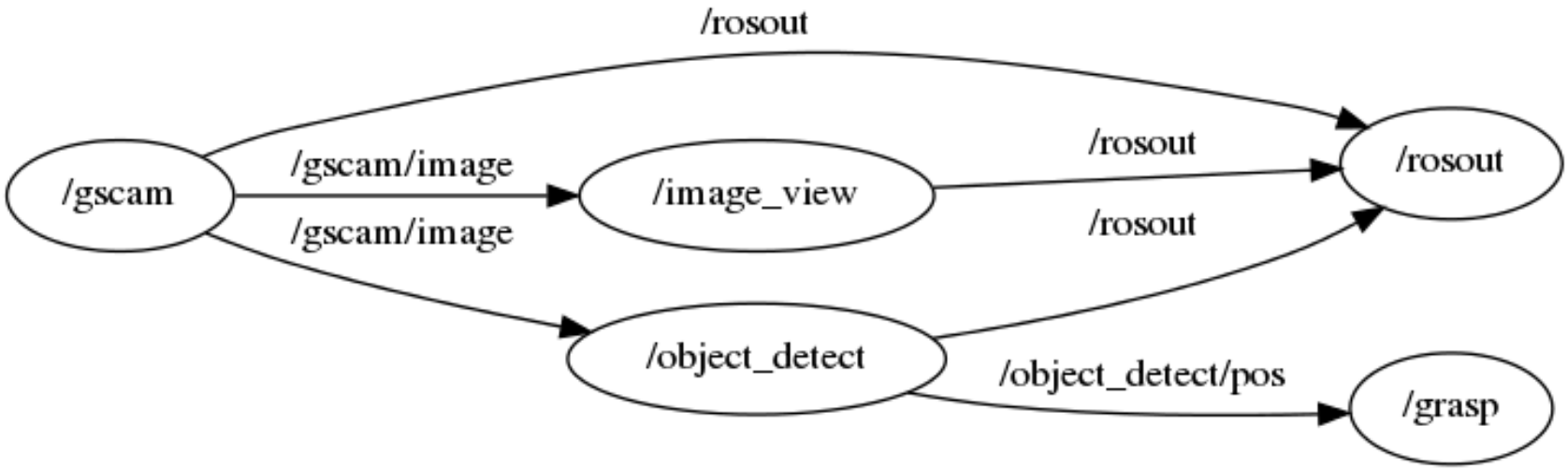
- The primary goal of ROS is to support code reuse in robotics research and development.

- Well-designed software structure.
- Numerous libraries and drivers.
- Utilities which assist development.
- Community support.

Software Structure

ROS Computation Graph

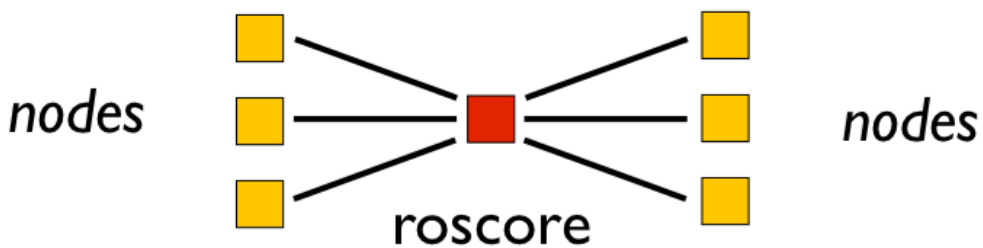
- The ROS runtime "graph" is a peer-to-peer network of processes using the ROS communication infrastructure.



Software Structure

Node and Master

- Nodes are processes that perform computation.
- ROS is designed to be modular at a fine-grained scale. Robot control system will usually comprise many nodes.
- For example, one node controls a laser range-finder, one node controls the wheel motors, one node performs localization, one node performs path planning, one node provides a graphical view of the system, and so on.
- Nodes get to know each other via roscore (master).



Software Structure

Message

- Nodes communicate with each other by passing messages.
- A message is simply a data structure, comprising typed fields.
- Standard primitive types (integer, floating point, boolean, etc.) are supported, as are arrays of primitive types.
- Messages can include arbitrarily nested structures and arrays (much like C structs).

Software Structure

Message example

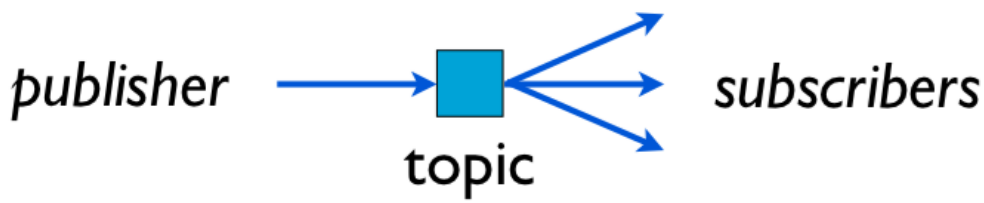
- sensor_msg/Image

```
Header header
  uint32 seq
  time stamp
  string frame_id
uint32 height
uint32 width
string encoding
uint8 is_bigendian
uint32 step
uint8[] data
```

Software Structure

Topic

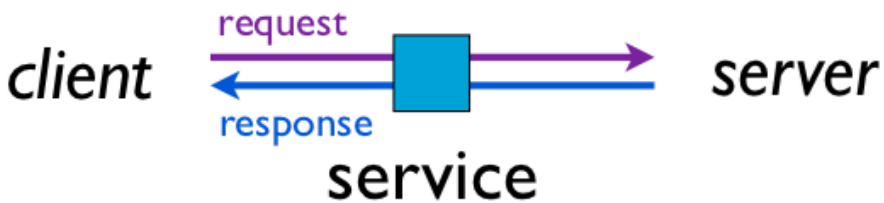
- Topic is a mechanism to send messages from a node to one or more nodes.
- Follows a publisher-subscriber design pattern.
- Publisher is the node which send messages to the topic.
- Subscribers are the nodes which get called whenever a message is published.
- Example : Camera publishing images.



Software Structure

Service

- Service is a mechanism for a node to send a request to another node and receive a response in return.
- Follows a request-response design pattern.
- A service is called with a request message, and in return, a response message is returned.
- Example : Request the camera to tilt.



Software Structure

File system Hierarchy

- Release : collection of stacks and packages.
  - Fuerte
- Stacks : a full application suite.
  - geometry
- Package : software to solve a specific task.
  - tf
- Node : an executable.
  - tf_echo

Community

Software Distribution

- ROS code is maintained in a decentralized federation of repositories.
  - The core repository : ros-pkg;
  - 94 repositories in other institutions;
  - 14 personal repositories.
- Easy to contribute.
  - Host their code (and documents) in their own repository.
  - SourceForge.net, Google Code and GitHub.
  - Register at ros.org.
- Easy to search software.
  - Search across the federation of repositories is possible.
  - ros.org keeps tracks of updates of all repositories and generates index.
  - Documentation and tutorials are also updated automatically.

Community

- ROS Answers.

Conclusion

- ROS defines a standard for the communicate mechanisms and protocols between robot components.
- ROS provides libraries for various functions.
- ROS contains utility tools to help development.
- ROS promotes code sharing and reuse.

Questions?

Thanks!