Faculty of Sciences and Technology of the University of Coimbra
Department of Electrical and Computer Engineering

ROS
ROBOT OPERATING SYSTEM

Micael S. Couceiro
Cooperative Robotics Reading Club

2010
Overview

- Project Description
  - Why ROS?
  - What is ROS?
- Applications
  - Areas
  - Robots using ROS
- Conclusion
- Bibliography
- Questions & Discussion
Project Description

- Why ROS?
  - Open source-software platform designed to support a new generation of robots:
    - Simplifying the integration of different software libraries from different institutions;
    - Supporting many processors;
    - Offering rich set of tools to successfully manage the complexity;
    - Supporting modern software engineering techniques such as continuous testing and integration;
    - Supporting efficient communication;
    - Supporting real-time components.
What is ROS?

- ROS is a robot operating system originally developed in 2007 under the name switchyard by the Stanford Artificial Intelligence Laboratory in support of the Stanford AI Robot STAIR project;

- As of 2008, development continues primarily at Willow Garage, a robotics research institute/incubator, with more than twenty institutions collaborating in a federated development model;

- The philosophical goals of ROS can be summarized as:
  - Peer-to-peer;
  - Tools-based;
  - Multi-lingual;
  - Thin;
  - Free and Open-Source.
Project Description

- **Peer-to-peer**
  - Lookup mechanism to allow processes to find each other at runtime avoiding unnecessary traffic flowing across the wireless link.

- **Tools-Based**
  - A large number of small tools are used to build and run the various ROS components, rather than constructing a monolithic development and runtime environment;
  - These tools perform various tasks:
    - navigate the source code tree;
    - get and set configuration parameters;
    - visualize the peer-to-peer connection topology;
    - measure bandwidth utilization;
    - graphically plot message data;
    - auto-generate documentation.
**Project Description**

- **Multi-lingual**
  - Messages are passed peer to peer and are not based on a specific programming language; nodes can be written in C++, Python, C, LISP, Octave, or any other language for which someone has written a ROS wrapper.

- **Thin**
  - Placing virtually all complexity in libraries, and only creating small executables which expose library functionality to ROS, allows for easier code extraction and reuse beyond its original intent.
  - ROS re-uses code from numerous other open-source projects such as:
    - The drivers, navigation system, and simulators from *Player/Stage*;
    - Vision algorithms from *OpenCV*;
    - Planning algorithms from *OpenRAVE*;
    - Many others.
Project Description

- Free and Open-Source
  - The full source code of ROS is publicly available
  - Allows the development of both non-commercial and commercial projects
  - Easy to use:
    - Getting started page at [http://ros.org](http://ros.org) that explains installation instructions and guides the user through the initial tutorials;
    - Around 175 tutorials documenting the released stacks;
    - More than 15 repositories of useful packages from around the world;
    - ROS community includes almost 40 software developers at Willow Garage, numerous members of the academic robotics community at dozens of institutions, and researchers at other companies, notably, Intel and Bosch;
    - Several events are being created in order to accelerate the code sharing in robotics (e.g., Call for Proposals: PR2 Beta Program at [http://www.willowgarage.com/pages/pr2-beta-program/cfp](http://www.willowgarage.com/pages/pr2-beta-program/cfp))
Applications

- ROS areas include:
  - A master coordination node;
  - Publishing or Subscribing to data streams;
  - Multiplexing information;
  - Node creation and destruction;
  - Nodes are seamlessly distributed;
  - Logging;
  - Parameter server;
  - Test systems;

- ROS Package application areas will include:
  - Perception
  - Object Identification
  - Segmentation and Recognition
  - Face Recognition
  - Gesture Recognition
  - Motion Tracking
  - Ego-motion
  - Motion Understanding
  - Structure from motion (SFM)
  - Stereopsis Stereo vision
  - Mobile Robotics
  - Control
  - Planning
  - Grasping
Applications

- Robots using ROS:
  - Care-O-Bot 3
    - Mobile manipulator from Fraunhofer IPA in Germany;
    - Designed to have a human facing the side with a tray and a Schunk lightweight arm that can pick items up and set them on the tray;
    - The robot’s software is open source and available via [http://ros.org](http://ros.org) and includes some simple applications (a dashboard and teleoperation application), packages for controlling the motors and arms, and a simulator.
    - Many of these packages make use of the underlying ROS capabilities, including the messaging system and transform library.
Robots using ROS:

iRobot Create

- Platform for experimentation developed in Brown Robotic Group based on Romba – An autonomous robotic vacuum cleaner;
- ROS gives developers access to all kind of capabilities, from flexible-distributed computing to powerful data visualization.
Applications

- Robots using ROS:
  - Aldebaran Nao
    - The Brown Robotics Group also released drivers for the Aldebaran Nao platform, including basic movement, head control, speech, and camera access;
    - Researchers at the Albert-Ludwigs-Universitat in Freiburg, Germany, added joystick teleoperation, joint state inspection, and a basic robot model.
Applications

- Robots using ROS:
  - PR2
    - Willow Garage’s PR2 runs software for, among other basic functions, calibration, navigation, and manipulation, as well as for higher level applications including mapping and plugging into standard wall outlets;
    - All of the code developed will run on the distributed robots, and in exchange for use of the PR2 robot, the recipients will make their code available open source.
Applications

Robots using ROS:

- Skybotix's CoaX Helicopter
  - The CoaX helicopter from Skybotix is a micro UAV targeted at the research and educational markets that includes an IMU, a downward-looking and three optional sideward-looking sonars, pressure sensor, color camera, and Bluetooth, XBee, or WiFi communication;
  - It is a complete system with basic ROS setup so customers can use ROS right out of the box.
Conclusion

- ROS provides standard operating system services such as:
  - Hardware abstraction;
  - Low-level device control;
  - Implementation of commonly-used functionality;
  - Package management.

- The library is geared towards a Unix-like system:
  - *Ubuntu Linux* is listed as “supported”;
  - Other variants such as *Fedora* and *Mac OS X* are still considered “experimental”;
  - At present *Windows* is listed as having “partial functionality”.

- ROS has two basic “sides”:
  - The operating system side (as described above);
  - ROS-PKG, a suite of user contributed packages organized into sets called stacks that implement functionality such as simultaneous localization and mapping, planning, perception, simulation, etc.

- Free for commercial and research use.
Bibliography

- ROS Website. http://ros.org/
Questions & Discussion