

ROS: introduction, concepts and examples

Cédric Pradalier

October 16th, 2013



Objectives of this talk

ROS and middleware

- ▶ Quick presentation
- ▶ Quick comparison

Use case

- ▶ For teaching
- ▶ For research

Outline

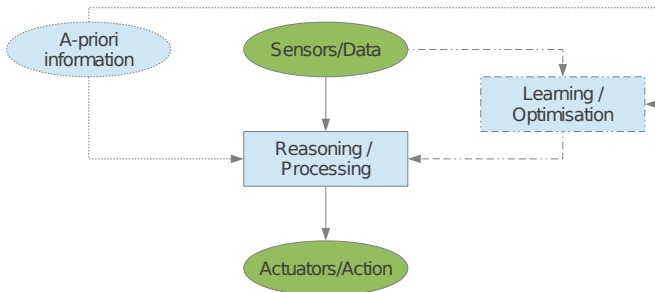
ROS: yet another middleware

Use-cases

Conclusion

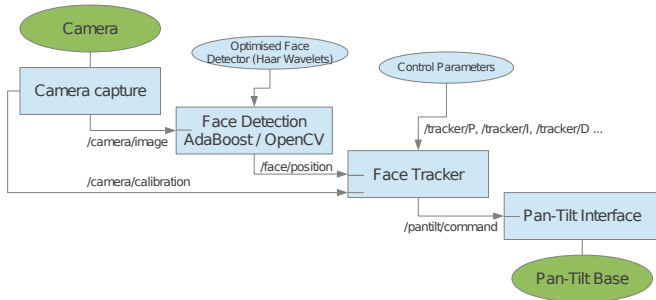
Why a middleware?

- ▶ Robotics development for the soft real-time people...



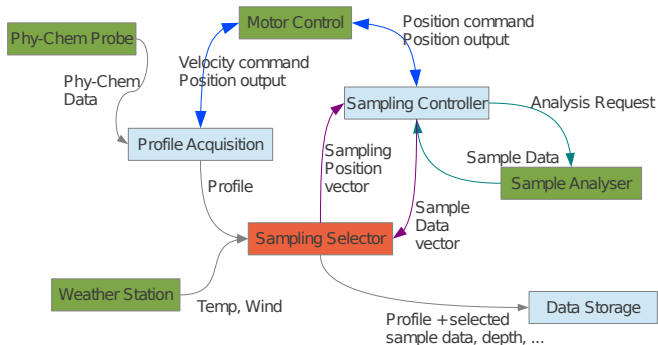
Why a middleware?

- ▶ Robotics development for the soft real-time people...



Why a middleware?

- ▶ Robotics development for the soft real-time people...



Examples of middleware for robotics

Limited to the one I used personally:

- ▶ Genom – LAAS
- ▶ DDX – CSIRO, maintainer
- ▶ Carmen
- ▶ IPC
- ▶ Yarp
- ▶ Moos
- ▶ Corba
- ▶ RTI – commercial
- ▶ ADTF – commercial
- ▶ ROS – core contributions

ROS

- ▶ Stands for Robot Operating System, but runs on top of existing OS like (mostly) Linux
- ▶ Originates from SAIL/Willow Garage, now moved to OSRF
- ▶ Combines several features into a consistent project:
 - ▶ A software distribution with a dependency mechanism
 - ▶ An integrated build system
 - ▶ A communication middleware
 - ▶ Helper tools: visualization, record/replay, etc.
- ▶ Provides client libraries for C++, Python, lisp; experimental ones for Java, Lua

ROS, the distribution platform

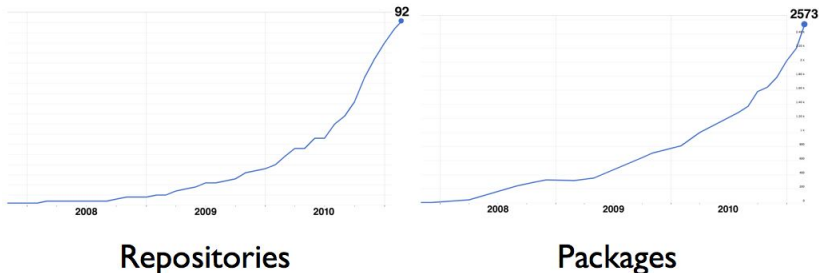
Software packages, organized in stacks/packages, that cover many areas of robotics:

- ▶ perception
- ▶ state estimation
- ▶ mapping
- ▶ navigation
- ▶ planning
- ▶ object detection
- ▶ grasping
- ▶ etc.

ROS, the robotics middleware

- ▶ Components (nodes) that may run on different computers
- ▶ Data exchanged through topics (event-based) and services (RPC), wrapper generated from descriptions
- ▶ TCP and UDP transport layers
- ▶ Standardized messages (images, laser scans, point clouds, IMU, GPS, joystick)
- ▶ Parameter server
- ▶ Visualization of data (rviz)
- ▶ Record and replay (bags)

ROS, status and future – from 2012



- ▶ Work force: currently 60 full-time positions at Willow Garage + many contributors from academia.
- ▶ Being a meeting point, ROS has led to dramatic improvements in the standardization of robotics software.
- ▶ Challenges: maintenance of core, integration of third-party contributions, further standardization, application to industry.

ROS, why do I use it?

Software

- ▶ Excellent documentation and tutorials
- ▶ Reliable software tools for data management
- ▶ Standardization of most message type (image, geometry, ...)
- ▶ Reliable communication system.
- ▶ Availability of most functionality I need to build an application.
- ▶ Link with OpenCV and PCL.

Education

- ▶ Excellent documentation and tutorials
- ▶ Lot of resources to let students focus on the important stuff.

ROS, what I don't like

Software

- ▶ Heavy: design for big machines, lot of RAM, lot of hard-drive.
- ▶ Not super high-performance communication system.
- ▶ Low-performance of communication system in python.
- ▶ New build system (catkin)

Management

- ▶ Very hard to do core contributions
- ▶ Uncertainty on future management

Outline

ROS: yet another middleware

Use-cases

Conclusion

Autonomous Shore Inspection



A ROS system

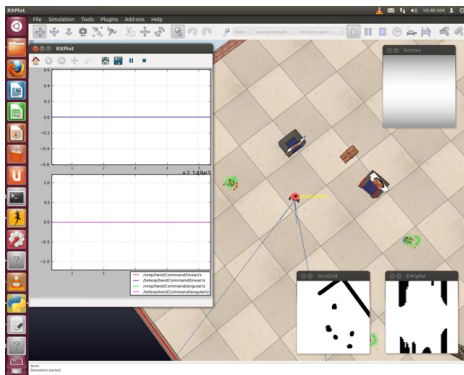
System

- ▶ Delivered as a ROS system by Clearpath Robotics
- ▶ Mix of Python and C++ sensor/actuator interfaces: open-source, changed/improved most of them since delivery.
- ▶ Standard message for commands and most data.

ROS advantage

- ▶ Simplification of development
- ▶ Simplification of data monitoring and logging from ground station.

Robotic classes with V-REP



A ROS system

System

- ▶ Integration of ROS as a plugin in V-REP.
- ▶ Simple use by students, simple scene creation.
- ▶ Standard message for commands and most data.
- ▶ Simplification of data monitoring .

Why not Gazebo?

- ▶ Not reliable enough when starting designing the classes.

Outline

ROS: yet another middleware

Use-cases

Conclusion

Conclusion

ROS

- ▶ Core set of tools in all my developments.
- ▶ Fits most needs by combining communications, build-systems and reach software ecosystem.
- ▶ Enabler for most research needs.
- ▶ Enabler for robotic education.

The final middleware?

- ▶ Space for improvement.
- ▶ (Hopefully) open enough to let the community take charge even without Willow Garage.
- ▶ Better support for other OS?

Thank you for your attention

