Running a robotics business through open source software

Luca Marchionni
CTO - PAL Robotics
Presentation outline

➢ Company presentation

➢ Open-source in PAL Robotics

➢ Continuous Integration

➢ Package Release System

➢ Distro stabilization

➢ Customer support

➢ What's inside our robots

➢ Public repositories
PAL Robotics Team
PAL Robotics in a Nutshell


REEM-A  REEM-B  REEM-H  REEM  REEM-C  TIAGO  STOCKBOT  TALOS
PAL Robotics Expertise

Mechanics
- Bio-inspired designs
- Actuation designs
- Integration

Electronics
- Motor Control Boards
- Sensor Control
- Connectivity

HMI
- Speech synthesis and recognition
- Touchscreen
- Face recognition

Software
- SLAM and Navigation
- Visual perception

Control
- Real Time
- Manipulation
- Walking

Robotics
Collaboration - Projects
Collaboration: European H2020 Project

- **Factory-in-a-Day** FP7-2013-NMP-ICT-FoF
- **SosSMCs** FETPROACT-2-2014
- **GrowMeUp** PHC-19-2014
- **SACRO** Eurostar
- **Socrates** MSCA-ITN-2017
- **Co4Robots** ICT-25-2016-2017
- **RobMoSys** ICT-25-2016-2017
Open Source in PAL Robotics

ubuntu

ROS

The Orocos Project
Smarter control in robotics & automation!

Gazebo

MoveIt!

OpenCV

pcl

Jenkins

Redmine

GitLab

GitHub

PAL Robotics
Why do we use open source?

As software developers:

➢ Do not re-invent the wheel

➢ Learn from others' code

➢ Enforce use of best practices

➢ Share the effort

➢ Be recognized by community
Why do we use open source?

As providers of robotics research platforms:

- ROS is a selling argument
- Documentation and tutorials
- Easy to integrate new software or hardware
- Gazebo + ROS are taught in universities
- Evaluation through public available simulation
Why do we use open source?

As manufacturer of products:

➢ Shorten time to market
➢ If you have the code, you can improve it
➢ Leverage powerful development tools
➢ Customizability
➢ Quality, security
➢ Build value, focus on core IP
How we use open source

Choose long term versions

- Ubuntu 14.04 LTS
- ROS Indigo
- PAL Dubnium
- Continuous integration
- Release stable package versions
Continuous Integration

- Triggered by pushes to our Gitlab
- Building in a clean environment
- Testing
- Coverage
- API/ABI check
Continuous Integration

- Jenkins UI showing various jobs and their statuses, including Last Success, Last Failure, and Last Duration.
- The jobs are categorized under different tabs such as Debian/CentOS, Firmware, Fuerte jobs, Gitlab, Jobs, Integration Jobs, Measurement, Production Jobs, Release, and others.
- The UI displays build status icons for each job, indicating if they are running, failed, or successful.
- The jobs are tagged with specific labels and statuses, indicating their current status.
- The layout includes a sidebar for quick access to different sections.
Continuous Integration

Email notification to developers and project managers in case of:

- Compilation error
- Test failure
- Coverage below 70%
- API/ABI breaking
Software Package Release System
Software Package Release System

**pmb2_navigation** updated-releasable

{dubnium-devel} version depends on:
['depthimage_to_laserscan', 'pal_local_planner', 'pal_pcl', 'sick_tim', 'pal_navigation_sm', 'pal_laser_filters', 'pal_vo_server', 'slam_gmapping', 'rviz_plugin_covariance', 'pal_karto', 'navigation']

Released at version: 0.10.0-staging.0

Released against:
{'depthimage_to_laserscan': '1.0.8-staging.0', 'pal_local_planner': '2.0.1-staging.0', 'pal_pcl': '0.1.12-staging.0', 'sick_tim': '0.0.9-staging.0', 'pal_navigation_sm': '0.1.8-staging.0', 'pal_laser_filters': '0.0.2-staging.0', 'pal_vo_server': '0.0.16-staging.0', 'slam_gmapping': '1.3.6-staging.0', 'rviz_plugin_covariance': '0.0.5-staging.0', 'pal_karto': '0.7.1-staging.0', 'navigation': '1.11.28-staging.0'}

Depended on by
pal_metapkg_development_pmb2
pmb2_simulation
pal_metapkg_pmb2
pmb2_navigation_specifics
pmb2_navigation updated-releasable

{dubniu-devel} version depends on:
['depthimage_to_laserscan', 'pal_local_planner', 'pal_pcl', 'sick_tim', 'pal_navigation_sm', 'pal_laser_filters', 'pal_vo_server', 'slam_gmapping', 'rviz_plugin_covariance', 'pal_karto', 'navigation']
Released at version: 0.10.0-staging.0

Released against:
{'depthimage_to_laserscan': '1.0.8-staging.0', 'pal_local_planner': '2.0.1-staging.0', 'pal_pcl': '0.1.12-staging.0', 'sick_tim': '0.0.9-staging.0', 'pal_navigation_sm': '0.1.8-staging.0', 'pal_laser_filters': '0.0.2-staging.0', 'pal_vo_server': '0.0.16-staging.0', 'slam_gmapping': '1.3.6-staging.0', 'rviz_plugin_covariance': '0.0.5-staging.0', 'pal_karto': '0.7.1-staging.0', 'navigation': '1.11.28-staging.0'}

Depended on by
pal_metapkg_development_pmb2
pmb2_simulation
pal_metapkg_pmb2
pmb2_navigation_specifics

➢ catkin_lint
➢ catkin_prepare_release
➢ catkin_generate_changelog

Software Package Release System
Distro stabilization

- Staging repository
  - for developers
  - for non-stable installation

- Candidate repository
  - Initialized with all debians required for a full installation of a robot platform
  - Verification with UAT

- Stable repository
  - API/ABI changes are not allowed
  - Bug fixes only
Customer Support

Updated Repositories from PAL Robotics
Graphical installation Tools
Remote support

http://support.pal-robotics.com
What's inside our robots (all of them)

- URDF (accurate dynamic model)
- ros_control
- ros_controllers: joint_state_controller, diff_drive_controller, joint_trajectory_controller, ...
- slam: slam_karto, slam_gmapping, cartographer
- ros navigation stack: amcl, move_base, costmap_2d, ...
- MoveIt
- Rviz, Rqt

...and more
Public repositories

http://wiki.ros.org/Robots/REEM
http://wiki.ros.org/Robots/REEM/Tutorials
http://wiki.ros.org/Robots/REEM-C
http://wiki.ros.org/Robots/REEM-C/Tutorials
http://wiki.ros.org/Robots/TIAGo
http://wiki.ros.org/Robots/TIAGo/Tutorials
Thank you for your attention!