Towards Live Programming in ROS with PhaROS and LRP

Pablo Estefó¹, Miguel Campusano², Luc Fabresse¹, Johan Fabry², Jannik Laval¹, and Noury Bouraqadi¹

http://pleiad.cl
http://car.mines-douai.fr
Robotic Application Lifecycle

1. Develop / Borrow Parts
2. Assemble Parts
3. Run + Analyze Behavior
4. Stop
5. Fix bugs / Improve
6. Goto step 2
ROS Robotic Application Lifecycle

1. Develop / Borrow ROS Packages
2. Assemble = Define Launch file
3. Run + Analyze Behavior
4. Stop
5. Fix bugs / Improve
6. Goto step 2
**ROS Robotic Application Lifecycle**

1. Develop / Borrow ROS Packages
2. Assemble = Define Launch file
3. Run + Analyze Behavior
4. Stop
5. Fix bugs / Improve
6. Goto step 2

Context is lost
Forgotten details
Live Programming

Edit program at Run-time

Immediate Feedback
Programming with ROS

Node 1

Parameters

Topic

Node 2

Code
ROS Example

draw_square

square_length

velocity

turtle

Code
Requirements for Live Programming with ROS

1. Stop/Start = Replace individual nodes
2. Change parameters at run-time
3. Change code at run-time
4. Change connections = topics at run-time
Requirement 1

• Start/Stop Start = Replace individual nodes

• Problem in ROS:
  – Application = 1 .launch static file
    • Command line Changes are lost
  – Bug when restarting individual nodes
Requirement 1

• Start/Stop Start = Replace individual nodes
  – Problem in ROS:
    • Application = 1 .launch static file
    • Bug when restarting individual nodes
  – Example restarting a turtle
Requirement 2

• Change node parameters at run-time
• Problem in ROS:
  – Parameters fixed in launch file/roslaunch command line
  – Changing them → restart the full app
Requirement 2

• Change node parameters at run-time
• Problem in ROS:
  – Parameters fixed in launch file/roslaunch command line
  – Changing them → restart the full app
• Example: Change square\_length

![Diagram showing setup with draw\_square, square\_length, velocity, and turtle nodes]
Requirement 3

• Change node code at run-time
• Problem in ROS:
  – Use of C/C++ mainly
  – No Hot Code Swapping
Requirement 4

• Change node connections at run-time
  – Change topics names (remape / name space)

• Problem in ROS :
  – Topic names are fixed at launch time
Proposal

- Highly reflective and Dynamic
  - “Everything happens at run-time”
Proposal

• Highly reflective and Dynamic
  – “Everything happens at run-time”

• DSL for hierarchical state machines
  – Integrated with Pharo
Proposal

• Highly reflective and Dynamic
  – “Everything happens at run-time”

• DSL for hierarchical state machines
  – Integrated with Pharo

• PhaROS : ROS Client for Pharo
  – Nodes developed in Pharo
LRP + PhaROS in Action

- Laser Range
- Odometry
- Differential Drive

LRP Automata
PhaROS node

/ros
/command_velocity
/laser
/pose

ROS Graph
Example 1: Stop on Obstacle

stop

On User Command

On Obstacle

forward

state

transition
Example 2: Obstacle Avoider

turnRight

- **Left Obstacle**
- **NO Left Obstacle**

turnLeft

- **Right Obstacle**
- **NO Right Obstacle**

stop

- **NO Obstacle**
- **Obstacle**

forward
Example Video
Towards Live Programming in ROS with PhaROS and LRP

Pablo Estefó\textsuperscript{1}, Miguel Campusano\textsuperscript{2}, Luc Fabresse\textsuperscript{1}, Johan Fabry\textsuperscript{2}, Jannik Laval\textsuperscript{1}, and Noury Bouraqadi\textsuperscript{1}

http://pleiad.cl
http://car.mines-douai.fr