drag&bot as a software platform for development of ROS-based industrial applications

Pablo Quilez
Co-Founder & CTO
<table>
<thead>
<tr>
<th>Year</th>
<th>Event/Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Beginning of research at Fraunhofer IPA</td>
</tr>
<tr>
<td>2015</td>
<td>First business plan</td>
</tr>
<tr>
<td>2016</td>
<td>First prototype feedback</td>
</tr>
<tr>
<td>2017-2018</td>
<td>Positive feedback, First customers and pilot projects</td>
</tr>
<tr>
<td>2019</td>
<td>Financing by Speedinvest (VC), 10+ employees</td>
</tr>
</tbody>
</table>
Different robot hardware requires brand-specific know-how and is complex to configure and use.

It takes a long time to set up a working robot system. This time is missing for “the real” R&D work.

For R&D rapid changeovers are required.
drag&bot is the software platform for simple, graphical setup and programming of robot systems.

drag&bot enables manufacturing companies to flexibly and economically automate small lot sizes.

drag&bot pursues to be for ROS what Windows was for MS-DOS.
Main features of drag&bot

**Intuitive HMI**
Programming by drag&drop.
Experts are 5x faster.
Workers can use robots.

**Uniform HMI**
Same user interface for different robots.

**Plug&produce**
One-click installation of compatible hardware.

**ROS-based**
Extends ROS for graphical use.
Standard use of ROS is allowed.
drag&bot system architecture – online (1)

drag&bot STUDIO: develop
- Website running on browser
- Intuitive
- Graphical

drag&bot RUNTIME: produce
- IPC-based execution environment
- ROS is running inside
drag&bot system architecture – offline (2)

**drag&bot STUDIO: develop**
- Website running on browser
- Intuitive
- Graphical

**drag&bot RUNTIME: produce**
- IPC-based execution environment
- ROS is running inside
drag&bot system architecture – cloud sim (3)

**drag&bot STUDIO: develop**
- Website running on browser
- Intuitive
- Graphical

**drag&bot RUNTIME: produce**
- ROS running inside on the cloud
- Includes a robot simulator with different URDF models and inverse kinematics
- Includes a gripper and a machine tending interactive environment
drag&bot is an open platform
Third parties can create extensions

- **FUNCTION BLOCKS**
  - move robot
  - ERP connection
  - robots
  - vision systems
  - tools

- **WIZARDS**
  - intuitive parameters
  - AngularJS (HTML, JavaScript)

- **GUIDES**
  - step-by-step for complex tasks
  - AngularJS (HTML, JavaScript)

- **DRIVERS**
  - new hardware
  - ROS (C++ / Python)

- **MODULES**
  - new functionality
  - ROS & Non-ROS

- **SOFTWARE**
  - teach-panel
  - trajectory wizardd
  - palletizing
  - screwing

- **TOOLS**
  - 2D vision
  - force
  - bin picking

© 2019 drag and bot GmbH
• Drivers & modules are based on ROS and can be developed in Python or C++.

• Drivers, modules and function blocks can be easily packaged together as d&b Components.

• Catkin overlay space /dnb_catkin_ws automatically loaded by drag&bot.

• The resulting Component can be graphically added and configured.

---

**How to use your own ROS packages in d&b?**

1. **Drivers & modules** are based on ROS and can be developed in Python or C++.

2. **Drivers, modules and function blocks** can be easily packaged together as d&b Components.

3. **Catkin overlay space** /dnb_catkin_ws automatically loaded by drag&bot.

4. The resulting Component can be graphically added and configured.

---

**Config YAML**

- Configurable parameters, startup (ROS Package)

```
package: component name
name: displaying component name
description: description of the component
type: driver/category/subcategory
roslaunch: startup .launch file
configuration: parameters passed as roropams
nodes: ROS nodes running in this component
status_topic: information about the status of the robot
```
How to create associated Function Blocks?

- Defined structure for data formatting & exchange, exception handling
- Function blocks are based on Python and access drivers and modules

Real example of Set IO Function Block

```
# !/usr/bin/env python
import rospy
from robot_movement_interface.srv import *

def execute(input_parameters, output_parameters):
    rospy.wait_for_service('set_io')  # Wait until the service is active
    set_io_srv = rospy.ServiceProxy('set_io', SetIO)  # Create the service interface

    request = SetIOResult()
    request.number = int(input_parameters[0]['value'])  # IO pin
    request.number = int(input_parameters[1]['value'].lower() == "true")

    output_value = str(input_parameters[0]['value'])
    output_value = bool(set_io_srv(request).error)

    output_parameters[0]['value'] = output_value
    return output_parameters
```
Example: Basler 2D Camera & Vision components

- ROS Nodes
  - Basler 2D camera driver
  - Extrinsic calibration module
  - Intrinsic calibration tool

- Component configuration
  - Connection IP, MTU
  - Intrinsic calibration ROS file
  - Active camera profile

- Function Blocks for pattern matching and object localization

Official ROS driver can be found: https://github.com/basler/pylon-ros-camera
## Supported robots and components

<table>
<thead>
<tr>
<th>Robots</th>
<th>Universal Robots</th>
<th>Denso</th>
<th>KUKA</th>
<th>Fanuc</th>
<th>ABB</th>
<th>Yaskawa</th>
<th>Mecademic</th>
<th>Epson</th>
<th>Stäubli (TBA)</th>
<th>Franka Panda (TBA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grippers</td>
<td>Zimmer</td>
<td>Schunk (WSG serie)</td>
<td>Robotiq</td>
<td>All pneumatic</td>
<td>All vacuum</td>
<td>All I/O, IO-Link controlled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cameras</td>
<td>Basler 2D cameras</td>
<td>Balluff Smart-Cams</td>
<td>Sick PIM60</td>
<td>many USB/GigE cameras</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modules</td>
<td>2D / Vision</td>
<td>Force control</td>
<td>PLC</td>
<td>Bin Picking bp3™ Roboception</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>Siemens S7</td>
<td>IO-Link</td>
<td>Modbus TCP</td>
<td>TCP/IP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
drag&bot allows to create ROS-based industrial applications with an easy interface for the user.

Users are able to use, configure and reprogram industrial cells.
drag&bot controls two robots in a human-robot mixed assembly line.

The line runs 24/7.

Handling and screwing processes.

Thanks to drag&bot can the maintenance engineers adapt the robots to new product variants.
What our customers say
Example: fast prototyping – Atlas Copco

With drag&bot you can fast prototype new applications.

New hardware or software modules can be integrated in a couple of days or even hours.

Atlas Copco
Different robots can be flexible used by end-customers.

Standard hardware can be added by one click; Customers can extend the software themselves using Python.

Faster iterations to evaluate new applications of robotics, since the programming is much faster compared to the software of robot manufacturers.

Greater focus on application logic and manufacturing process rather than hardware integration and programming.

Jonas Arnold, Tech-Center Robotics, ZF

„With drag & bot we save a lot of time in our daily work with robots, which we can use now for more important things.“
Create your own robot program
With drag&bot learning factories are able to develop new robot programs.

Lessons with robots
Handling technology can be taught live on the robot.

Skilled work / project teams
Students can develop small and simple applications with the software.

Tasks that are fun
With the simulation environment, simple robot controls are also possible for students without previous knowledge.

Jochen Vohwinkel, Teacher
„drag & bot makes the operation of our KUKA Ilwa really easy - so we can integrate the robot even better into our lessons in the future.“
drag&bot enables research organizations to work more agile and flexible with robots

<table>
<thead>
<tr>
<th>Focus on your research question</th>
<th>Extendable for your specific needs</th>
<th>Do-It-Yourself automation</th>
<th>Flexible modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid complex hardware integration and focus on the work that matters.</td>
<td>Extend the software in Python/C++. Write your own ROS Nodes.</td>
<td>Use robots quickly without extensive training.</td>
<td>Use one robot for different tasks with fast changes.</td>
</tr>
</tbody>
</table>
drag&bot Starter Package

- drag&bot license (either floating or for one robot)
- Service package for first year: updates, support
- Ready-to-use industrial PC
- Online training session

Available add-on modules

- SDK for developers
- 2D vision
- Siemens PLC communication
- Force control (for some robots)

drag&bot at Mecademic booth at ATX Toronto and New York 2019

Watch video:
Interested?
Talk to us!

CONTACT
Pablo Quilez
+49 163 627 3778
pablo.quilez@dragandbot.com
www.dragandbot.com