ROS Tools: roslaunch

ARRA / AR2A

Advancements for **R**obotics in **R**escue **A**pplications

March 13, 2016

ARRA/AR2A

What do we want?

- ARRA / AR2A aims to improve the current state of technology of robotics in rescue applications.

Who are we?

- A volunteer non-profit organisation of robotic enthusiasts.

How can you help?

- Check us out at `https://github.com/ar2a`

License information

Overview 1

roslaunch

roslaunch is a tool for easily launching multiple ROS nodes locally and remotely via SSH, as well as setting parameters on the Parameter Server.

It takes in one or more XML configuration files (with the .launch extension) that specify the parameters to set and nodes to launch.

Overview 2

Advantages:

- Starting implicitly the ROS "core", that's a collection of nodes and programs that are pre-requisites of a ROS-based system.
- The launch file syntax itself is stable, and every effort will be made to provide backwards compatibility with new features.
- Option to set the necessary parameters for the single packages.

Launch Files 1

- Many ROS packages come with "launch files", which can be runned with:

```
$ roslaunch package_name file.launch
```

  These launch files usually bring up a set of nodes for the package that provide some aggregate functionality.

- Launch files can be found under the following path:

```
$ ./src/Package/launch/Package.launch
```

Launch Files 2 - Example 1

Launch-File from example learning_tf

```
<launch>
   <!-- Turtlesim Node-->
   <node pkg="turtlesim" type="turtlesim_node" name="sim"/>

   <node pkg="turtlesim" type="turtle_teleop_key"
              name="teleop" output="screen"/>
   <!-- Axes -->
   <param name="scale_linear" value="2" type="double"/>
   <param name="scale_angular" value="2" type="double"/>

   <node pkg="learning_tf" type="turtle_tf_broadcaster"
         args="/turtle1" name="turtle1_tf_broadcaster" />
   <node pkg="learning_tf" type="turtle_tf_broadcaster"
         args="/turtle2" name="turtle2_tf_broadcaster" />

 </launch>
```

Launch Files 3 - Example 2

Launch-File from example chickenonaraft

```
<launch>

   <!-- chickenonaraft_pub -->

   <node pkg="chickenonaraft" type="chickenonaraft_node"
                               name="sim"/>


 </launch>
```

Launch file calling launch files

- Large applications on a robot typically involve several interconnected nodes, each of which have many parameters. For this you can create a separately package only for the startup launch file:

```
$ catkin\_create\_pkg startup [dependencies]
```

For the startup package are no further dependencies necessary.

- In the separately package the launch file calls the other launch files:

```
<launch>
<include file=
"$(find␣learning_tf)/launch/start_broadcast_demo.launch"
/>
</launch>
```

The End