

rxDeveloper: GUI-Aided Software Development in ROS

Filip Müllers, Dirk Holz, and Sven Behnke

I. INTRODUCTION

The Robot Operating System ROS [1] provides—under the BSD license—libraries and tools that aid software developers in creating applications for robots. In the easiest case, creating a useful application breaks down to connecting and parameterizing existing components or *nodes*. In ROS, tools exist for visualizing the resulting computation graph, checking for correct linkage, examining passed messages in between nodes, as well as monitoring and visualizing communicated data and results. What is does not provide, is a tool for setting up and editing computation graphs, and creating and parameterizing the involved nodes. We present *rxDeveloper* [2], a graphical user interface and tool for creating and editing computation graphs (and launch files).

II. RXDEVELOPER

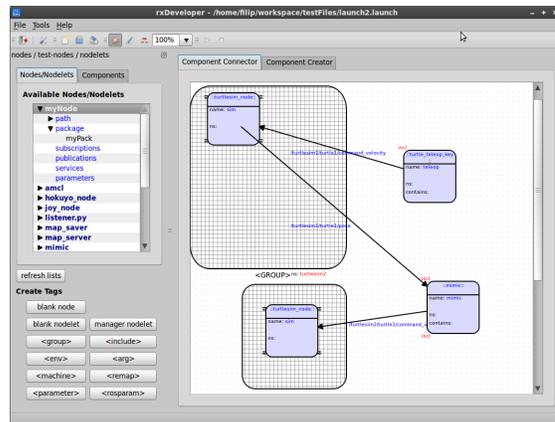
rxDeveloper is primarily meant as a replacement for manually editing launch and parameter files. As in the graph visualization tool *rxgraph*, computation graphs are visualized just like the involved nodes, their connections and parameterization. The user can simply drag and drop existing and new nodes into the graph, perform modifications to node parameters, and save the resulting graph into a launch file. In order to ease rapid prototyping and debugging of applications, *rxDeveloper* allows for directly starting the currently edited graph, and (in combination with the other ROS tools), inspecting and monitoring its execution.

Our tool allows for intuitive arrangements and grouping of nodes. The arrangements on screen are stored in XML comments, i.e., the stored launch files remain compatible. Packages, node specifications, and computation graphs can be created and edited using both drag&drop and specific built-in editors. By supporting the include-file concept, common combinations of nodes can be stored in individual graphs (and dragged or *included* into the current project).

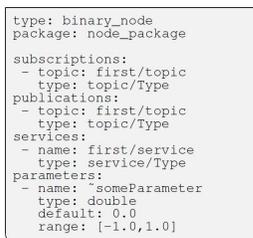
We also introduce the concept of node specification files that carry all node-specific information about subscriptions, publications, parameters (incl. value range) and services. The modular design of *rxDeveloper* allows for extending and customizing the framework, e.g., for automatically generating source code based on node specifications in future releases.

Behind the concepts of *rxDeveloper* is a user-study collecting the requirements of ROS developers and users. The resulting environment integrates well with other ROS tools and the ROS infrastructure.

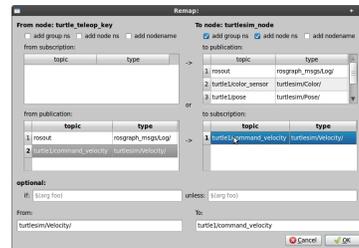
F. Müllers, D. Holz, and S. Behnke are with the University of Bonn, Bonn, Germany. Contact: holz@ais.uni-bonn.de



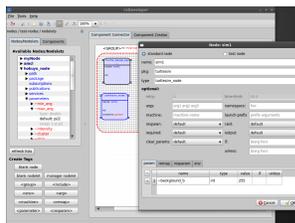
(a) Editing computation graphs and launch files



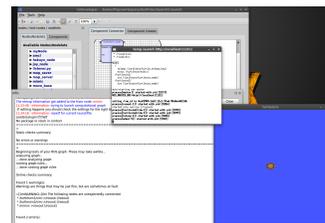
(b) Node specification



(c) Connections and remappings



(d) Grouping and components



(e) Running and testing

Fig. 1: Screenshots and examples of the *rxDeveloper* GUI.

What remains a matter of current and future work is the automatic generation of source code given node specifications, and—in the other direction—the automatic generation of launch files from currently running graphs. Just like many other ROS packages, *rxDeveloper* has been released under the BSD license and is publicly available at [2]. A video tutorial and usage example is available at: www.youtube.com/watch?v=RvGsbhp3kWE.

REFERENCES

- [1] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.
- [2] "rxDeveloper," <http://code.google.com/p/rxdeveloper-ros-pkg>.