R&D project defense Samuel Parra

Date: Friday, 26.03.2021
Time: 10:15
Room: Online

Title: Model-driven engineering support for ROS 2 through meta-models and tooling

Abstract:
The development of a robotic application is an error-prone process since it requires the input of a team of developers, who are experts in different domains and have different responsibilities. The complexity increases with the number of components and interactions in the system architecture. This is true with ROS 2, which introduces improvements and new constraints to the popular framework. Model-driven approaches are enticing to developers since they provide a means to document a system through models, validate them, and transform them into software artifacts. A model-driven approach that supports ROS 2 developers is presented in this work. It consists of an extension to the Component DSL, a modeling language of the VeriComp project, to specify framework-independent software architecture in robotic applications. The ROS 2 extension is composed of three new languages and a modification to the Component DSL.

The ROS 2 DSL adds concepts and annotations to express the missing structural components of ROS in the models of the Component DSL. The QoS DSL and the DDS DSL are two configuration languages that enable the specification of quality of service capabilities and requirements profiles for communication between components. These profiles are linked with the Component DSL through annotations available in ROS 2 DSL since QoS is an important aspect of the new version of ROS. The tooling provides generation of three files: the launch file of the system for ease of deployment, configuration files for configuring the component parameters, and monitor components to supervise the performance of the interactions at run-time.

The presented languages and tooling are evaluated through a case study consisting of an application with a team of multiple robots with quality of service settings. The case study demonstrates the practicality of constraint checks and text generation, especially when multiple QoS profiles and configurations are needed to deploy a system that is reliable to bad network conditions and must be configured correctly. The languages introduced, alongside the tooling and the modified Component DSL, provide an approach that can entice ROS developers to document their system, reduce the time spent in debugging it, and improve the quality of their software.