



MoveIt

Workshop 2019 Macau

Sponsored by





Movelt 2 Progress & Roadmap



Movelt 2.0 Alpha

Initial implementation by Acutronic

Announced: June 2019

ROS2 distro: Dashing Diademata

OS: Ubuntu 18.04 and Mac OS X 10.14

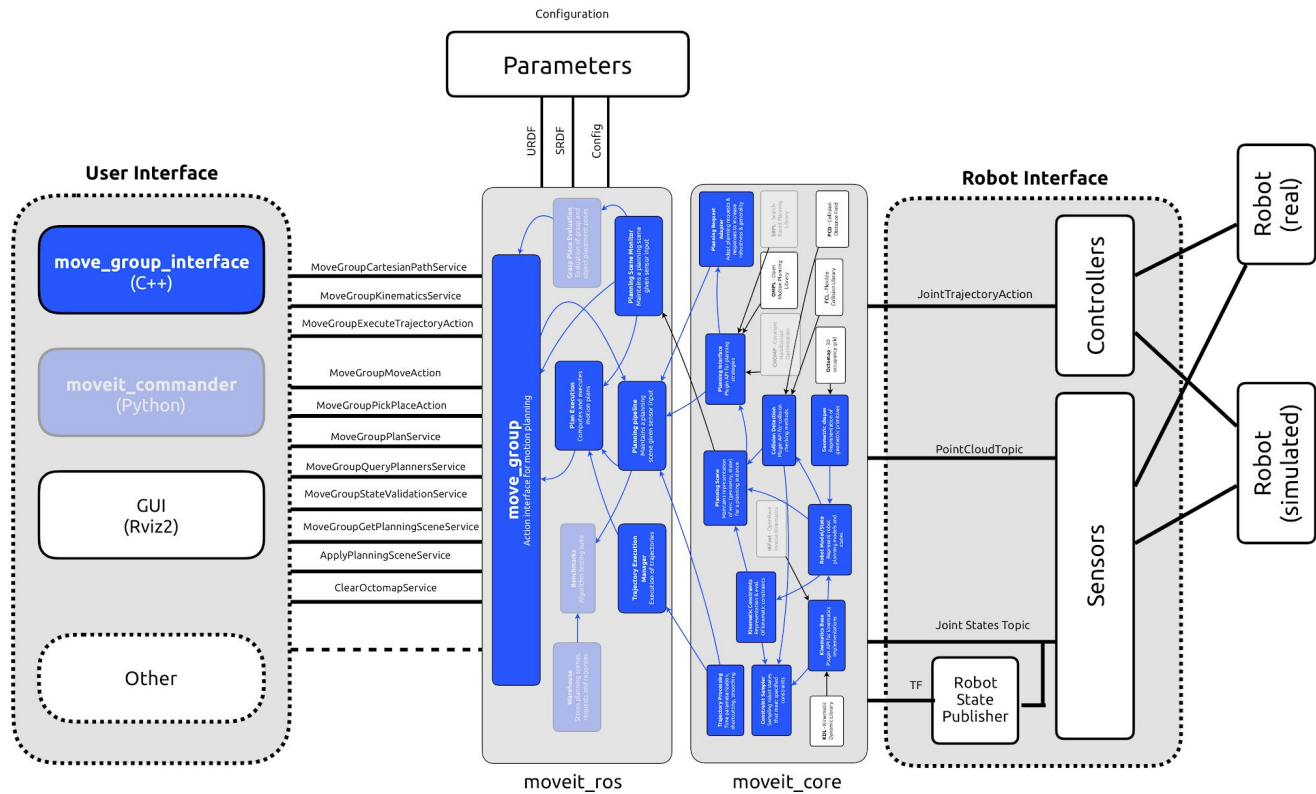




Movelt 2.0 Alpha - Progress

- Most of `moveit_core` `moveit_ros` ported to ROS2
- 11 external dependencies have been ported
- Functional CI infrastructure: `moveit_ci`
- Capability for simple planning to joint-state goal
- Example ROS 2 control framework for Acutronic's MARA robot
- Engaged PickNik to help advise them
- PickNik and the maintainers helped significantly with porting the dependencies, and porting Movelt CI to ROS 2







Future development

Goal: Beta version by Q1 2020

Initial Project Funding: ROSin FTP + PickNik

Two developers working half time for limited development of MoveIt 2

Additional resources still needed for full ROS 2 conversion



Movelt 2.0 Future Development

Roadmap and Milestones



Milestone 1

Straight Port to ROS 2

Fully migrate existing Movelt packages to ROS 2

Wrap up Acutronic's work porting core Movelt functionality

Leverage ROS 2:

Build system (ament), middleware, loc parameters

Cleanup Movelt 2 codebase



Milestone 2

Realtime Support

Reactive, closed-loop control to sensor input

Visual servoing, octomap updates

Preempt motion if new collision detected

Separate global and local planner (hybrid planning)

Global planner (full collision checking): 30hz

Local planner (IK-based, field-based): 300hz

Zero-memory copy integration to controllers (ros_control)

Tighter integration to ros_control

Integrate pilz_industrial_motion

Movelt Survey Results

91% most excited about ROS 2 realtime control

55% reactive planning and closed loop control

48% better integration with lower level realtime control

48% planning with dynamics

Milestone 3

Fully Leverage ROS 2

Lifecycle management of Movelt nodes

Deterministic startup, reset, & shutdown sequences

Leverage ROS2 component nodes

Ability to run Movelt as single or multi-process

Replace pluginlib with components

Cleanup API

More generic and standalone interfaces

Movelt Survey Results

47% excited about component nodes



Realtime Support

- Reactive, closed-loop control to sensor input
 - Visual servoing, octomap updates
 - Preempt motion if new collision detected
- Separate global and local planner (hybrid planning)
 - Global planner (full collision checking): ~30Hz
 - Local Planner (IK-based, field-based): ~300Hz
- Zero-memory copy integration to controllers (ros_control)
 - Tighter integration to ros_control
- Integrate pilz_industrial_motion

PickNik's Vision

Movelt is a globally recognized, highly capable open source manipulation platform that enables organizations of all sizes to leverage robotics for their applications.

Milestone 1

Straight Port to ROS 2

Fully migrate existing Movelt packages to ROS 2
Wrap up Acutronic's work porting core Movelt functionality
Leverage ROS 2:
Build system (ament), middleware, logging, parameters
Cleanup Movelt 2 codebase

Milestone 2

Realtime Support

Reactive, closed-loop control to sensor input
Visual servoing, octomap updates
Preempt motion if new collision detected
Separate global and local planner (hybrid planning)
Global planner (full collision checking): 30hz
Local planner (IK-based, field-based): 300hz
Zero-memory copy integration to controllers (ros_control)
Tighter integration to ros_control
Integrate pilz_industrial_motion

Movelt Survey Results

91% most excited about ROS 2 realtime control
55% reactive planning and closed loop control
48% better integration with lower level realtime control
48% planning with dynamics

Milestone 3

Fully Leverage ROS 2

Lifecycle management of Movelt nodes
Deterministic startup, reset, & shutdown sequences
Leverage ROS2 component nodes
Ability to run Movelt as single or multi-process
Replace pluginlib with components
Cleanup API
More generic and standalone interfaces

Movelt Survey Results

47% excited about component nodes

Future Milestones

Determinism

Out of box / default planners return reliable paths
Tune or replace OMPL, BIT*
Further optimize / smooth paths
Default use TOTG, TOPP time parameterization
Use post-processing optimization (STOMP, TrajOpt)
Fully featured Cartesian Planner
Like Descartes but better and fully integrated
Force-torque control

Improved Interfaces / State Machines

Deprecate the Pick and Place pipeline
Fully support the Movelt Task Constructor
First class support of state machines
Non-ROS C++ API
Similar to MoveGroup but without middleware

Machine Learning

Neural-network based motion planning - new plugins
General near-optimal heuristics for path planning
e.g. MPNet



Group Roadmapping This Afternoon

More to come!