Easy Robot Software

And the MoveIt! Setup Assistant 2.0

Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study
David Coleman, Ioan Sucan, Sachin Chitta, Nikolaus Correll
Journal of Software Engineering for Robotics, April 2014
Outline

- Unique Challenges To Robot Software
- Why Do We Care?
- The 6 Entry Barrier Design Principles
- The MoveIt! Setup Assistant
Building Robot Software Is Hard

So let's all work together :)

PICKNIK
Unique Challenges Facing Robotic Software

● No single developer can have the necessary **domain knowledge**
● Large variety in **complexity** and scale of robotic platforms.
● Software/hardware interaction with **unstructured real world**.
● Long term desire **reduce reliance on GUIs**.
Barriers to Entry

The time, effort, and knowledge that a new user must invest in the integration of a software component with an arbitrary robot.
Why do we care?

Larger User Bases = Better Software
Why do we care? Larger user bases

As number of users increases, bugs are identified and fixed faster [2]

Why do we care? Larger user bases

More users involved in quality assurance, documentation, and support [3]

Why do we care? Larger user bases

New feature contributions increase (weaker correlation) [2]

Why do we care? Larger user bases

Critical mass of skilled contributors has been shown to make open source projects successful [1]

Why do we care? Hiring and Innovation.

Increase number of creative minds working on today's robotic challenge
The 6 Entry Barrier Design Principles
The 6 Entry Barrier Design Principles

**Immediacy**

Minimize the amount of time to accomplish the most basic task.

- Quick start demo
- Cursory feedback to new user that software is worth investing in
- Combat the paradox of the active user
The paradox is that the users would actually save time in the long run if they learned more about the system before attempting to use it, but these studies showed that in reality people do not tend to invest time upfront into learning a new system.

Users never read manuals
The 6 Entry Barrier Design Principles

Transparency

Configuration steps are performed automatically for the user while at the same time being as visible as possible

- Understand what parameters are specific to their robot
- "layered" approach of quick initial setup while allowing later customization as needed
Installation type

This computer currently has no detected operating systems. What would you like to do?

- **Erase disk and install Ubuntu**
  
  Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.

- [ ] Encrypt the new Ubuntu installation for security
  You will choose a security key in the next step.

- [ ] Use LVM with the new Ubuntu installation
  This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.

- [ ] Something else
  You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

[Quit] [Back] [Install Now]
The 6 Entry Barrier Design Principles

**Reconfigurable**

The automatically generated parameters and default values for the initial setup of a robot should be easy for the user to modify at a later time.

- Typically chosen to work with the largest number of robots
- Not optimal for any robot
- Varying applications require different configurations
The 6 Entry Barrier Design Principles

Intuitive

The need to read accompanied documentation, and the amount of required documentation, should be minimized.

- Follow standard design patterns
- Provide interface context clues
- Ideally an interface does not require additional documentation
The 6 Entry Barrier Design Principles

**Extensible**

The user should be enabled to customize as many components and behaviors as possible within the reasonable scope of the software.

- Makes the software far more powerful and re-usable for varying use cases
- Plugin interface
The amount of reference material explaining how to use the software should be maximized for as many aspects and user levels as possible.

- No software is intuitive enough to not require documentation
- Different types of documentation are required for different types of users
  - Developers vs end-users
The 6 Entry Barrier Design Principles Documented
Movelt! Setup Assistant
Quick Setup of MoveIt!

MoveIt! Setup Assistant

These tools will assist you in creating a Semantic Robot Description Format (SRDF) file, various yaml configuration and many roslaunch files for utilizing all aspects of MoveIt! functionality.

Create new or edit existing?

All settings for MoveIt! are stored in the MoveIt! configuration package. Here you have the option to create a new configuration package or load an existing one. Note: changes to a MoveIt! configuration package outside this Setup Assistant are likely to be overwritten by this tool.
Optimize Collision Checking

Optimize Self-Collision Checking

This searches for pairs of robot links that can safely be disabled from collision checking, decreasing motion planning time. These pairs are disabled when they are always in collision, never in collision, in collision in the robot's default position, or when the links are adjacent to each other on the kinematic chain. Sampling density specifies how many random robot positions to check for self collision.

Sampling Density: Low \[ \rightarrow \] High \[ \leftarrow \] 100000

Min. collisions for "always"-colliding pairs: 95%  \[ \rightarrow \] Generate Collision Matrix

<table>
<thead>
<tr>
<th></th>
<th>panda_link0</th>
<th>panda_link1</th>
<th>panda_link2</th>
<th>panda_link3</th>
<th>panda_link4</th>
<th>panda_link5</th>
<th>panda_link6</th>
<th>panda_link7</th>
<th>panda_hand</th>
</tr>
</thead>
<tbody>
<tr>
<td>panda_link0</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_link1</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_link2</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_link3</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_link4</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_link5</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_link6</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_link7</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>panda_hand</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Specify Metadata

- Planning Groups
- Robot Poses
- End Effectors
- Passive Joints
- Virtual Joints

Define Virtual Joints

Create a virtual joint between a robot link and an external frame of reference (considered fixed with respect to the robot).

<table>
<thead>
<tr>
<th>Virtual Joint Name</th>
<th>Child Link</th>
<th>Parent Frame</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtual_joint</td>
<td>panda_link0</td>
<td>world</td>
<td>fixed</td>
</tr>
</tbody>
</table>
MoveIt!

MoveIt! Setup Assistant 2.0

Special thanks to Mohmmad EI Khzragy, Open Robotics, and GSoC
Auto-configure Depth Sensors

Setup 3D Perception Sensors
Configure your 3D sensors to work with MoveIt! Please see Perception Documentation for more details.

Optionally choose a type of 3D sensor plugin to configure:
None
Setup Gazebo Simulation Integration

Simulate With Gazebo

The following tool will auto-generate the URDF changes needed for Gazebo compatibility with ROSControl and MoveIt!. The needed changes are shown in green.

You can run the following command to quickly find the necessary URDF file to edit:

```
roscol franka_description
```

[Button] Generate URDF
Setup ROS Control

Setup ROS Controllers

Configure MoveIt! to work with ROS Control to control the robot's physical hardware.

- **Auto Add FollowJointTrajectory Controllers For Each Planning Group**

<table>
<thead>
<tr>
<th>Controller</th>
<th>Controller Type</th>
</tr>
</thead>
</table>

**Options:**
- Expand All
- Collapse All
- Delete Controller
- Add Controller
- Edit Selected
Hello World
Generated Quick Start Demo
New MoveIt! Tutorials

MoveIt! Quickstart in RViz

Getting Started
Step 1: Launch the Demo and Configure the Plugin
Step 2: Play with the Visualized Robots
Step 3: Interact with the Panda
Step 4: Use Motion Planning with the Panda
Next Steps

Move Group C++ Interface
Move Group Python Interface
MoveIt! Commander Scripting
Robot Model and Robot State
Planning Scene
Planning Scene ROS API
Motion Planning API
Motion Planning Pipeline
Visualizing Collisions
Time Parameterization
Planning with Approximated Manifold
Pick and Place Tutorial
MoveIt! Setup Assistant
URDF and SRDF
Low Level Controllers
Perception Pipeline Tutorial

The quickest way to get started using MoveIt! is through its RViz plugin. RViz is the primary visualizer in ROS and an incredibly useful tool for debugging robotics. The MoveIt! RViz plugin allows you to setup virtual environments (scenes), create start and goal states for the robot interactively, test various motion planners, and visualize the output.

Getting Started
If you haven't already done so, make sure you've completed the steps in Getting Started.

Step 1: Launch the Demo and Configure the Plugin
World Movelt! Day 2018

Thursday, October 25th, 2018
Conclusion

- Robot software is hard
- Make your software easier to use
- Apply the entry barrier design principles
- By building a community around your software, the software gets better
- The MoveIt! Setup Assistant is a good example