



**Hochschule  
Bonn-Rhein-Sieg**  
University of Applied Sciences

# Software Development Project

Final Presentation

---

Wei-Chan Hsu, Torsten Jandt, Ramesh Kumar, Danning Wang

July 10, 2017

# Introduction

---

# Basic Navigation Test

- Environment: Workspaces, waypoints and obstacles.
- Task specification: Sequence of poses.



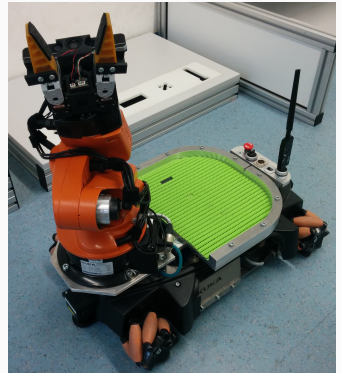
# Challenges

- **Perception:** Accessing and processing sensor data.
- **Mapping:** Building map of the environment.
- **Localization:** Pose inside map.
- **Path planning:** Determine sequence of poses between waypoints.
- **Motion control:** Execution of path.

# KUKA youBot

The youBot is a mobile manipulator designed for education and research purposes. It comes with fully open interfaces and API.

- Omnidirectional, four-wheeled
- 5-DOF manipulator with a two-finger gripper
- On-board PC with CPU, 2GB memory, 32GB SSD drive
- Sensors: vision sensors, rangefinders



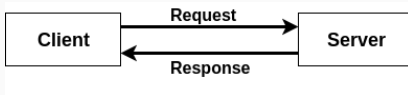
# Robot Operating System (ROS) I

Set of software and libraries.

- **Node:** A process using ROS.
- **Topic:** Message queue, used for communication between nodes.



- **Service:** Offers synchronous service calls.



# Robot Operating System (ROS) II

- **Actionlib**

- Provides client interface to send requests to server
- Client and server communicate with messages:
  - Goal
  - Feedback
  - Result

# Approach

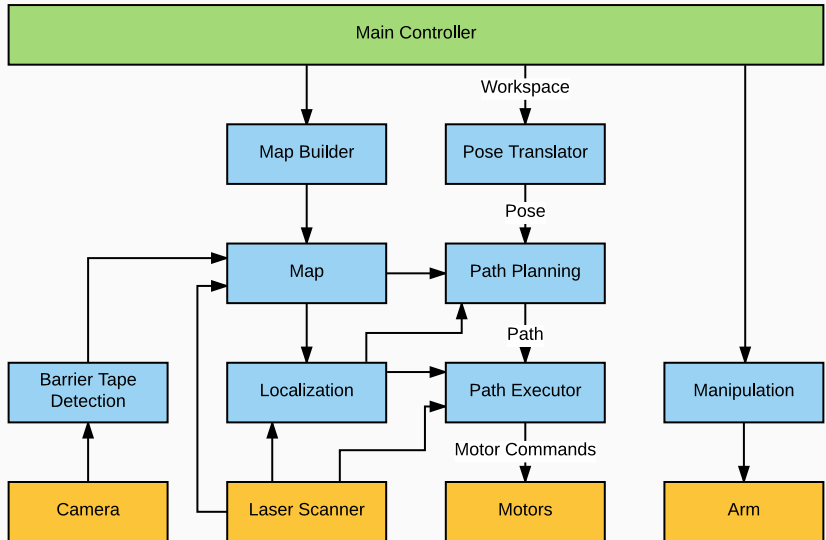
---



# Approach

- Divide problem into smaller parts.
- Each part is defined by
  - It's function and
  - Interface
- Parts are replaceable.
- Try to use as many available components from ROS as possible.

# Software Modules



# Realization

---

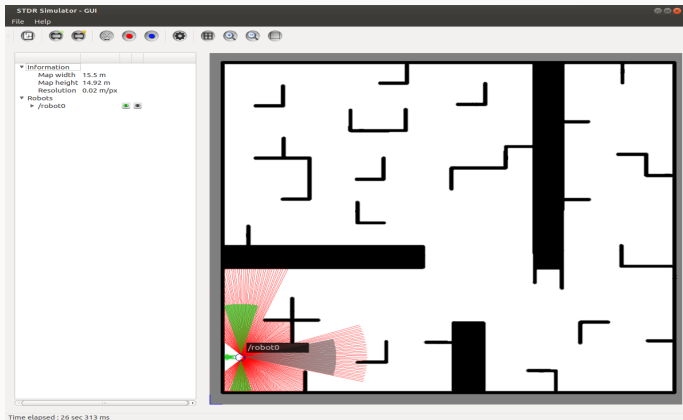
# Realization

---

## Simulation

# Simulation

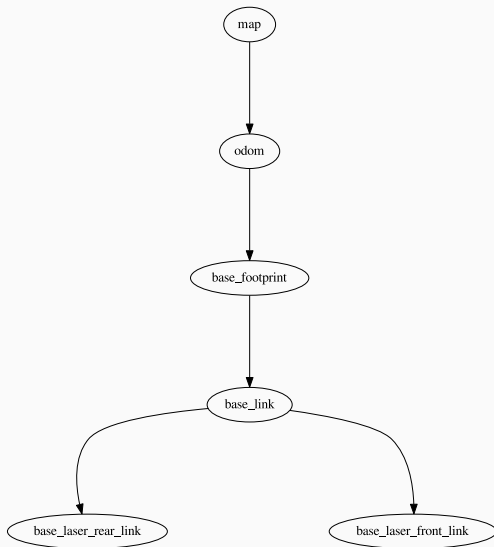
- Simple Two Dimensional Robot (STDR) simulator
- Tasks performed:
  - Map Building
  - Localization



# Map building I

- Gmapping is used to build 2D occupancy grid map
- Uses laser sensors to build the map
- Map Server
  - Provides map saver utility, to save generated map in files(yaml and pgm)
  - Offers map data as a ROS Service

# Map building II



# Localization I

- Adaptive Monto Carlo Localization(AMCL) is used to localize the robot
- Uses particle filter to track the pose of robot
- Problems:
  - AMCL could not find laser data on /scan topic
  - AMCL node crushes after some time
- Solutions:
  - Remap /scan\_front and /scan\_rear to /scan topic
  - Because of transformations provided by STDR simulator



# Realization

---

KUKA youBot

- Driver for the KUKA youBot robot
- youBot API represents robot system as a combination of sub-systems
- Contain three main classes
  - youBot Manipulator
  - youBot Base
  - youBot Joint
- Uses ROS wrapper for translation to ROS

# Map building III

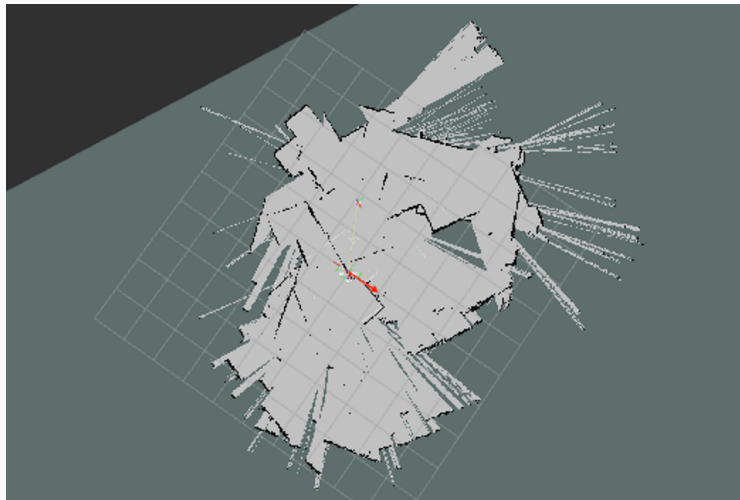
- **Problems**

- Messy Map

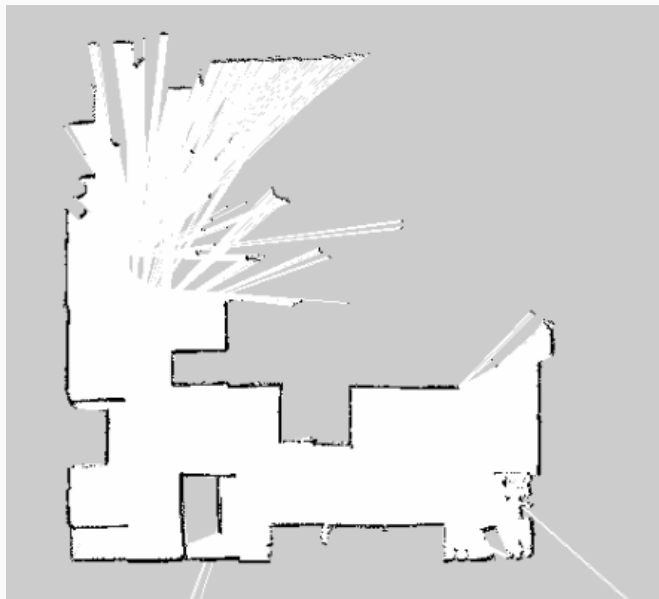
- **Solutions**

- Because of different translations in simulation and real world
  - Configuration of laser scanners in simulator is upside down

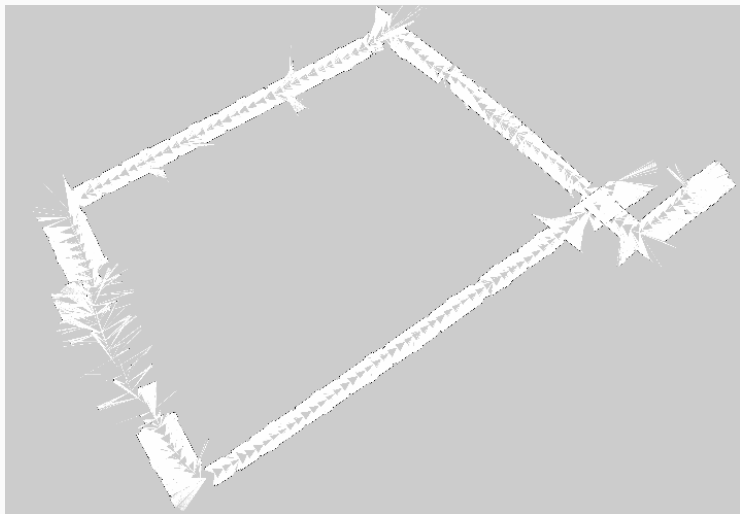
## Map building IV



# Map building V



# Map building VI



# Navigation

- Requirements
  - Map (map\_server)
  - Localization (amcl)
  - Odometry source
  - Transforms
  - Sensor sources
  - Goal (move\_base)
- Planners: global, local
- Costmaps: global, local
- Local planner: dwa\_local\_planner
  - Given: plan, costmap, odom
  - Generates costs of transversing through map grids
  - Output: Velocity command
- Output: Velocity command (cmd\_vel)

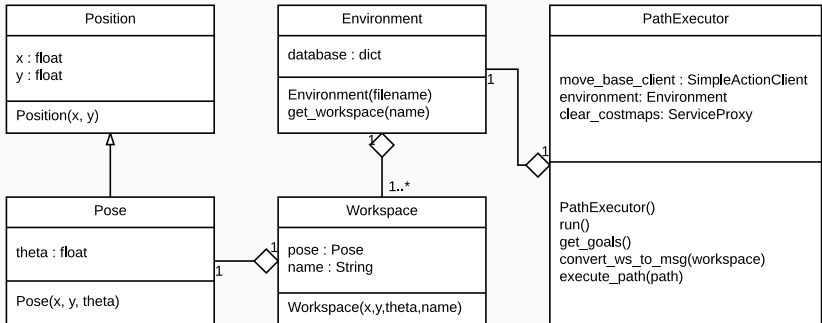
# Path executor

Path executor reads a set of user inputs and convert them to `move_base_msgs`.

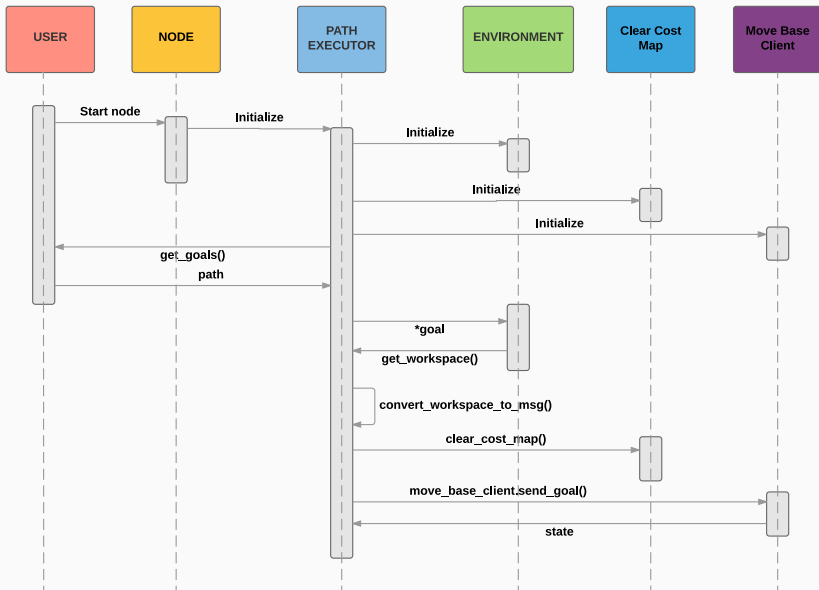
- Class: Position, Pose, Environment, Workspace, PathExecutor
- Functions:
  - Reads user inputs
  - Reads workspace from file
  - Converts workspace to `move_base_msgs`
  - Clears cost map
  - Sends goal message



# Class Diagram



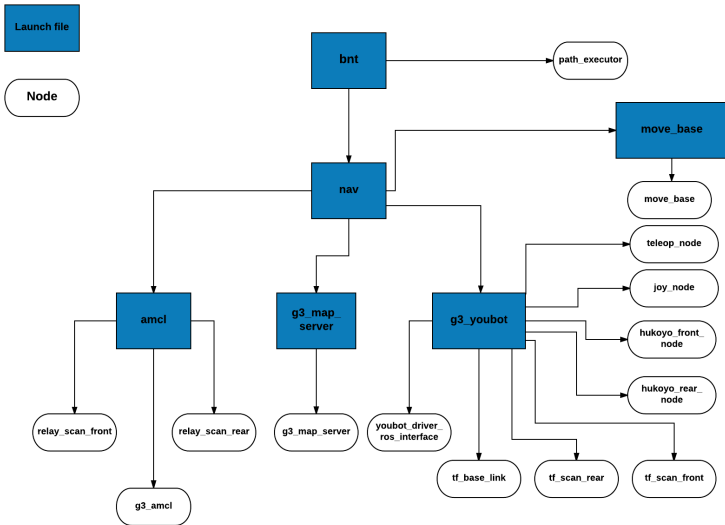
# Sequence Diagram



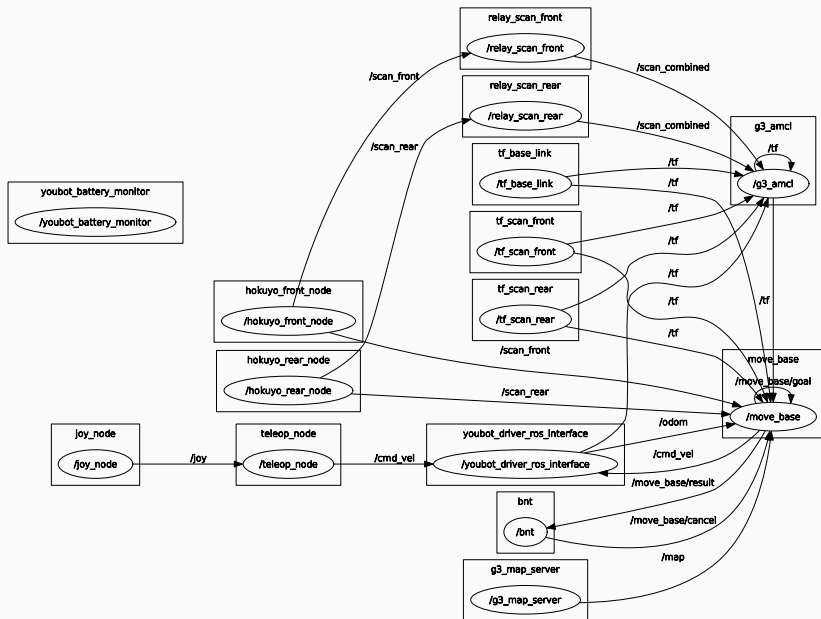
# Results

---

# Launch Files



# RQT Graph



# Conclusions

---

# Conclusions

- Navigation was analyzed and applied to youBot.
- The task contains mapping, localization, path planning, motion execution.
- Mapping was realized using gmapping, which requires laser scans and correct transforms.
- Localization was achieved using AMCL, which relies on a map, laser scans, transforms, and initial pose.
- DWA local planner was employed for path-planning.
- A node was created as a path executor that requests user input.
- The robot was able to navigate around the lab by user input of a series of workspace.

# Future Work

- User interface can be improved.
- The parameters can be retuned to improve the performace of robot.