Che-Yen Lu
Team E: PLAID
Teammates: Michael Beck, Akshay Bhagat, Matt Lauer, Jin Zhu
ILR02
October 21, 2016

1. Individual progress

For this project, we need to leverage the resource from RI faculties and last year's team, which means we should stand on their shoulder to achieve higher success. My primary role is to design new software architecture. New software architecture is needed due to requirement for flexibility and maintainability. After discussion with senior, Alex, the features for new software are as follows.

- Modularity
  - The software should divide into sub-systems properly
  - The interface between each sub-system should keep clean and coherent
- Easy to Test
  - Hardware should be replaced with minimal software modification
    Ex: yaml file configuration
  - Record point cloud data and log file for issue analysis
- Version Control
  - GIT version control should be used to ease the pain of integration
  - Review meeting should be held once a week
  - Sub-version for every sub-systems

In the project early stage, it's hard to sketch out the draft of software architecture without any experience, so I decide to roughly go through SMACH state machine, item classification, item localization, and grasping planner from last year's codebase. After I analyze the code, the planner from SBPL lab and planner service could be reused, but perception, grasping and state machine should be created on our own.

The new software architecture consists of four rosnodes, which are system control, perception, arm control, and grasping. The principle of dividing nodes is based on functionalities. I want to minimize the number of rosnodes, but the system still keeps the flexibility and the potential to run sub-systems in parallel.

Basically, the whole system flow is controlled by system control node, and it will issue commands to other three nodes. Every node need to parse the command and execute the task it has been asked for. Since those commands are blocking calls, the system control node will wait until the result is returned. For arm control node, the task is pretty simple. It moves arm to the specific position client request. The request could specify either bin number or arm poses. As for the perception node, it is responsible for localization and classification. The point cloud and confidence scores for every items will be generated for grasping and system control node. Finally, the grasping

node take point clouds of bin and items as inputs to generate grasping poses. After generating poses, it will try to use arm service to move end effector to grasping position and grab target item. The flow chart for picking task is listed below. The dashed line boxes stand for rosnodes, and solid line boxes represent services or functionalities nodes provide.

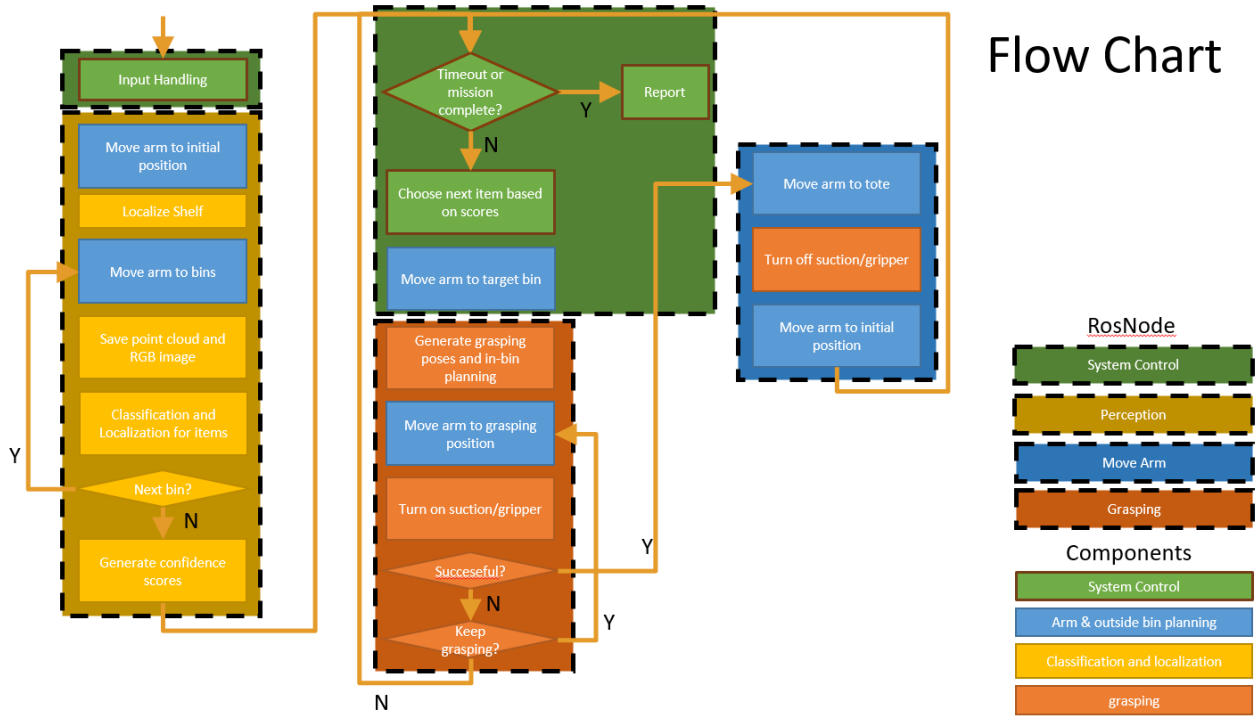

Figure 1. The software flow chart for picking task

Also, the interfaces between sub-systems are also defined below.

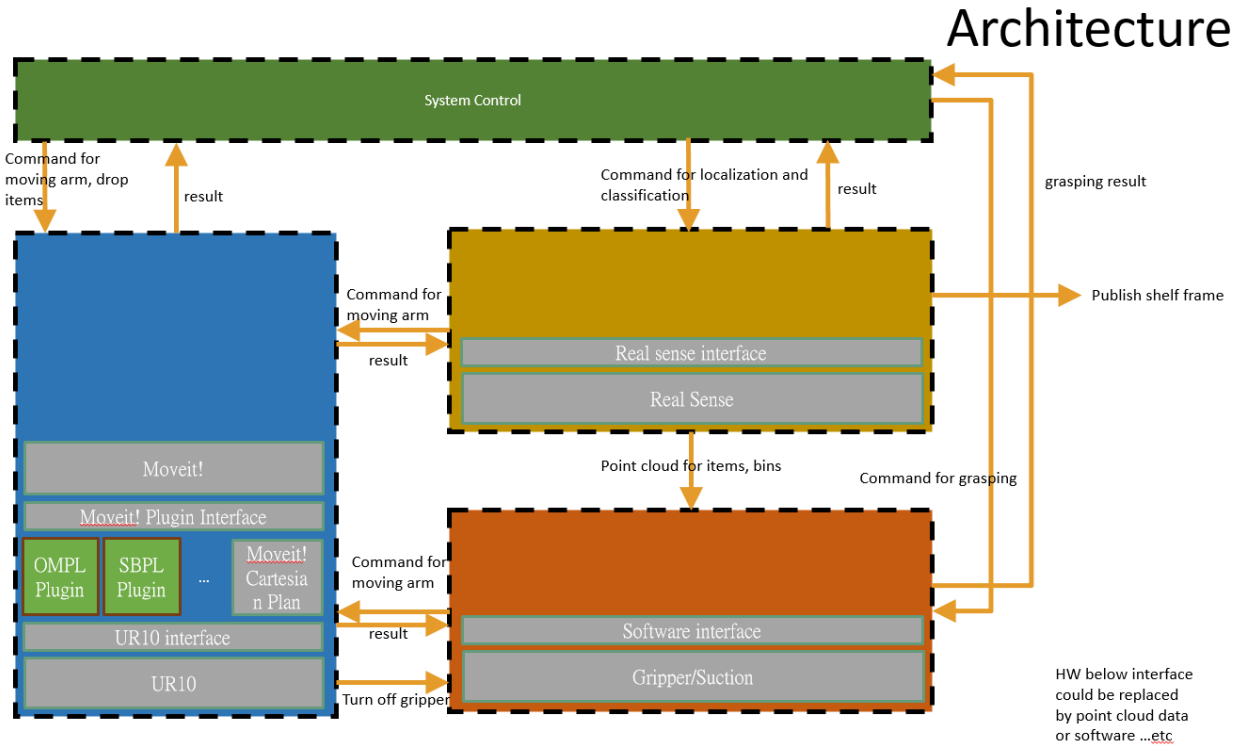

Figure 2. Software interface and architecture

2. Challenges

The biggest challenge so far is to familiarize with ROS. Since we need to understand the code and software architecture from last year's codebase, we suffer from learning numerous ROS package they have ever used. Even if I spend much time on ROS and getting familiar with it, understanding PCL, SMACH, CAFFE package is a totally different story. I need to manage myself through it to be more productive.

The second challenge worth mentioning is package installation and environment setting. Handling package dependency and solving build error is time-consuming. If I use a new package no one has ever tried before, all I can do is to google the solution and give it a try. Although I can learn a lot from solving such issues, it reduces productivity. I think writing the installation guideline should be helpful if this project will be taken over by MRSD students next year.

3. Teamwork

For this project, we focus on different domains and break down the tasks as follows:

- Michael Beck – Gripper. Survey existent facilities and develop prototype/CAD model of end effector, including suction cap, two finger gripper and suction with one finger gripper.
- Akshay Bhagat – Perception. Survey and choose the RGB-D camera based on metrics. Fuse point cloud data from PCD file.
- Matt Lauer – Arm Planning. Survey the maneuverability of linear moving base. Get simulation up and running on UR10. Deploy planner on UR5.
- Che-Yen Lu – Software Architecture and Perception. Create cheat sheet for team. Design new software architecture. Help perception group to survey algorithm for fusing the point cloud data.
- Jin Zu – Perception. Work on the turntable, which will be used for Alex's CNN.

.

4. Future plans

After our group have consensus about the new architecture, I will focus on implementing the new software architecture. The software architecture will be based on the refactoring harp code, and the harp_arm package will be reused. I will create the communication interface for all four rosnodes so that every member could take advantages on it. Also, master and development branch on github will be created for development and integration. Moreover, I will try to define the coding convention and survey if there is any test suite for ROS.