Ability Mapping

This project was inspired by the things I learned in my elective 325 AI Programming. Ability Map is what I'll call this ros-package. It will take the description of services, their inputs and outputs, and the way in which these descriptions set up creates a set of rules that specifies what functions to run based on what information it has to complete a task. The end user will only need to write a .yaml config file. The package uses a combination of roslisp and either roscpp or rospy in order to make decisions on what functions to call and when. Roslisp will utilize a package that utilizes symbolic deductive reasoning.

– Fallback goals:

- The package roslisp will be able to call services that are written in either rospy or roscpp.
- A user will be able to use a .yaml file to dictate what services are called.
- I will be able to use this package in place of SMACH for my fall 495 final-project.

– Core goals:

- Core functionality will include the deductive reasoner in order to call services.
- User will be able to issue commands to give a robot different objectives.
- Package should match the flexibility of SMACH or other finite state machine available on ROS.

– Reach

- Another MSR student will be able to use the package in their project if they see fit and the integration of my package into their project should be quick and easy.
- I will be able to look through past projects done with more complex robots and use my package to have them achieve over arching goals.
- Get the Ridge Back to where I can ssh run my package on it and tell it to look for something with the ability to detect and open doors.
-

I'm hoping to learn about:

- Roslisp,
- Meta-Programming
- ROS
- AI
- Robot Control and Autonomy

Tasks

- Create a method for a node to call a rospy or roscpp service based on what a roslisp node broadcasts (1 Week)
- Create a method for roslisp node to create a set of frame rules based on the format of information given in a config.yaml (1 Week)
- Test and debug deductive reasoner so that it can reproduce the reliable behavior shown in my fall 495 group final project. (2 Weeks)
- Publish package (3-5 days)
- Apply the package to other robots and test. (The remaining Quarter)

The greatest risk I take with this project has to do with the deductive reasoner. I have no idea what I'll need to modify in order to get the project to work. Also Lisp is still a relatively new language I've learned which presents its own issues. Currently I don't know how to dynamically call a service.

Here is the issue: I know roslisp will be taking in info on services from the parameter server and will decide on which service to call, however I don’t know what node will need to call the service itself. I don’t even know if what language in which the service calling node should be written. I’m uncertain about how the package will be able to equate the actual service to its symbolic counter part in roslisp in a manner that will keep user involvement with the users application of the package to a minimum. In order to address the issue of the deductive reasoner and Lisp I’ll simply read up on and practice with the language and read over the documentation of the package. In order to address the other uncertainties I’ll have to do experiments and take note of which theories seem to work or not.

- The cs325 deductive reasoner lisp packages
- ROS
- Turtlebots
- Franka Emika Panda
- The marker holding station that was built last quarter
- Jackal Robot
- Other hardware will be requested as I test my package.