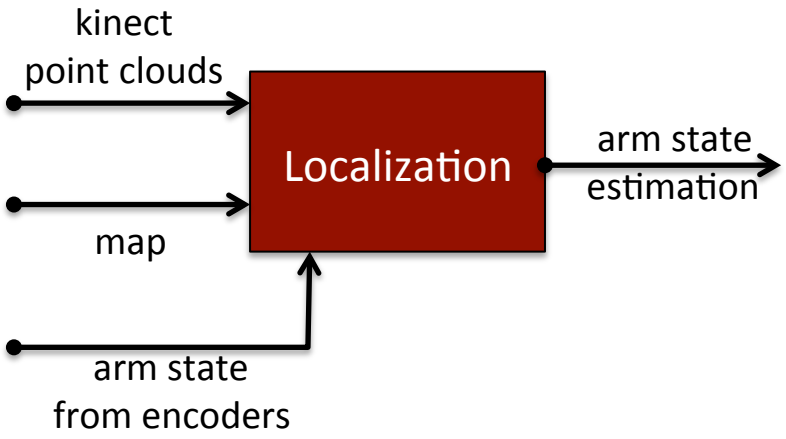
Amazon Picking Challenge

- Software components that need to be developed and integrated
- Assignment of tasks
- System architecture



Team:

- Alberto
- Ainesh
- Shaojun
- Colin

Purpose:

Provide an accurate estimation of the arm's state

Simple version:

Just forward the information provided by the arm encoders

Final version:

Correct information from the encoders by using the currently constructed map and the kinect point clouds (arm + body)

Team:

- Alberto
- Colin
- Shaojun
- Ainesh

Purpose:

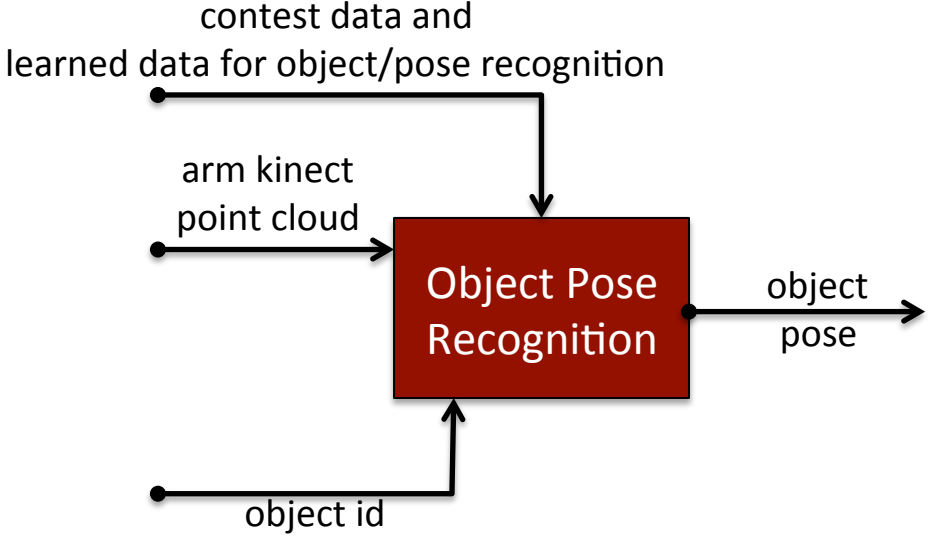
Provide a map of the world for collision-checking purposes

Simple version:

Initially use a static kinect on the body to provide a map

Final version:

Use the localization module's output for the arm state and integrate the data from the arm's kinect



Team:

- Shaojun
- Alberto
- Colin
- Ainesh
- Tarek

Purpose:

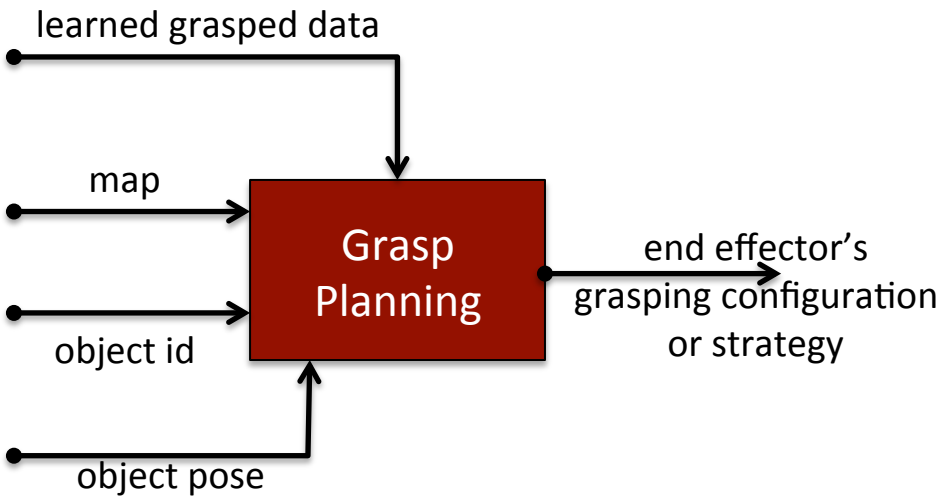
Identify the pose of an object in the shelf

Simple version:

Do it for a single item

Final version:

- Solve the problem for all the items
- Deal with multiple items of the same type in the same shelf



Team:

- Colin
- Thanasis
- Shaojun
- Rahul
- Alberto

Purpose:

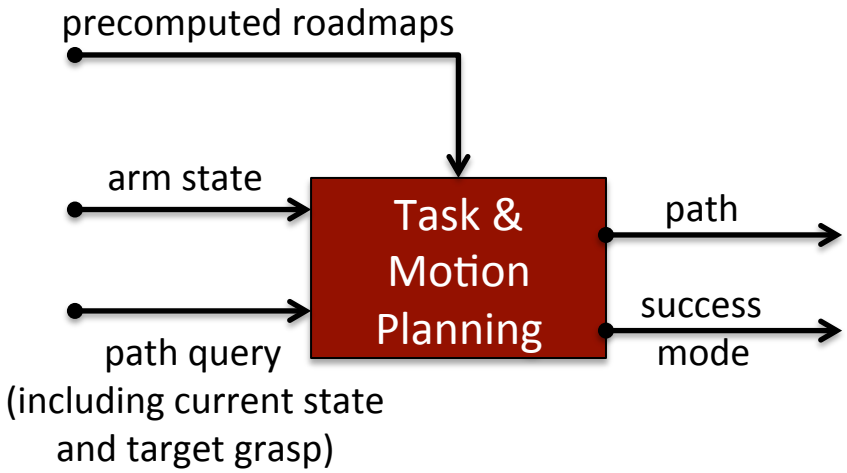
Identify the grasping configuration/strategy for the robot's hand

Simple version:

Initially assume only grasping with gripper, assume objects in the center of the shelf in a standard pose (no other objects) - ignore map

Final version:

- Take the map into account
- Deal with multiple objects in the shelf in general configurations



Team:

- Chuples
- Rahul
- Zakary
- Andrew
- Kostas

Purpose:

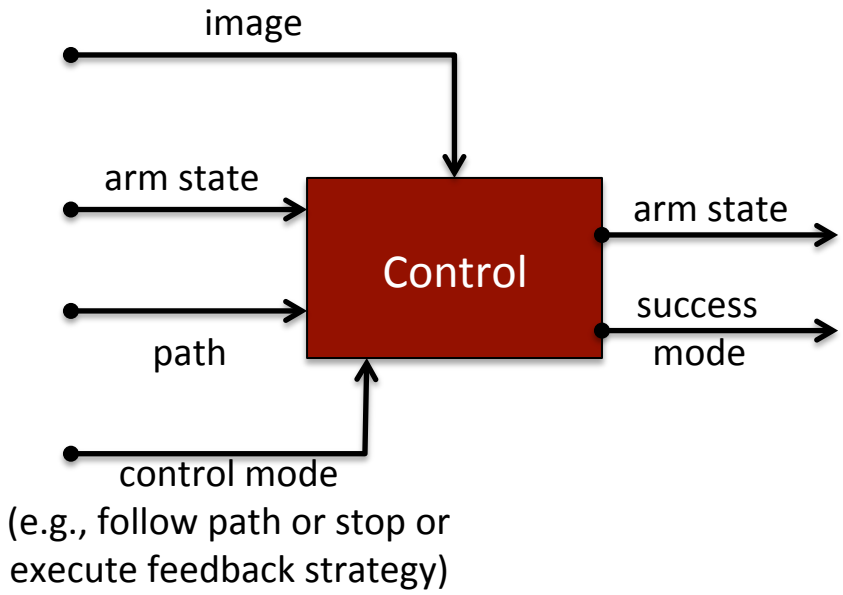
Compute paths for the robot

Simple version:

Initially assume simple setups where one of the arms can always easily grasp a single object in the shelf

Final version:

Increase the complexity of the setup



Team:

- Rahul
- Andrew
- Zakary
- Chuples
- Kostas

Purpose:

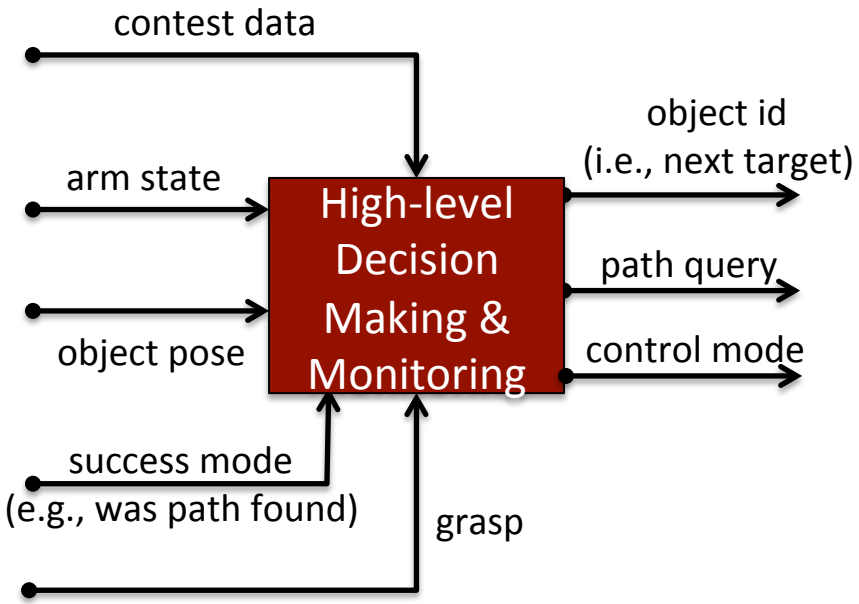
Provide controls to the robot

Simple version:

Initially just follow a path computed by the motion planning module, interrupt execution if not followed appropriately

Final version:

Provide a feedback strategy for following the path or more complex visual servoing strategies for complex objects



Team:

- Alberto
- Kostas
- Chuples
- Rahul

Purpose:

Error monitoring of the process and high-level decision making

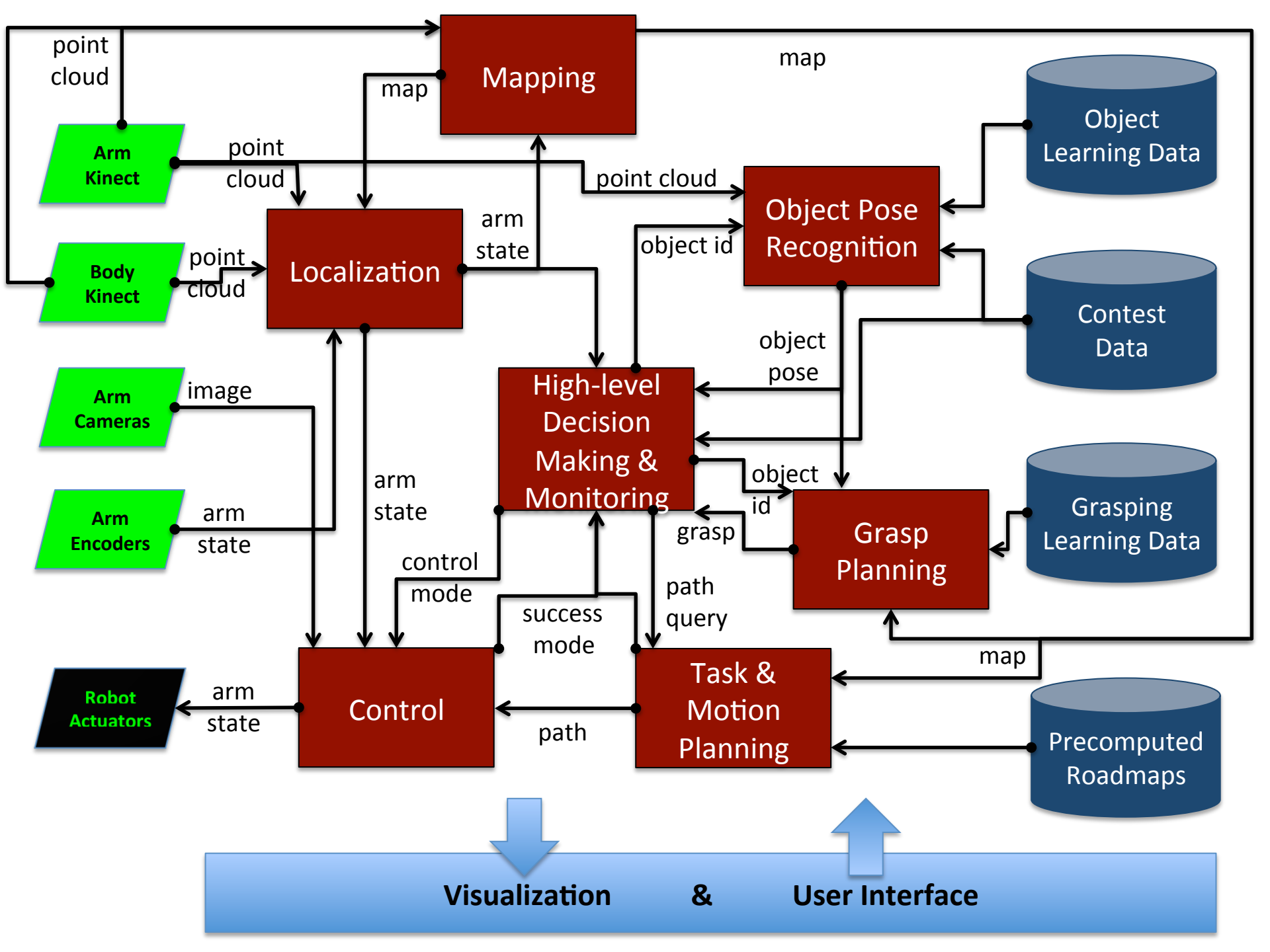
Simple version:

Map the space, plan a path for the arm in front of the bin, find pose of an object, identify grasp, compute path and execute it

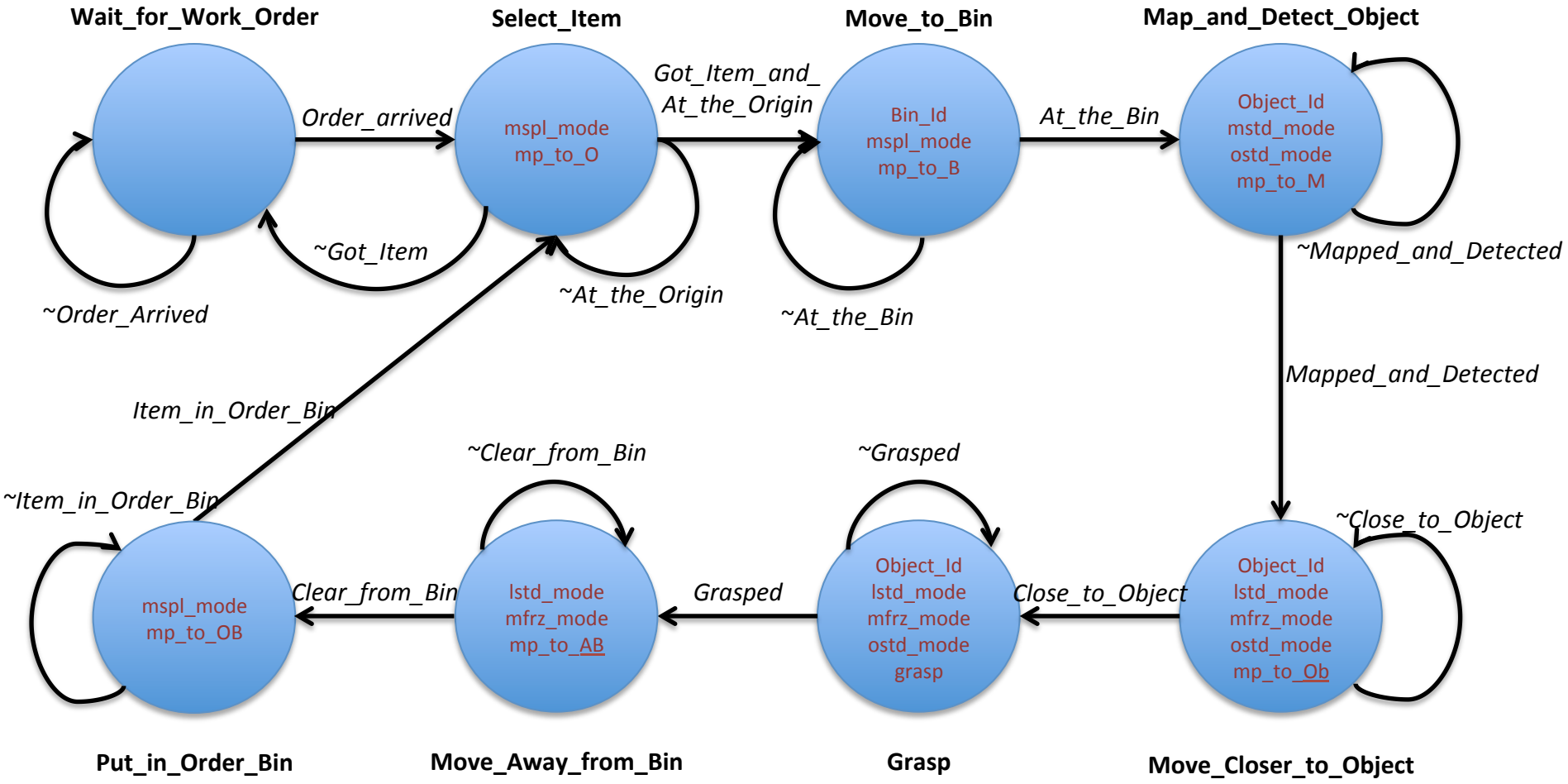
Final version:

Deal with the complete challenge: select order of objects, use both arms simultaneously, handoffs (??)

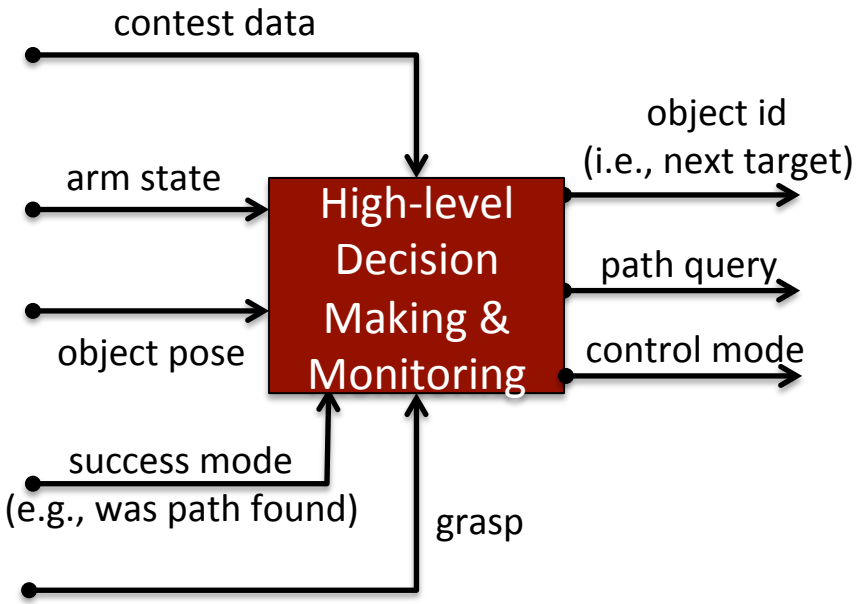
Full System Architecture


point cloud
Mapping
map
map
Arm Kinect
point cloud
point cloud
Object Learning Data
Object Pose Recognition
object id
Body Kinect
point cloud
Localization
arm state
Contest Data
object pose
Arm Cameras
image
High-level Decision Making & Monitoring
object id
Arm Encoders
arm state
arm state
grasp
Grasp Planning
Grasping Learning Data
control mode
path query
success mode
map
Robot Actuators
arm state
Control
Task & Motion Planning
path
Precomputed Roadmaps
Visualization & User Interface

High-level Decision Making & Monitoring: State Machine



High-level Decision Making & Monitoring model publish the **system state**



Team:

- Alberto
- Kostas
- Chuples
- Rahul

Purpose:

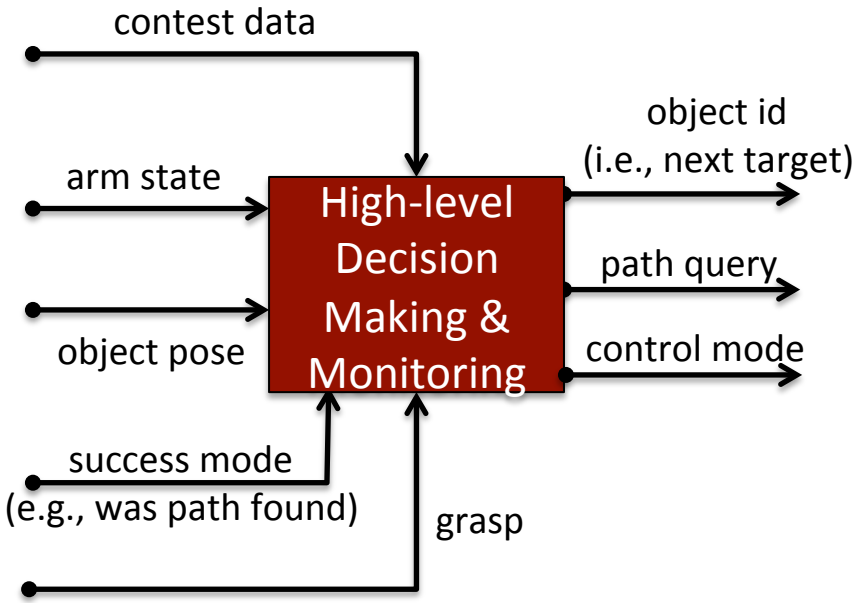
Error monitoring of the process and high-level decision making

Simple version:

Implements the state machine

Final version:

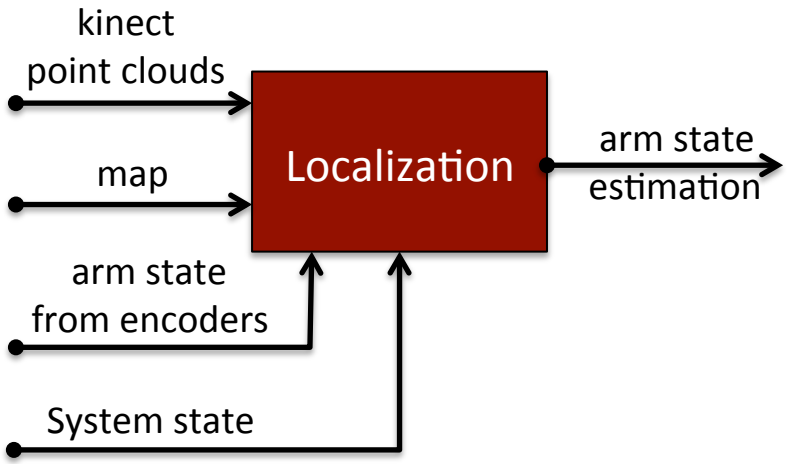
Deal with the complete challenge: select order of objects, error monitoring, use both arms simultaneously, handoffs (??)



Team:

- Alberto
- Kostas
- Chuples
- Rahul

- This module' state machine operates at a frequency of about 40Hz
- Each circle of the state machine represents one state
  - In **bold** we have the name of the state
  - In *italics* we have the conditions (flags) for state transition, which are examined at each cycle
  - In color we have the information (variables) that is published together with the state
    - Binary variables are only shown when true
    - Non-binary variables are only shown when they have a valid value



Team:

- Alberto
- Ainesh
- Shaojun
- Colin

Purpose:

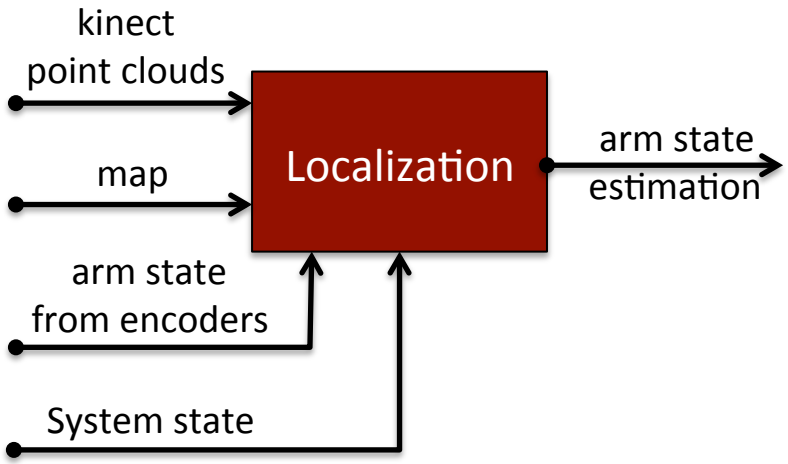
Provide an accurate estimation of the pose of the sensor in the arm with respect to the map

Simple version:

Publish the pose of the sensor in the arm with respect to the robot using the arm's encoders

Final version:

Publish the pose of the sensor in the arm with respect to the map



Team:

- Alberto
- Ainesh
- Shaojun
- Colin

- Operates in two modes, **lstd_mode** and **~lstd_mode**, governed by a binary variable published together with the system state
  - **~lstd_mode:** Publish the pose of the sensor in the arm with respect to the robot using the arm's encoders. The world reference frame is provided by the map.
  - **lstd_mode:** Publish the pose of the sensor in the arm with respect to the map – use the kinect point cloud and the map to compute the pose (and does not use the arm's encoders). The world reference frame is provided by the map.

Team:

- Alberto
- Colin
- Shaojun
- Ainesh

Purpose:

Provide a map of the world for collision-checking purposes

Simple version:

Publish a fixed model of the shelf

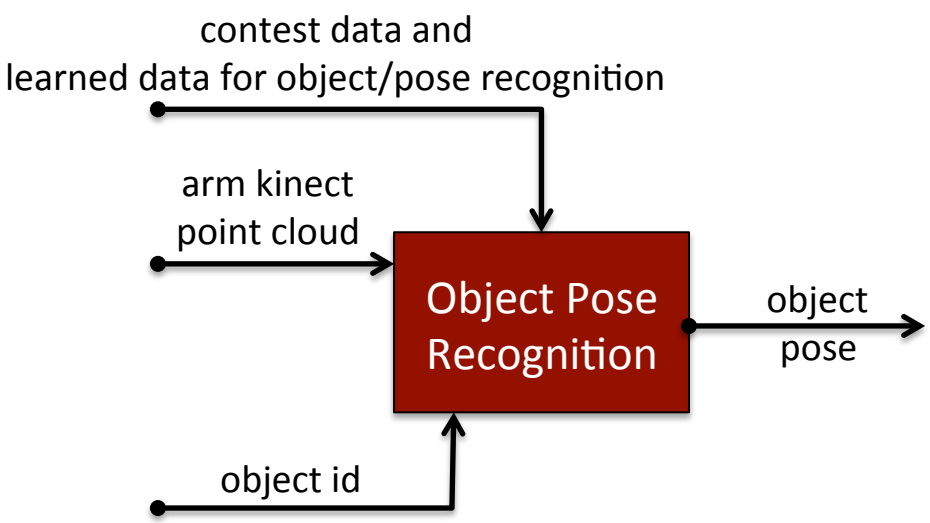
Final version:

Build and publish a map of a target bin of the real shelf and its surroundings

Team:

- Alberto
- Colin
- Shaojun
- Ainesh

- Operates in three modes: **mspl_mode, mstd_mode, mfrz_mode**
  - **mspl_mode:** Publish a fixed model of the shelf (in world coordinates) at an approximate expected position with respect to the robot.
  - **mstd_mode:** Build and publish a map of the target bin of the real shelf and its surroundings continuously.
  - **mfrz_mode:** Publish the last map of the target bin of the real shelf and its surroundings continuously.



Team:

- Shaojun
- Alberto
- Colin
- Ainesh
- Tarek

Purpose:

Identify the pose of an object (Item) in the shelf

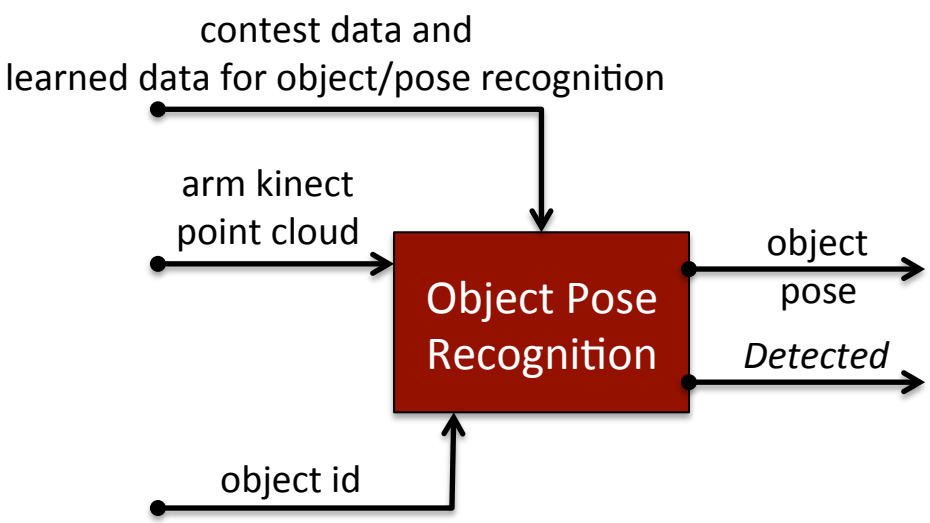
Simple version:

Do it for a single item in a limited number of positions

Final version:

Solve the problem for all the items

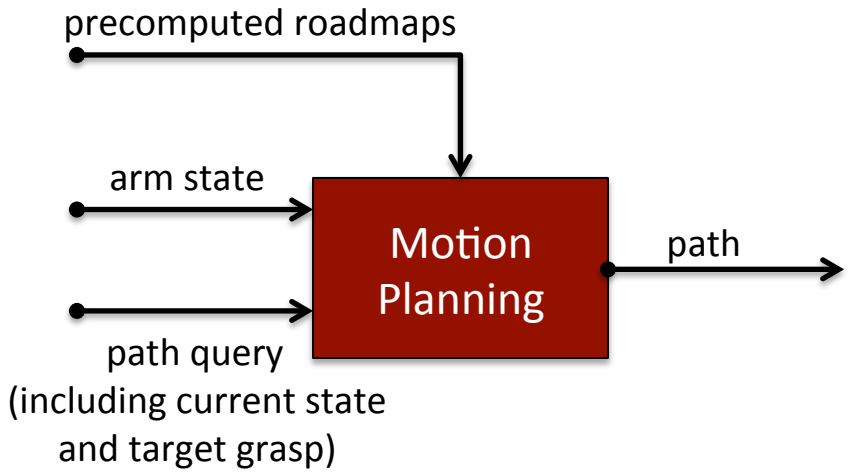
Deal with multiple items of the same type in the same shelf



Team:

- Shaojun
- Alberto
- Colin
- Ainesh
- Tarek

- Operates in two modes: **ostd_mode** and **~ostd_mode**
  - **~ostd_mode:** Do nothing.
  - **ostd_mode:** Publish the pose of the detected object with respect to the map. Publish the flag *Detected* when the object is found.



Team:

- Chuples
- Rahul
- Zakary
- Andrew
- Kostas

Purpose:

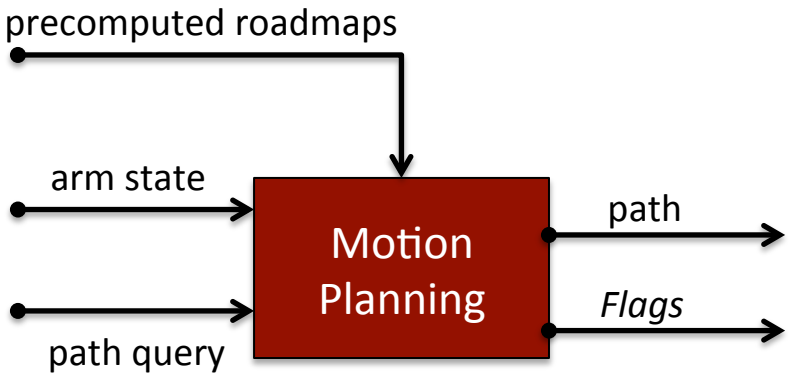
Compute and publish trajectories for the robot

Simple version:

Publish pre-defined trajectories or compute and publish trajectories from current position to given target position not considering obstacles

Final version:

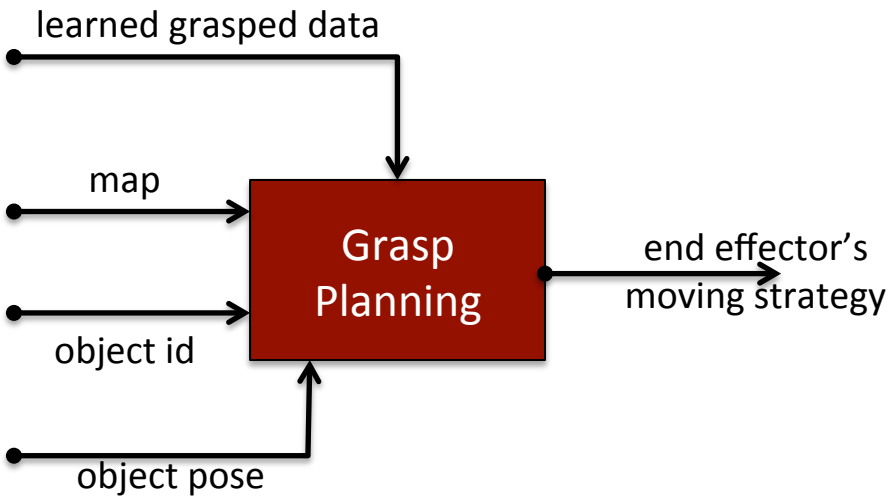
The same but avoiding obstacles



Team:

- Chuples
- Rahul
- Zakary
- Andrew
- Kostas

- Operates in two modes: publish pre-defined trajectory X, **mp_to_X**, and compute and publish trajectory from current position to target position X, **mp_to_X**
  - **mp_to_X:** Publish pre-defined trajectory X. The origin of the robot is fixed in the world. Publish the flag *At_the_Origin* after executing mp_to_O, the flag *At_the_Bin* after executing mp_to_B, the flag *Mapped* after executing mp_to_M, and the flag *Item_in_Order_Bin* after executing mp_to_OB.
  - **mp_to_X:** Continously compute, publish and monitor the trajectory to move the position of the sensor in the arm (with respect to the map) to target position X (with respect to the map). At the beginning of each step of the movement (or serie of steps), computes the position of the robot origin given the sensor pose, and then publish the trajectory to move the robot. Repeat until the sensor is in the X position. Publish the flag *Close_to_Object* after executing mp_to_Ob, and the flag *Clear_from_Bin* after executing mp_to_AB.



Team:

- Colin
- Thanasis
- Shaojun
- Rahul
- Alberto

Purpose:

Identify the grasping configuration/strategy for the robot's hand

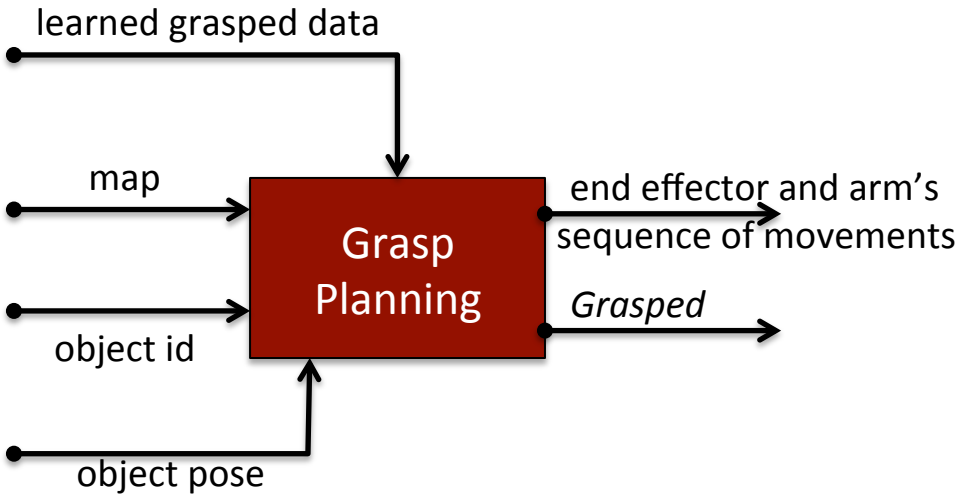
Simple version:

Assume only grasping with gripper and that objects are in standard pose free of obstacles and in front of the gripper - ignore map

Final version:

Take the map into account

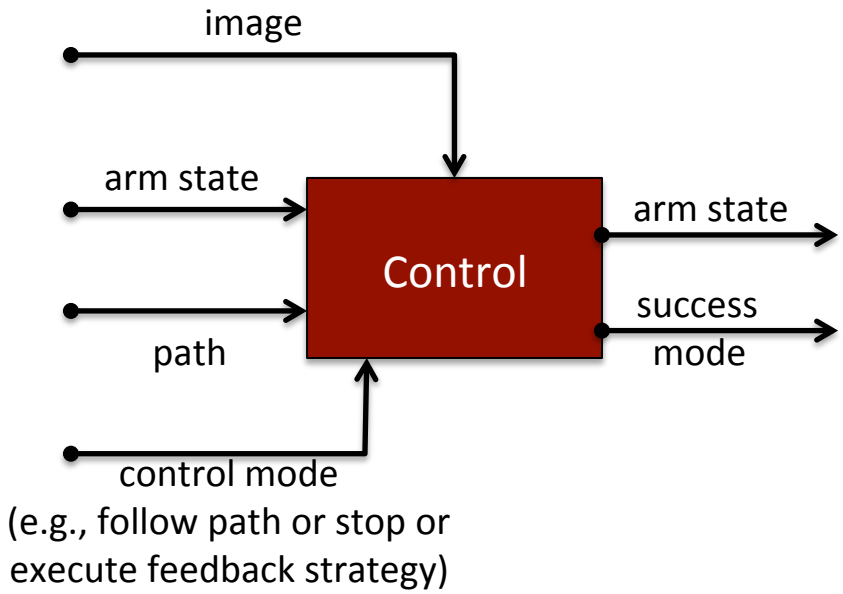
Deal with multiple objects in the shelf in general configurations



Team:

- Colin
- Thanasis
- Shaojun
- Rahul
- Alberto

- Operates in two modes: **grasp** or **~grasp.**
  - **~grasp:** Do nothing.
  - **grasp:** Continuously compute, publish and monitor sequence of movements for grasping the item. Publish the flag *Grasped* when finished.



Team:

- Rahul
- Andrew
- Zakary
- Chuples
- Kostas

Purpose:

Provide controls to the robot

Simple version:

Just follow the trajectory computed by the Motion Planning or the Grasp Planning modules (interrupt execution if not followed appropriately)

Final version:

Provide a feedback strategy for following the trajectory