Modelling the Environments and the Robots

Kick-Off Project Presentation

Elif Erbil, Eesha Kumar

Software Engineering in Robotics
Summer Semester 2019

Use Case (High level)

Warehouse exists with/without items and with robots. Provide a model to move/process items in a warehouse using the robots.

Alexa bring me item ‘x’	Robot goes to storage unit with ‘x’	Robot returns with storage unit
The workflow begins with user (near a conveyor belt) issuing a command to retrieve item ‘x’ from the warehouse.	An API is triggered which identifies the location of ‘x’. The API finds a suitable robot and moves the robot to location of ‘x’.	The robot reaches the required position and returns with the appropriate storage unit. User/Helper picks up the item from shelf.

Components and Architecture

A simulation environment - warehouse with storage space and robots

Components of a Warehouse

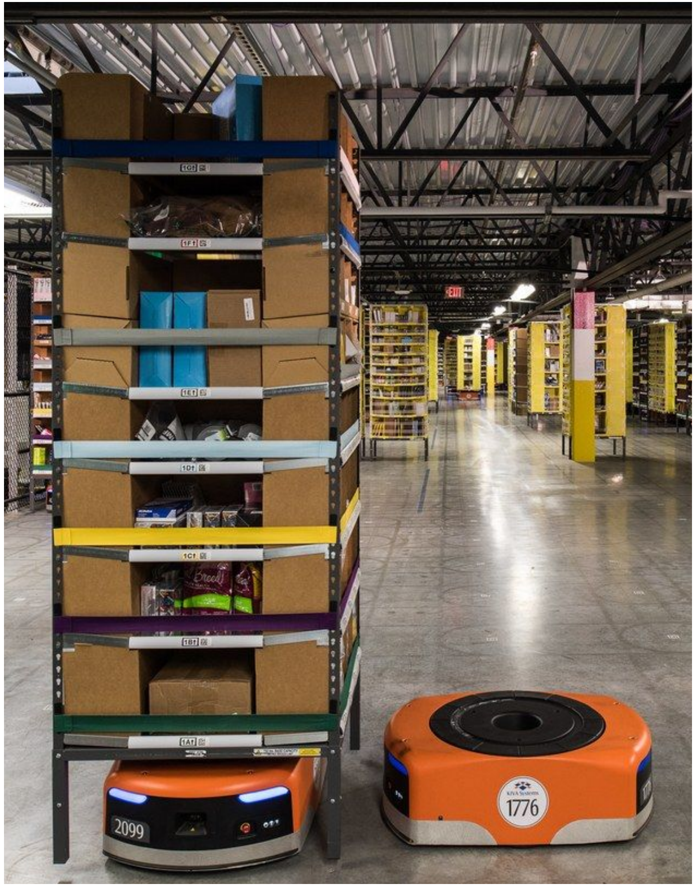
- Storage Units
  - Shelves
  - Items
- Carrying (Kiva) Robots
- Conveyor Belts
- People Using the Conveyor Belts (or Grasping Robots)

Warehouse Assumptions and Limitations

- Warehouse will be modelled as a grid to facilitate movement.
- Robots can move beneath the storage units.
- Each storage unit must be placed in such a way that it can be moved anytime without being blocked.
- Warehouses are provided with default sizes depending on choice of input. I.e. S, M, L etc.

Storage Unit

- Each storage unit consist of shelves which can hold multiple items.
- Storage units will help optimize the lookup for items and orders.
- Storage units will have constant dimensions but will vary in weight.



Kiva Robots

Functionality:

- Moving (moveP2P())
- Rotating (rotate())
- Lifting storage units (liftShelf())
    - Receives input for weight estimation
- Transmitting its location and status (transmitPosition(), transmitOrientation())
- Transmitting sensor information (transmitSensorInformation())

Kiva Robots

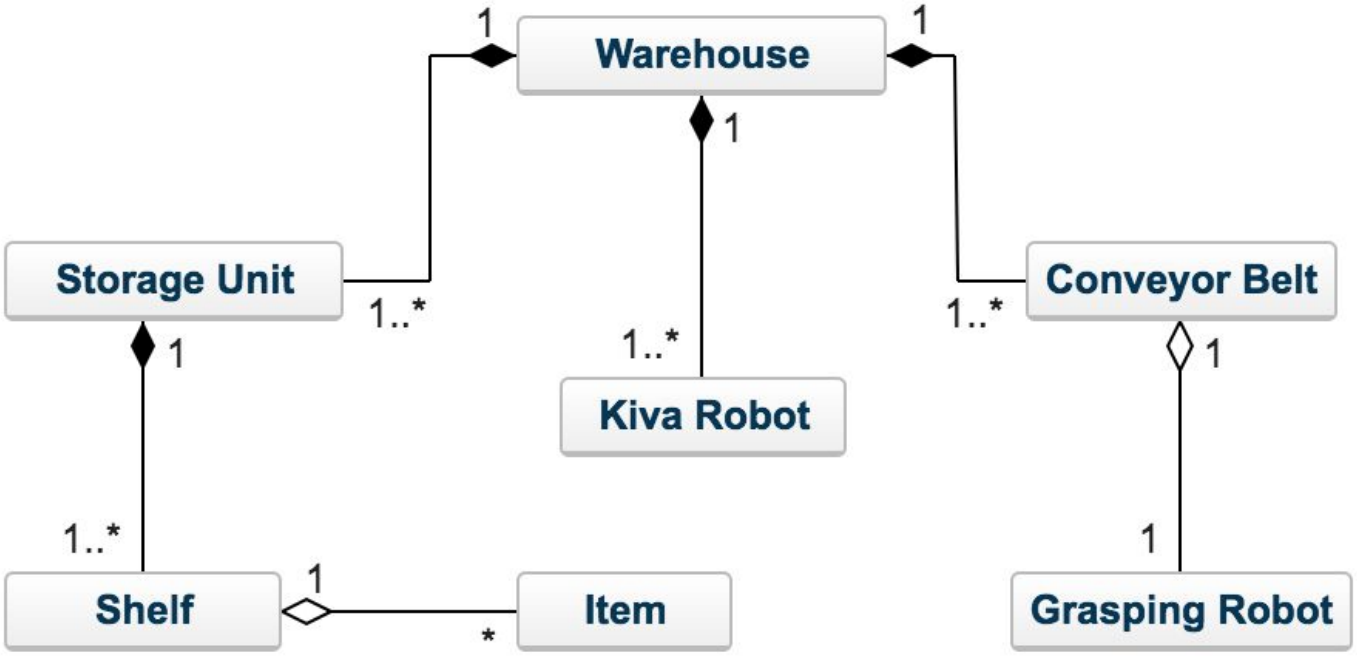
Necessary Sensors and Hardware:

- Sonar Sensor (collision detection)
- Force/Torque Sensor (actuator/joint movement estimation)
- Wireless Transmitter and Receiver (information flow)
- GPS Sensor (position and orientation)
- Actuator (to lift shelves)
- Actuator (speed check, brakes)
  - (goal based agent) - Sensor/actuator feedback loop.

Conveyor Belts

- Initially static objects in the warehouse which serves as the destination of the Kiva robots.
- A warehouse may have one or more conveyor belts which may require optimizations for the warehouse layout to keep storage units closer to conveyor belts if necessary.
- People controlling the conveyor belts can be switched with grasping robots in the future versions.
- Conveyor belt drop locations are marked and specified.

Warehouse Structure/Architecture



Requirements

- Software: A linux OS, cloud services for service API and environment integration.
- Hardware: Sensors, Transmitters, Receivers, Actuators.
- Models: Warehouse, Storage Units, Shelves, Items, Orders, Robots.
- Further Models: Conveyer belt objects, Coordinators.

Responsibilities

- Environment setup(Gazebo + ROS integration) and modelling of robots is shared effort.
- Modelling the warehouse and its objects will be divided once we decide/narrow down on requirements.
- Pos, Torque, Weight estimation/relaying, Actuator facility and similar functionalities will also be divided.
- Sensor-Actuator loops for speed check, movement will be divided effort.
- Any requirements specified during the presentation can be added and effort shared/divided accordingly.

Collaboration

- Input from other groups
  - Warehouse attributes - size, number of robots
  - Warehouse configuration - places of items and orders to the conveyor belts
  - Commands to be run on the robots from the API
- Output to other groups
  - Warehouse layout
  - Sensor information from Kiva robots
  - Robot actuator output/confirmation
  - Graphical simulation in Gazebo

Milestones

- Modelling the Kiva robots with sensor, actuator and transmitter functions
- Modelling a simple warehouse
- Controlling the robots for some basic tasks in a static warehouse model
- Adding ids to items in the storage units to control which items are being carried
- Parametrization of warehouse attributes (size of the warehouse, number of kiva robots and conveyor belts)

Timeline

First Status Update

Basic models for the warehouse and its components

Basic models for the Kiva robots and sensors

Midterm Presentation

Core functionality of Kiva robots with inputs from sensors

A fixed warehouse model to test the functionality of Kiva robots

Second Status Update

Integration with the Web API to control the movement of the robots

Adding ids to items on the shelves to create realistic simulations

Final Presentation

Parametrization of the properties of the warehouse and creating a warehouse according to these parameters and simulating the motion of robots

Suggestions and Final Demo

- Initializing the warehouse with parameters from the web API or Alexa.
- [OPTIMIZATION] Changing the layout/location of storage units/number of Kiva robots depending on various factors. e.g. usage frequency and assignment.
- Collecting path information for storage units for movements patterns of Kiva robots.
- [CONCEPT] An order includes multiple items. when a user asks for an order, all available items in that order need to be brought be robots.
- [PUB SUB] For read/write operations while managing inventory and warehouse.