Architectural Document

Nozama-Hotmetal

Soyeon Shin, Dongkeun Kim, Hwijeen Ahn, and Minseok Kang

Date: February 14, 2020

Contents

1 Introduction

1.1 About This Document

This document describes Nozama-Hotmetal's autonomous package delivery system. This includes the project description provided by the project organizers and our work on the project. We followed the typical software engineering procedure; analyzing project requirements, identifying architecture drivers, building structures based on the architecture drivers, and the implementation. This document will provide our thought in each components. The expected reader of this document include developers, architects, and stake holders. We note that we provide brief explanations about technical terms used in software engineering literature in order to aid readers of different background.

1.2 About Team Members and Roles

Nozama-Hotmetal is a team of four people from different background. We did not have a team leader so as to promote democratic democratic workspace. At the early stage of the project, all the team members participated in discussing architectural drivers and structure. Once the initial design choices were made, each one of us took responsibility for different tasks. Each one volunteered to take a task according to their interest and strength. We note that the division of labor was not exclusive and collaboration and question across tasks were promoted.

Role	Person in Charge	Responsibilities
Developer, Security Manager	Hwijeen Ahn	Server, Data collection, Simulator
Developer, Quality Assuarance	Minseok Kang	Scheduling, Robot control, Order management
Developer, Project Manager	Soyeon Shin	Order generation, Sign detection, User interface
Developer, Safety Manager	Dongkeun Kim	Line tracing, Obstacle detection, Robot control

Table 1: Team members and their roles

2 Project Overview

2.1 Project Description

This project aims to develop autonomous package delivery system. The Nozama company wants to test autonomous package delivery system in the test environment before full scale deployment. 8 teams are hired as consultants and should design and implement a system to accept online customer orders, make a schedule for delivery, deliver the packages to different addresses, and allow workers load/unload packages to win a contract.

3 Architectural Drivers

Architectural drivers are requirements that will shape the software architecture. Architecture drivers were constantly revised after peer meetings, mentor meetings, and standup evaluation. In this document, we describe the final version of architecture drivers. We utilize three perspectives to describe our software architecture: Functional requirements, Quality attributes, and Constraints.

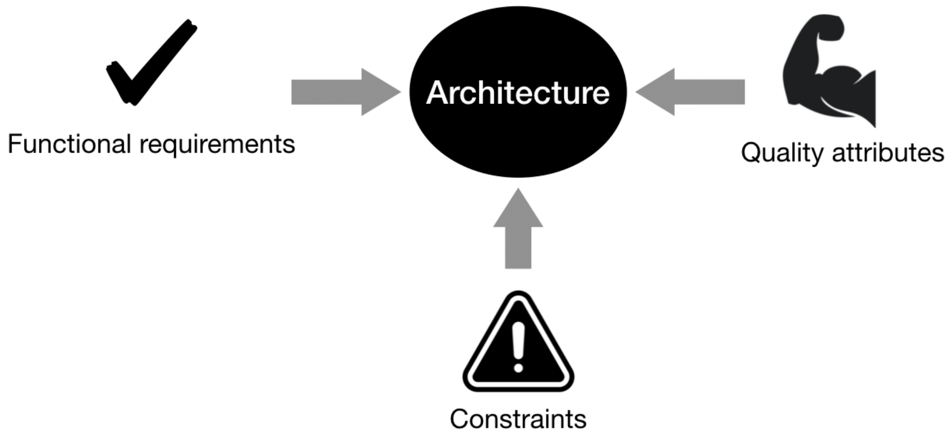

Figure 1: Architecture drivers that influence design.

3.1 Functional Requirements

Functional requirements are the least requirements that out system has to do. Nozama company, the customer, has outlined high level description of functional requirements in the project description paper. Based on the description on the paper, we came up with an extended list of the requirements as below.

- Accept online customer orders with web server with a designated URL.
- Store the orders in the order DB with customer name, items of purchase, address, and pending marker.
- Implement a scheduling algorithm that maximizes throughput, minimizes backlog, and guarantees safety.
- The scheduling algorithm need to handle partial orders; Some orders with large number of items cannot be handled in a single delivery.
- Send the next destination, according to the scheduler, to the robot.

- Signal the robot to start a delivery.
- Robot autonomously drives to the destination without crashing into obstacles. It could be line tracing or any other method using computer vision.
- Stream a live video, captured with robot camera, to a remote host in order to monitor robot movement.
- Remotely control robot servo with the continuous keyboard input.
- Monitor robot status such as malfunctioning of any parts.
- Manually put the robot into maintenance mode when abnormal behaviors are detected. The robot may go into maintenance mode by itself.
- Allow the team members to intervene and fix the part that causes maintenance mode.
- Once the problem is solved, reset the robot into delivery mode. The robot should be able to get back onto where it left.
- Signal workers to load/unload specific items from the robots at designated places and get signal back from them with a UI.
- Return the robot to the loading dock when it completes a delivery.
- Monitor the inventory status at loading dock on a regular basis. The number of items left in the dock must be identified when requested.

After having a couple of meeting with the customer and stand-up evaluation, we were able to identify the customer's need in a more fine-grained fashion. We provide several use case diagrams to describe what our system has to do in various situations.

3.1.1 Use Case: Autonomous Delivery

Client places an order on the web. The server checks that the order information is valid. If the order is valid, put the order from the server into the database and save the order. While managing the database, operator schedule the order of delivery, and send the schedule to the worker and the robot. The Worker loads boxes on the robot according to the scheduling result. The robot is loaded with boxes and delivers them to the corresponding address through the designated path. If the robot recognizes the address sign and it is on schedule, it stops. The loader unloads the orders to the address and gives a signal to robot. The robot returns to the loading dock.

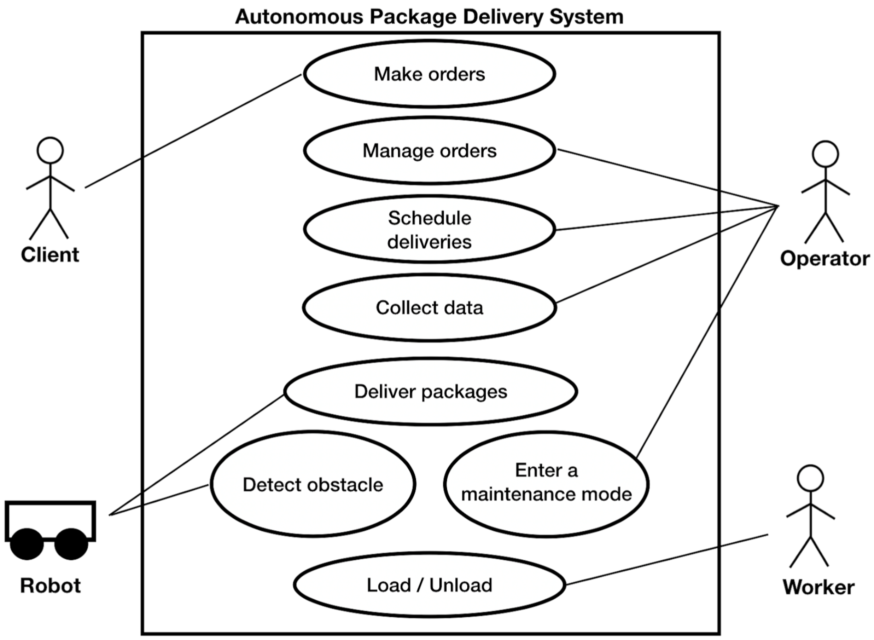

Figure 2: Epic usecase diagram on autonomous package delivery system.

3.2 Quality Attributes

Quality attributes are the characteristics of our system that we would like to stress. These are the strong points of our system that is not necessarily listed in functional requirements. Our team has identified 7 quality attributes that need to be reflected in the system design. We provide a table summarizing the attributes and their priorities and a detailed scenarios for each attribute.

ID	Quality Attributes	Description	Priority
1	Safety	Avoid physical accidents. (Crash to signs, obstacles, etc.)	High
2	Efficiency	Deliver as many items as possible in the given amount of time.	High
3	Reliability	System always works stably and can recover when in trouble.	High
4	Accuracy	Deliver specific items in designated address.	Middle
5	Customer Satisfaction	Get delivered within expectations in a timely manner.	Middle
6	Security	Minimize the risk of exposing customer's private information.	Low
7	Modifiability	Make code maintenance simpler.	Low

Table 2: List of Quality Attributes

3.2.1 Safety

Our robot can avoid physical accidents. (Crash to obstacle, signs, etc.)
1. Stimulus: Robot detects the obstacle on the way.
2. Source of stimulus: Obstacle.
3. Environment: The location of robot delivering packages and its surrounding. The waiting time of orders.
4. Artifact stimulated: Robot.
5. Response: The robot stops without hitting an obstacle or makes a detour.
6. Response measure: Number of time that the robot hits obstacles.

3.2.2 Efficiency

Our system delivers as many items as possible in the given amount of time.
1. Stimulus: Delivery orders received on our server.
2. Source of stimulus: Customers.
3. Environment: Scheduling algorithm have knowledge of distance from each house to loading dock.
4. Artifact stimulated: Scheduling system, the path the robot will take.
5. Response: Prioritize orders from far address and bundle up orders from the same address. Remaining inventory fills up with partial orders from the next far address.
6. Response measure: Total number of delivery completed in a given time, delivery

3.2.3 Reliability

Our system always work stably and can recover when in trouble.
1. Stimulus: Battery issue happens. For example, remaining battery is under the threshold and robot stops.
2. Source of stimulus: Robot's battery.
3. Environment: Robot in operation, current location, robot's inventory, the waiting time of orders.
4. Artifact stimulated: Running out robot battery.
5. Response: Check the remaining orders and their waiting time and return to the loading zone. Alternatively, the robot can go into maintenance mode and call for manual intervention.
6. Response measure: The number of times the robot stops running out battery.

3.2.4 Accuracy

Our robot delivers specific items in the designated address.
1. Stimulus: An order that is being fulfilled.
2. Source of stimulus: Customer or the scheduler.
3. Environment: The camera of robot, the signage in destination.
4. Artifact stimulated:
5. Response: Extra check on robot to verify it is on the right destination and has the right item.
6. Response measure: Number of correct deliveries / number of total deliveries (%).

3.2.5 Customer Satisfaction

Our system gets delivered within expectations in a timely manner.
1. Stimulus: Waiting time of a pending order exceeds some threshold.
2. Source of stimulus: Pending orders.
3. Environment: All the pending orders including the ones that are selected by scheduling system and the ones that are not. Scheduling system. The location and the room left in robot inventory.
4. Artifact stimulated: Scheduling system.
5. Response: Prioritize the orders whose waiting time exceeded the threshold.
6. Response measure: Longest time taken for any order to be fullfilled.

3.2.6 Security

Our delivery system minimizes the risk of exposing customer's private information.
1. Stimulus: An attempt to hack into robot or arbitrary connection errors.
2. Source of stimulus: Hackers and other people on the same e or external network.
3. Environment: The robot on network, the server.
4. Artifact stimulated: An unexpected connection into robot detected.
5. Response: Prevent unauthorized connection with secure network protocol, and send minimal information to the robot. For example, the robot only knows the next destination, without access into full delivery schedule.
6. Response measure: The number of leaked customer information

3.2.7 Modifiability

Our code is separated into many independent pieces, make code maintenance simpler.
1. Stimulus: Some updates on codes.
2. Source of stimulus: Members of NOZAMA-HOTMETAL.
3. Environment: Current codes to be modified and new codes to be added.
4. Artifact stimulated: Some bugs in software occurred or there are some updates on codes.
5. Response: Well-modularized codes.
6. Response measure: Updates codes without looking into internal structures of autonomous robot delivery system.

3.3 Constraints

For a realistic implementation of the system, we define various constraints that our project is under. These constraints will be criteria we could use when making design decision. Some constraints are set by Nozama company according to their business needs. Other constraints are related to technology that we are allowed to use.

3.3.1 Business Constraints

This project is a competion among 7 teams to win the project.Nozama company will evaluate the teams on the same measure, and there will be one final team who sign the contract. Since this is a competition, we need to focus on increasing the performance as much as possible.

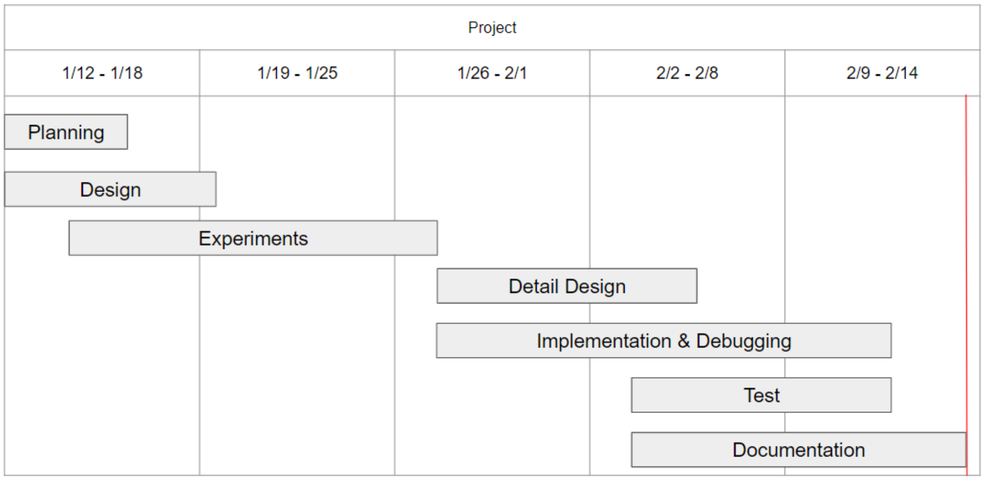

Figure 3: Project timeline

The final evaluation date is Feb 14th. There will be a number of standup evaluation from Nozama company to check if the teams are making progress. We scheduled our project accordingly, so we can showcase our technology at the right time. Also, effort should be put into how we should present our technology.

All works have to be done by our team. No additional workforce is allowed other than the team members. However, a mentor will be designate to our team. We should consult our decision and implementation decision with him on a regular basis.

3.3.2 Technical Constraints

Delivery robot is a Raspberri-pi equipped with camera, servo, and inventory. No hardware tuning is allowed in this project. Below is the technical specification of the robot provided by the project organizers.

We have four additional laptops to utilize in the project. Laptops are Macbook pro or others with equivalent computation power. These machines can be utilized in order to provide computation power, client side, or server side.

The robot will be delivering packages on the provided map that has signage chosen by the signage committee. The robot will in tested in a designated location. This means that we have access to the environment that we are going to be tested on, during development.

4 Architectural Structures

In this section, we show the actual structure of our system that resulted from the architectural drivers described above. This is an abstract description in that it will omit some details that are not relevant in revealing architectural drivers. However, it is concrete enough to to include the general structure that will help implementation. We view our system in three different perspectives: Physical perspective, Static Perspective, and Dynamic perspective.

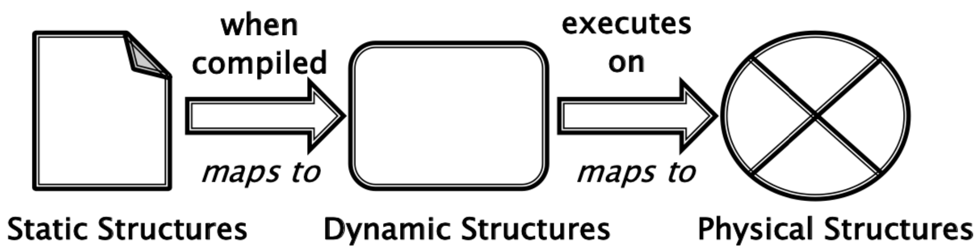

Figure 4: Several views on structures. Image borrowed from Tony's slides.

4.1 Physical Perspective

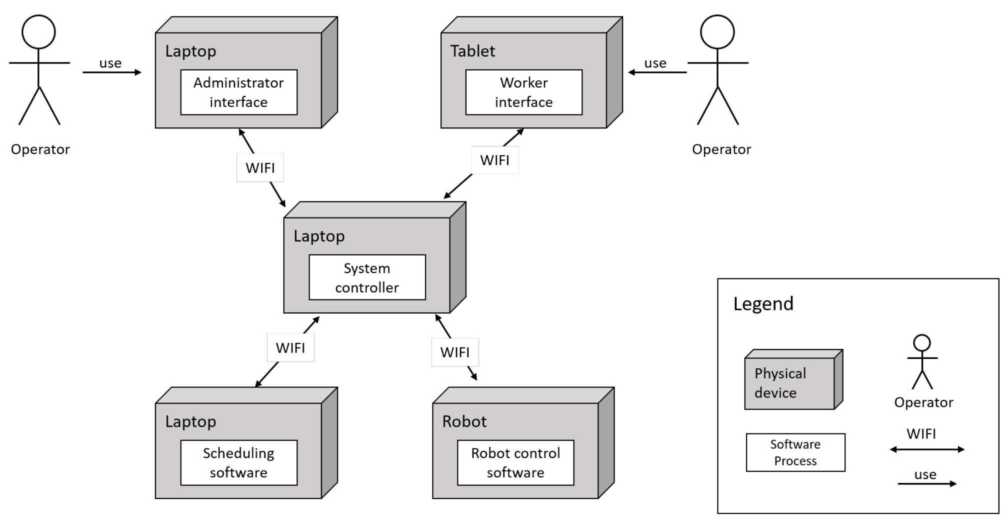

Figure 5: Physical view of the system

4.2 Static Perspective

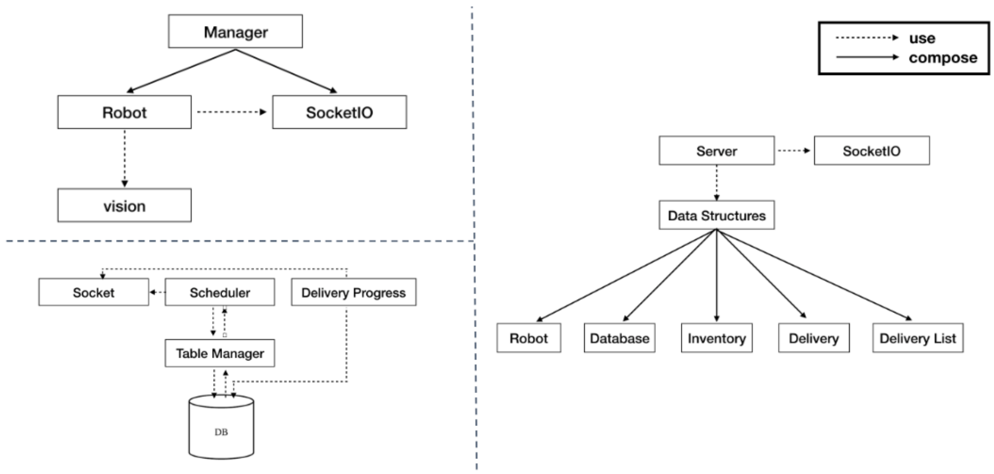

Figure 6: Static view of the system

4.3 Dynamic Perspective

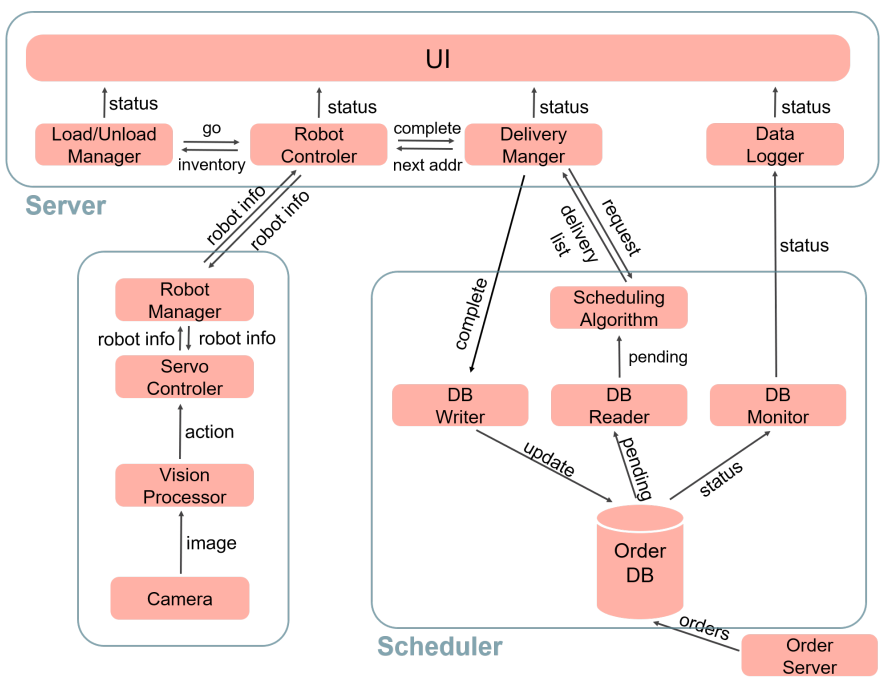

Figure 7: Dynamic view of the system

5 Key Parts of the System

Although all parts of the system are important since our system run as a result of collaboration of different parts, there are some key components that are more closely related to the performance. (In this project, the performance was measured in terms of security, throughput, and backlog) We identify the key parts so as to prioritize the tasks.

Key parts of our autonomous package delivery system is the efficient scheduling algorithm. As we analyzed, the time takes to unload from one address is relatively long, so we think that it will be advantageous to arrange as many orders from one address as possible. Additionally, we plan to deliver to the furthest address first with loading the maximum number of items. The remaining items will be delivered to the other addresses depending on the various situations. Since, the robot will move the minimum distance and always loads the maximum items, our algorithm should show the high throughput.

6 Other concerns

This is the questions we raised during the project, that are not necessarily related to architectural design.

6.1 Technical Difficulties

- What is the distribution the orders are generated from? Should we consider random distributions?

6.2 Project Management

- How to communicate among team members? To what detail will we explain what each one has done? How should we engage with others' responsibilities?