Mihai Sirbu

Assignment 2- Comp 3004

February 18th, 2024

As the documents are plentiful and I doubt you wish to read a lot of extremely *interesting* design choices, I will keep this document to the essentials.

General overview

Centralized System

In a centralized elevator system, the mediator pattern is applied to simplify the relations between the components belonging to different floors of a building, including the floors, the elevators, and the buttons. Essentially, the ElevatorControlSystem class fulfills the role of the mediator. The ElevatorControlSystem is the manager for elevator requests, which provides direction of elevators to go to floors and the optimization of the overall movement of elevators for an optimal operation. This centralization reduces the complexity of having multiple systems communicating with each other: which can bring a system that is very tightly coupled, and hard to maintain and extend. In addition, the use of a mediator results in better scalability, since one may add new elevators or floors to the system without much change in the existing code.

The Distributed System

The design of the distributed elevator system changed from a central mediator to a coordinator model. The ElevatorCoordinator is an enabler and not a regulator; it enables the information about the status and intentions of each of the elevators to be reported. The design decision is made with the intention that the system should be more autonomous, so that each elevator is given the ability to make its decisions on the basis of the shared information by itself. The coordinator assists in this process by keeping a registry of elevators and by broadcasting requests to all elevators in the system. The elevators then use this information, with some internal algorithms(not implemented yet), to decide upon the best action they can take to handle that request. This way, since any failure is not on a single point, the system can achieve higher levels of redundancy and flexibility with the decentralized approach it takes. In this sense, the distributed nature of this architecture can also make the operation of elevators more effective in complex scenarios, such as each time they make decisions based on real-time information and negotiate with one another as to which request to handle in the most efficient way.

Sequence Diagrams

There are 7 sequence diagrams in total. 2 Success diagrams, and 5 safety feature diagrams. Each success diagram can be broken down into each passenger/elevator combination(and thus 3 individual pictures), effectively using all 3 passengers and elevators for the sequence diagrams. Below are the different scenarios encompassed by the success diagrams:

Success 1:
Situation 1: Passenger 1 travels from floor 1 to floor 2.
Situation 2: Passenger 2 travels from floor 2 to floor 3.
Situation 3: Passenger 3 travels from floor 5 to floor 4.

All these 3 situations happen **simultaneously.**

Success 2:

Situation 1: Passenger 1 travels from floor 7 to floor 1.

Situation 2: Passenger 2 travels from floor 1 to floor 7.

Situation 3: Passenger 3 travels from floor 2 to floor 3.

All these 3 situations happen **simultaneously.**

Class Diagrams

Class diagrams are relatively simple and straightforward. Their functions can be seen in the header files (header folder) for further understanding. I must concede the fact that I do not have a clear picture of the actual code that will be written due to these classes, as during production it is highly likely that I will change the function names, classes, attributes, etc. I wrote these as a general guidelines and not an extensive diagram. To put it simply, the main difference between the centralized vs distributed systems(as mentioned in the first chapter) is the fact that the centralized system uses the mediator pattern, and thus all classes are related to the ECS class. In the distributed system, the ElevatorCoordinator class does not **control** the elevators, but the elevators use the ElevatorCoordinator class to make their judgement.