Elevator Control Systems: A Case-Study

Prof. Juri Vain

1 Problem Formulation

The problem is modelling the elevator control systems with their operational and functional requirements/specifications. Then we use this as the framework for understanding the concepts introduced in the formal methods course. Considering simplicity, we restrict our analysis to 4 floors. The students are free to experiment with the implementation, but the mentioned requirements need to be verified.

2 System Overview

The elevator control systems has the following components:

S.No.	Component	Functionality
1	Floor Panel	Panel at each floor
2	Floor display	Display of current floor and direction
3	Floor Door	Control unit to open and close doors at each floor
4	Car Panel	Panel in each car with request controls
5	Car Display	Display of each floor and direction
6	Motor Controller	Control Unit for the motor for each door
7	Elevator Manager	The main controller
8	Carriage	Unit carrying the people active
9	User Interactions	External User Interactions active
10	Service Queue	to store the value of the floor queue with operations
11	Illumination Control	Control the illumination in the floor and car active

These different components have to be designed in UPPAAL as UTA models. The timing constraints will be updated as the course progresses. In our control systems, the *Elevator Manager* orchestrates the different operational and functional components except for the user interactions.

Examples of components to be modelled include:

1. Call-button on 0, floor
2. Call-button on 1, floor
3. Call-button on 2, floor
4. Button for 0, floor
5. Button for 1, floor in the elevator
6. Button for 2, floor in the elevator
7. Lift on 0, floor
8. Lift on 1, floor
9. Lift on 2, floor
10. Door is closed on 0, floor
11. Door is closed on 1, floor
12. Door is closed on 2, floor
13. Moving, upwards
14. Moving, downwards
15. Open the door on 0, floor
16. Open the door on 1, floor
17. Open the door on 2, floor

3 Requirements

The requirements of the control systems are captured as logical operations and they are presented here: (not exhaustive though- will be improved during the course)

1. The *Elevator Manager* has the following elevator states {*inService*, *floorStop*, *noOperations*, *inMaintenance*}.

2. Floor panel should have the following *up* and *down* button is depressed by the user, the up or down LED should glow. They can both glow together and the complimentary operations should be cut-out on reaching the floor.

3. When a *carCall* or *flCall* is requested it should be registered in the service queue.

4. Impose priority to call requests.

5. In the *carPanel* the *flNumber* should be displayed.

6. For servicing any floor control, the elevator door needs to be closed, the floor button is depressed, no overload or any other stopping operations initiated. Once the door opens the status of the lift should be brought to floor-stop.

7. When the call is received, assign a direction (forward or reverse) and service the call in the queue.

8. Update the car-display on the floor when a button is depressed.

9. Based on the assigned direction show the *Up* and *Down* display in the *carDisplay* and *floorDisplay.*

10. Door closing is achieved after this either by depressing the door close button or when the user selects the floor (a timing will be included later here) and there are no warnings of users coming in from a motion detector sensor.

11. The lift should not operate if the user selects the same floor.

12. Execute the task queue and provide the service to the floor.

13. The motor control should latch the *Direction* and *Enable* the operation.

14. The sensor inputs should cut out the queue when the elevator reaches the floor and load the queue with a new value. It should service the requests coming in descending and ascending order while moving up and down, i.e., if there is a request from floor 2 after floor 3, still floor 2 service request can be handled while going up.

15. The above task should be repeated for *Car Control* command as well.

16. *pickUp* and *dropOff* operations, de-register the current floor from the queue, set the operating mode to be *Idle* and *doorOpen* status to be 1.

17. During dropping and picking operations keep the elevator stopped.

18. Update the elevator status after this operation and continue with the service queue.

3.1 Messages

The typical messages that will be used by the different UPTA models are given here. Again this is not exhaustive will be filled as we go on.

S.No.	Messages	Comment
1	Floor request	A request made in the floor
2	Car request	A car request is similar to the floor request
3	Open door	Open door signal is initiated by the user or motor
4	Close Door	Close door signal is initiated by the user or motor
5	Car-go-to	Is a request placed by the main control
6	Car position	Indicate the current car position
7	Car Arrived	Indicate that car has arrived in a floor
8	Car requested granted	Self-explanatory
9	Floor requested granted	Self-explanatory