ECE 153A/253, CS 153A - Homework 4

Consider a simplified finite state machine controlling an elevator for a 5 story building such as HFH. The details of the model and functionality of the FSM are as follows:

- The elevator is a simple one with 1 call button on each floor and 5 in the cab. The behavior of the elevator is identical to a button call from either inside or outside.
- On arrival at each selected floor, the elevator pauses for 10 seconds to allow people to get on and off before proceeding to its next destination.
- It moves from floor to floor at 5 seconds per floor unless it is stopped on a floor.
- Buttons are active until the elevator arrives on a given floor to clear it. Any combination of buttons is possible.
- The elevator updates the list of destination floors when it passes each floor or if it is stopped. Thus, it won't respond to a call request to a floor closer than 1 stop away unless it is currently stopped.
- The elevator must service all call buttons in a given direction before the elevator is allowed to change direction. This means that if the elevator is currently moving up, it must stop at any requested floors that are above the current floor before the elevator is allowed to stop at any floors below the current floor.
- If request for a floor is already pending, subsequent calls to that floor can be ignored.

A QP-Nano based FSM that models this behavior is provided to you. The model is instrumented to keep track of the cumulative time taken to service calls and the total number of calls made to each floor. This information will be helpful for gathering statistics about the time to service call requests for different call frequencies.

1. Read the FSM model code and answer the following questions:
   (a) Draw a diagram showing the states of the FSM and the hierarchy of states. (You do not need to put in all the transitions).
   (b) List all the input signals and state variables of the FSM.
   (c) Describe what Q_TRAN(), Q_SUPER() and Q_HANDLED() are used for.
   (d) Describe the actions taken when the elevator is stopped and a call for any floor is made.
   (e) Describe the actions taken on "TICK" when the elevator is stopped. Under what conditions would the elevator switch to the "moving" state?
   (f) Describe the actions taken on "TICK" when the elevator is moving. Under what conditions would the elevator switch to the "stopped" state?

2. The code currently keeps track of the time from when a call is first made for a given floor and when the elevator arrives at that floor. Run the code and determine the average time between call and arrival for calls to each floor, assuming that a random floor is called at a constant rate R (time per call). Do this for the following 5 cases: R = 200, 100, 50, 20, or 10 seconds per call. You have an executable code model – just run it for long enough that you have a sound statistical sample for the average service time on each floor accurate to 1 second with 95% confidence.

3. Instrument the code to model the elevator door: the stop time for the elevator is now variable based on the number of people entering or exiting the elevator. At random, between 1 and 10 people can enter/exit the elevator. Each entering/exiting person adds 1 second to a fixed minimum stop time of 5 seconds. Determine the new average service times for each floor for call frequencies of 200, 100, 50, 20, or 10 seconds.

4. Instrument the code to model an emergency key: if the elevator is already stopped, it proceeds to the ground (floor 1) as if called there and ignores all other calls after emergency is activated; if the elevator is moving, it continues moving in the current direction and stops at the closest floor as if called to that floor, does not open the door, then proceeds to floor 1 until the emergency has ended. In both cases, it clears the pending calls when emergency is set. Simulate the elevator for 500 seconds at a call rate of 20 secs per call to randomize the elevator state, then determine the average time to service an emergency call (i.e., time from emergency start to arrival at floor 1). To get adequate statistics, let the emergency run for long enough to get the worst case response time, then deactivate emergency for another 500 seconds and repeat.

5. The original code for this assignment has several of the issues discussed in the lecture as indications of poorly shaped hierarchy. For example, several signals are handled in multiple places with identical or nearly identical actions. What other indicators do you note in the code? Describe briefly how you'd restructure the code (don't re-write it!) to remove these redundancies.