Recitation #9

18-649 Embedded System Engineering

Friday 10/30/2015

Electrical & Computer ENGINEERING

Note: Course slides shamelessly stolen from lecture

Carnegie Mellon

Announcements and Administrative Stuff

- Project 9 is due Thursday Oct. 29st by 10pm.

- Hand in ALL the files needed to run your tests.

Project 9

- **Pick up where you left off on Project 8**

- **Finish designing fast speed drive and smart dispatcher**
  - Statecharts
  - Unit tests
  - Implementation
  - Traceability
  - Peer reviews:
    - Dispatcher & DriveControl statechart
    - Dispatcher & DriveControl implementation
    - Revised unit tests

New Requirements

- **R-T6: The Car shall only stop at Floors for which there are pending calls.**
- **R-T7: The Car shall only open Doors at Hallways for which there are pending calls.**
- **R-T8: The Car Lanterns shall be use in a way that does not confuse passengers.**
  - **R-T8.1:** If any door is open at a hallway and there are any pending calls at any other floor(s), a Car Lantern shall turn on.
  - **R-T8.2:** If one of the car lanterns is lit, the direction indicated shall not change while the doors are open.
  - **R-T8.3:** If one of the car lanterns is lit, the car shall service any calls in that direction first.
- **R-T9: The Drive shall be commanded to fast speed to the maximum degree practicable.**
- **R-T10: For each stop at a floor, at least one door reversal shall have occurred before the doors are commanded to nudge**

Fast Drive Speed

- **Simulator assumes that car can instantly stop from slow speed**
- **Need to ramp down speed from fast in time to stop at desired floor**
  - Cannot instantly stop from fast speed (engages emergency brake)
- **Commit Point:**
  The elevator position at which you must decide whether to stop at particular floor
  - Occurs when elevator reaches the stopping distance from that floor location
  - Think of it as a "point of no return"

Fast Speed Drive - Commit Point

- **Stop speed = 0.00 m/s**
- **Slow speed = 0.25 m/s**
- **Fast speed = 1.00 m/s**
- **Constant acceleration/deceleration = 1.00 m/s^2**

- **Calculate the maximum stopping distance of the elevator**
  - $x(t) = x_0 + v_0*t + ½ * a * t^2$
  - $v_f^2 - v_0^2 = 2*a*\Delta x$

- **Include slack for:**
  - Sensor granularity (CarLevelPosition is in 10 cm increments)
  - Delay of DriveControl control loop
  - Delay for message to be sent periodically
  - Be conservative!!
    - Leveling behavior may save you, but better not to overshoot in a real elevator

Fast Speed Drive – Verification Example

- **Commit point computation:**
  - Ideal case: kinematics equations
  - Real-world: kinematics + delays

- **Suggestion: use the monitoring infrastructure to verify commit point calculations**

- **What conditions would you check?**

- **What sensor inputs would you need?**

Only Service Landings with Pending Calls

- **Elevator must only stop at floors/hallways that need to be serviced**

- **DesiredFloor**
  - Floor – the floor we intend to go to next
  - Direction – the direction we intend to go **after** we reach the desired Floor
  - Hallway – which doors should open

Only Service Landings with Pending Calls

- **Update desired floor/direction based on current state of hall/car calls**
  - When is it OK to update these?

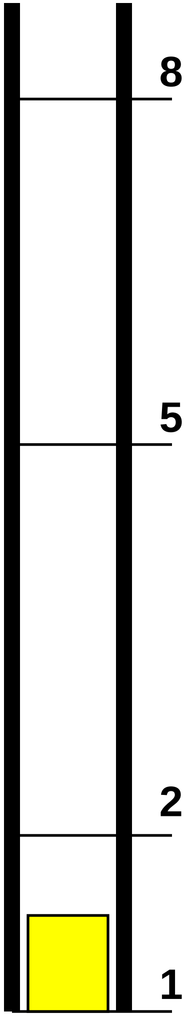
- **For example:**
  - If the elevator is stopped and opening its doors
    AND there is no pending call at the current floor
    AND there is a pending call at another floor
    THEN:
    - DesiredFloor.Floor must NOT BE current floor by the time the doors are fully open
    - DesiredFloor.Direction must correspond to illuminated lantern direction

- **What about between floors?**

- **When should you NOT update these values?**

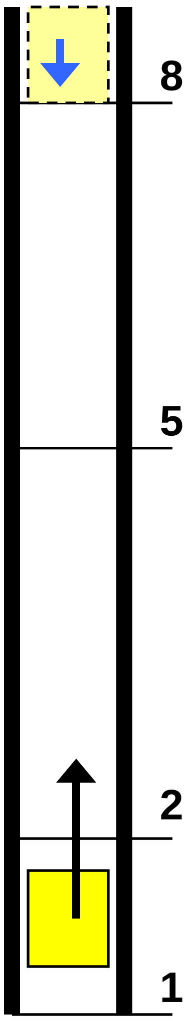
- **Above example is not a hard requirement**
- **Follow the requirements and do what makes sense for your design**

Example



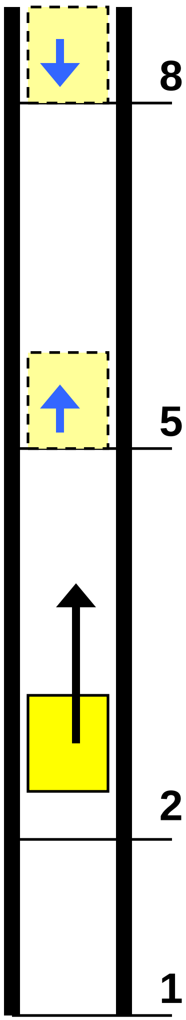
- **Suppose car is initally at floor 1 and stopped**
  - No calls
  - Desired Floor = (1, stop)

Example



- **Get a hall call for (8, down)**
  - Car begins moving up
    - Current direction = Up
  - DesiredFloor.floor = 8
  - DesiredFloor.direction = Down
    - Where we're going after servicing floor 8

Example



- **Get a hall call for (8, down)**
- **Then receive a hall call for (5, up)**
  - Dispatcher decides to service floor 5 first
    - Depends on your algorithm
  - Current direction remains Up
  - DesiredFloor.floor = 5
  - DesiredFloor.direction = Up
    - Where we're going after we service floor 5
- **How do you decide where to go next?**
  - Based on current set of car/hall calls
  - Anything that meets the requirements is OK
    - Example: Sweeping up and down servicing calls in the current direction first

Questions?