18649 - Behavioral Requirements II

Distributed Controller Requirements

18649 <semester>
Group x - Student names and andrewIds

Introduction

This document describes the complex sensor, actuator, and environmental objects in the system.

Table of Contents

- Passenger (environmental)
- Safety Sensor (environmental)
- Drive (environmental)
- DoorMotor[b,r] (environmental)
- DoorControl[b,r]
- DriveControl
- LanternControl[d]
- HallButtonControl[f,b,d]
- CarButtonControl[f,b]
- CarPositionControl
- Dispatcher

1. Passenger[p] (environmental object)

Replication:

N passengers per system; N is unlimited subject to being less than steady-state carrying capacity of elevator.

Instantiation:

Passengers arrive at the elevator landings according to the specifications of the passenger input files. Passengers may also begin the simulation already in the car.

Assumptions:

Passengers do not "fight" each other for position in entry and exit queues.

Passengers weigh 150 pounds each.

Input Interface:

- DoorPosition[b,r]
- CarLantern[d]
- CarLight[f,b]
- CarPositionIndicator
- HallLight[f,b,d]
- CarWeightAlarm

Output Interface:

- CarCall[f,b]
- DoorReversal[b,r]
- HallCall[f,b,d]

- CarWeight

Constants:

These constant values are the same for all passengers:

- DoorCheckPeriod - how often the passenger checks the doors to see if they can enter or exit - 100ms.
- CallCheckPeriod - how often the passenger checks the hall or call button to make sure it is still lit - 200ms.
- CallRecheckPeriod - how long the passenger waits to check the hall or call button after pressing the button - 500ms.
- Backoffperiod - how long the passenger waits after the doors close when the car weight alarm has sounded - 10s.
- IgnoreLeveling - Whether or not the passenger ignores the leveling requirements - depends on presence of "il" flag.

Parameters:

Passenger behavior is parameterized by the following values. These values are defined per-passenger and may differ from passenger to passenger. These values are constant during the passenger's interaction with the elevator.

- These constants are defined in the passenger file:
  - arrivalTime: the time when the passenger arrives at the landing and begins interacting with the system
  - startFloor: the starting floor for the passenger. If this value is 0, the passenger begins the simulation in the car. Passengers that are injected after time 0s may not start in the car.
  - startHallway]: the starting hallway for the passenger, "front" or "back".
  - destFloor: the destination floor for the passenger.
  - destHallway: the destination hallway for the passenger, "front" or "back".
- These constants are computed from other parameters or chosen randomly at initialization. They do not change during the simulation.
  - travelDirection: the travel direction for the passenger, which is computed based on the relative locations of the startHallway and destHallway.
  - hallCallDirection: "up" or "down", indicates the hall call button the Passenger will press at the startFloor/startHallway.
    - In most cases, this value is the same as the travelDirection.
    - If startFloor==destFloor (the passenger enters and exits at the same floor), then the value is chosen randomly.
    - if startFloor==destFloor==1, then the value is always "up".
    - startFloor==destFloor==MaxFloor, then the value is always "down".
  - width: the minimum acceptable total opening distance of the doors, in percent of total opening width. The value is random in the range [20, 45].
  - hallPressTime: the amount of time the passenger will hold the hall button down. The value is random in the range [250ms, 1000ms].
  - carPressTime: the amount of time the passenger will hold the car button down. The value is random in the range [250ms, 1000ms].
  - doorTraversalDelay: the amount of time the passenger takes to enter or exit the car. The value is random in the range [1000ms, 2000ms].
  - doorBackoutDelay: the amount of time the passenger takes to abort entry if they are unable to enter or exit the car. The value is random in the range [2000ms, 4000ms].
  - missedOpeningThreshold: the number of times the elevator can come to a floor servicing the other direction before the passenger begins to ignore the car lanterns and enter the car regardless. The value is random in the range [3,5].

State:

These state values are maintained by the passenger during the interaction with the elevator:

- State - represents the passengers current state. One of:
  - INIT - created, but not injected into the system yet
  - WAITING_IN_HALL - waiting at the arrival landing

    - ENTERING - attempting to enter the car (only on passenger at a time should be in this state)
    - ENTER_BACKOUT - aborting entry because the doors closed
    - WAITING_IN_CAR - waiting in the car before arriving at destination floor.
    - EXITING - attempting to exit the car at the destination landing
    - EXIT_BACKOUT - aborting exit because the doors closed
    - OVERWEIGHT_EXITING - exiting the car because the car is overweight
    - OVERWEIGHT_BACKOFF - waiting in the hallway because the car is overwieght
    - DONE - delivered to destination successfully.
- hallPressCount - the number of times the passenger has had to press the hall call. This count may be reset in some cases (e.g. due to an overweight exit)
- carPressCount - the number of times the passenger has had to press the car call
- missedOpenings - the number of times the car has come to the floor servicing the other direction (see constant missedOpeningThreshold).
- abortCount - how many times the passenger has failed to traverse the door [illegible]le entering or exiting.

Passenger Metrics:

There are two metrics of interest for passengers: delivery time and satisfaction.

Delivery time is measured from the time the passenger arrives at the landing to the time that the passenger exits the car at their destination. Lower delivery times are better.

Safisfaction is a measure of the undesirable behavior that the passenger observes during their interaction with the elevator. Undesirable behaviors are behaviors that deviate from normal elevator behavior, but do not (in general) affect the eventual delivery of passengers. The satisfaction score begins at 100 (a perfect score). Each time the passenger observes an undesirable behavior, the passengers satisfaction is reduced by multiplying the current score by a factor that depends on the severity of the behavior observed. A higher satisfaction score is better, with 100 being a perfect score. The specific deductions for undesirable behavior are listed below, with the factor by which the satisfaction score is reduced.

- Wrong motion - 0.5 - The passenger enters the car when an up or down car lantern [illegible]sistent with their direction is displayed, but the car moves in the opposite direction. If the passenger enters the car when both lanterns are off, then the car may move in either direction without incurring this deduction.
- Skipped destination - 0.5 - The passenger was in the car, but the car passed the passenger's destination floor without stopping.
- missedOpeningThreshold exceeded - 0.5 - While the passenger was waiting in the hall, the car arrived at the floor missedOpeningThreshold times, but the lanterns were never consistent with the passengers travelDirection.
- Failed to enter/exit car - 0.8 - This deduction occurs every time the passenger attempts to enter or exit the car, but the doors are not open wide enough after the doorTraversalDelay.
- Car lantern direction change - 0.8 - This deduction occurs every time the passenger exits the car because the lanterns changed to be inconsistent with the passengers travelDirection.
- Repeated call press - 0.9 - This deduction occurs if the passenger must press the hall call or car call more than once because the light goes out without servicing the passenger.

Note that sometimes, especially in high traffic scenarios, it may be impossible to avoid having some deductions for a small number of passengers.

Constraints:

1.1: Passengers are prevented from entering or exiting the Car whenever the appropriate doors are not open far enough.

1.2: If Passenger p is outside the car and the CarWeightAlarm is triggered, the passenger shall not try to enter the car or press a hall call for 10 seconds after the doors fully close.

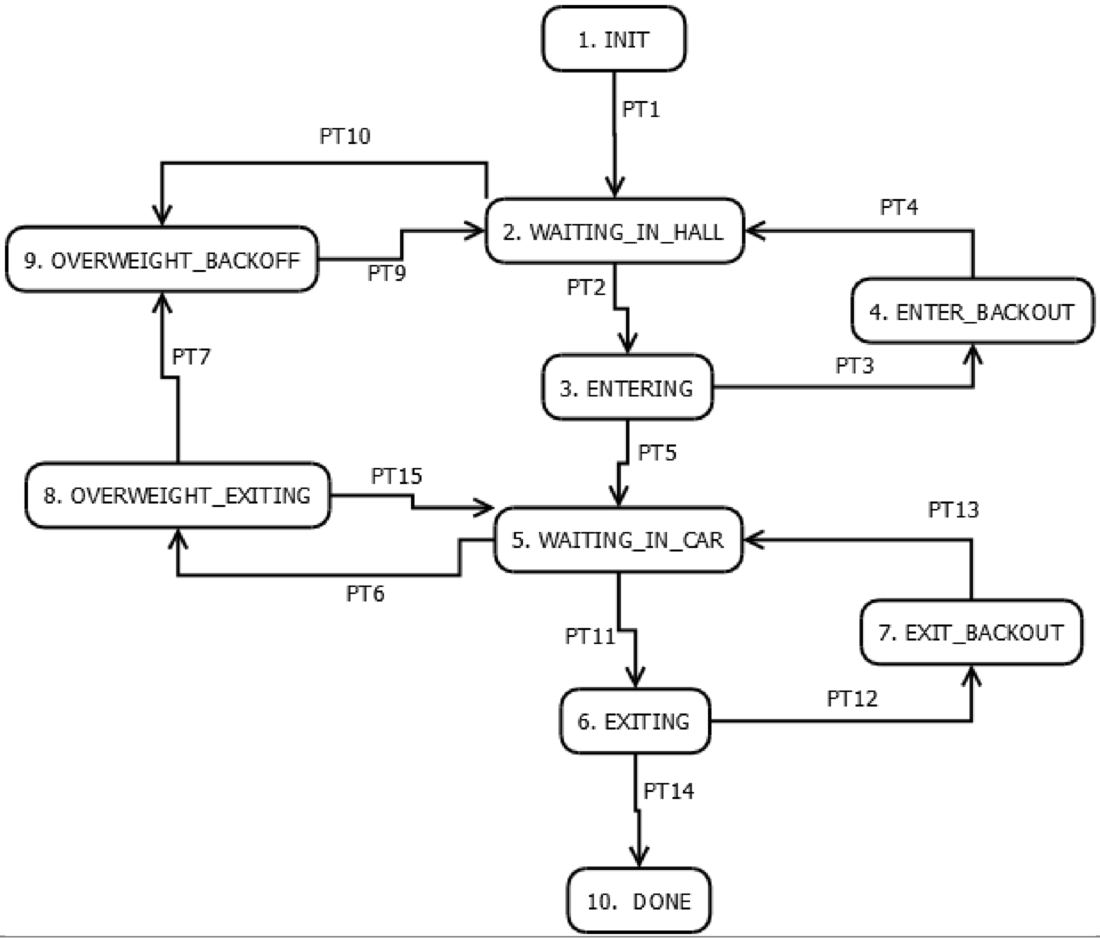
Behaviors:

Because passengers are inherently event-triggered, their behaviors differ from those of the other system objects.

1.3. When the passenger arrives at the start landing, if the hall call light it is not lit, the passenger shall press the hall call button in the hallCallDirection.
1.4. While waiting for the car (while the doors are closed at the landing), the passenger shall monitor the hall call button in hallCallDirection and shall press it again if the hall call light is extinguished.
1.5. When the passenger is waiting at the landing and the doors at the start landing begin to open, the passenger shall try to enter the car if:
  1.5.1 The leveling error at the floor is less than or equal to 5mm OR IgnoreLeveling is set to true AND
  1.5.2 The car lantern is consistent with the passenger's direction of travel OR missedOpenings exceeds missedOpeningThreshold (If neither car lantern is lit, the passenger will always consider the direction consistent.) AND
  1.5.3 The doors are open wider than the width constant.
1.6 After the passenger has entered the car, if the direction arrow becomes inconsistent with his travelDirection and the doors are still open, the passenger shall exit the car at the start landing.
1.7. The passengers shall traverse the door one at a time. Passenger priority shall be determined by the following guidelines:
  1.7.1 Passengers exiting the car shall take priority over passengers entering the car.
  1.7.2 Passengers in the car shall exit on a first-come-first-serve basis.
  1.7.3 Passengers at a landing shall enter on a first-come-first-serve basis.
  1.7.4 If a passenger fails to enter or exit the car, the passenger shall return to the end of the appropriate car or hall queue.
1.8 The passenger shall block the door for doorTraversalDelay while entering or exiting the car.
  1.8.1 After doorTraversalDelay has expired, if the door is not open wide enough for the passenger to traverse the door, the passenger shall block the door for an additional doorBackoutDelay while returning the passenger's previous location (hall landing or car).
1.9 When the passenger enters the car, if the car call light it is not lit, the passenger shall press the car call button for his destFloor.
1.10. While waiting in the car (while the doors are closed), the passenger shall monitor the car call button and shall press it again if the car call light is extinguished.
1.11 If the OverweightAlarm sounds, the last passenger to enter the car shall attempt to exit.
1.12 If the OverweightAlarm sounds and the doors are at least partially open at a landing, the passengers waiting at that landing shall wait a minimum of 10 seconds after the doors fully closed before they begin monitoring the hall call buttons again.
1.13 When the passenger is in the car and doors begin to open, the passenger shall attempt to exit if:
  1.13.1 The leveling error at the floor is less than or equal to 5mm OR IgnoreLeveling is set to true AND
  1.13.2 The car position indicator indicates destFloor AND
  1.13.3 The doors on the destHallway side are open wider than the width constant.
1.14 If the startFloor and destFloor are the same, after entering the car, the passenger shall immediately attempt to exit the car at the destHallway.

State Chart:

This state diagram and the accompanying tables describes the trigger conditions that can cause the passenger to move from one state to another, and the activities that accompany each state and transition. Passengers are inherently event triggered, so their interaction includes actions on transitions. You should NOT copy the passenger behaviors as a model for your elevator controllers. The elevator controllers (DriveControl, DoorControl, etc) may not have actions on transitions.



Trigger #	Trigger Conditions	Actions
PT1	Injection time reached	Passenger joins hall queue.
PT2	Doors at start landing are open wider than width, car is level or leveling is being ignored, car lantern is consistent with travelDirection, passenger has priority to use the door.	
PT3	After doorTraversalDelay, the door is not open wide enough for the passenger to enter.	Passenger registers a "failed to enter" satisfaction deduction
PT4	doorBackoutDelay expires	Passenger rejoins hall queue at the end of the queue.
PT5	doorTraversalDelay expires and the doors are open wider than passenger width.	Passenger exits the hall queue and joins the car queue.
PT6	The CarWeightAlarm sounds and this passenger is the last passenger in the car queue.	Passenger stops monitoring the car call.
PT7	doorTraversalDelay expires and the doors are open wider than the passenger width.	Passenger rejoins hall

		queue.
PT9	Doors are fully closed and BackoffPeriod has elapsed.	
PT10	CarWeightAlarm sounds and the passenger is in the hall queue with the doors at least partially open.	
PT11	Doors at destHallway are open wider than passenger width, the car is level or leveling is being ignored, the position indicator indicates destFloor, and the passenger has priority to traverse the door. OR, while the car is still at the startFloor and the doors on the startHallway are still open, the lantern changes to a direction that is inconsistent with the passengers travelDirection and the passenger has priority to traverse the door.	
PT12	After doorTraversalDelay, the door is not open wide enough for the passenger to enter.	Passenger registers a "failed to exit" satisfaction deduction.
PT13	doorBackoutDelay expires	Passenger rejoins the car queue at the end of the queue.
PT14	doorTraversalDelay expires and the doors are open wider than the passenger width.	Passenger records delivery time and stops interacting with the simulation.
PT15	Car ceases to be overweight before the passenger can exit the car.	Passenger rejoins the car queue

State	Description	Actions
INIT	passenger has not arrived at the hallway yet	none
WAITING_IN_HALL	waiting for the car to arrive at the start floor/hall, pressing the hall call	Passenger presses and/or monitors the hall call. Passenger may register a "repeated call press" deduction.
ENTERING	in the process of entering the car	Passenger may register a "car lantern direction change" deduction.
ENTER_BACKOUT	backing out because the passenger failed to enter	none
WAITING_IN_CAR	waiting for the car to arrive at the destination	Passenger presses and/or monitors the car call. Passenger may register "missedOpeningThreshold exceeded", "skipped destination" or "wrong motion" deductions. Passenger may register a "repeated call press" deduction.
EXITING	exiting the car at the start or destination landing	none
EXIT_BACKOUT	returning to the car because the passenger failed to exit	none

OVERWEIGHT_EXITING	exiting the car because it is overweight	Passenger registers a "failed to exit" satisfaction deduction.
OVERWEIGHT_BACKOFF	waiting after the overweight alarm has sounded	none
DONE	passenger delivered to destination	none

2. Safety (environmental object)

Replication:

- One Safety object per car. This is a separate object to simplify system safety certification.

Instantiation:

- The safety system starts assuming a safe system state at initialization (initialization must ensure that an unsafe state is not transiently generated).

Assumptions:

- Sensors monitored by Safety object do not fail.
- Safety Object can always trigger EmergencyBrake.
- Safety Object does not fail.

Input Interface:

- AtFloor[f, b]
- CarWeight
- CarPosition (internal simulator variable)
- DoorClosed[b, r]
- DoorMotor[b, r]
- DoorReversal[b, r]
- Drive
- DriveSpeed
- HoistwayLimit[d]

Output Interface:

- EmergencyBrake (internal simulator variable)
- mEmergencyBrake

State:

None

BEHAVIORS:

2.1 If all AtFloor[f, b]s are False and any DoorClosed[b, r] is False, set EmergencyBrake to on.
2.2 If any DoorClosed[b,r] is false when there is no landing at hallway b, set EmergencyBrake to on.
2.3 If DriveSpeed.speed exceeds LevelSpeed and DriveSpeed.direction is not equal to Stop and any DoorClosed[b,r] is false, set the Emergency Brake to on.
2.3 If any DoorReversal[b, r] is True and any DoorMotor[b, r] is anything other than Open or Nudge for greater than 200msec (accumulated while DoorReversal[b, r] remains True), set EmergencyBrake to on.
2.4 If any HoistwayLimit[d] is True, set EmergencyBrake to on.
2.5 If a Drive command is not "adjacent to" or the same as the current DriveSpeed value, set EmergencyBrake to on.

2.5.1 In the following table, any entry with an 'X' means that that Drive command is considered adjacent to the corresponding value of {DriveSpeed}:

	Drive Speed Values						
Drive Command Values	FastSpeed >= DriveSpeed >= SlowSpeed, UP	SlowSpeed >= DriveSpeed > LevelSpeed, UP	LevelSpeed >= DriveSpeed > 0, UP	DriveSpeed == 0,STOP	LevelSpeed >= DriveSpeed > 0, DOWN	SlowSpeed >= DriveSpeed > 0, DOWN	FastSpeed >= DriveSpeed >= SlowSpeed, DOWN
Fast, Up	X						
Slow, Up	X	X	X	X			
Level, Up		X	X	X			
Stop, Stop			X	X	X		
Level, Down				X	X	X	
Slow, Down				X	X	X	X
Fast, Down							X

2.6 If DriveSpeed.speed exceeds LevelSpeed and DriveSpeed.direction is not equal to Stop and CarWeight MaxCarCapacity, set the EmergencyBrake to on.

3. Drive (environmental object)

Replication:

- One Drive per Car, which tracks position of Car in hoistway as well as Drive direction/speed. The electric motor of the Drive is double-wound, so that if one winding breaks the Drive can still deliver both slow and fast speeds at approximately half the torque as for Slow and Fast of a fully operational drive. (There are many ways to deal with failure modes! This is a simple one for this project.)

Instantiation:

- Drive is Off at initialization.

Assumptions:

- Drive does not go faster than Fast speed.
- EmergencyBrake can stop car at any speed.

Input Interface:

- Drive
- EmergencyBrake (internal simulator variable)

Output Interface:

- CarPosition (internal simulator variable)

State:

- F_position[f]: an array initialized with the vertical position of each floor; used implicitly by the behaviors to determine floor position.

CONSTRAINTS:

3.1 If the EmergencyBrake is activated, it shall stop the Car regardless of Drive speed and direction.
3.2 Car movement caused by cable slip shall not exceed 1cm.

BEHAVIORS:

3.2 Drive(Stop,d) and Drive(s,Stop) shall stop the Car regardless of values for s or d.
3.2.1 The time to Stop the Car from Fast speed depends on the Car speed before Stop is commanded and is determined by an acceleration profile.
3.2.2 The time to Stop the Car from Slow speed shall be less than or equal to 250 msec.
3.2.2 The time to Stop the Car from Level speed shall be less than or equal to 50 msec.
3.3 Drive(level, d), where d is not Stop, shall move the elevator at a leveling speed in the direction d.
3.3.1 The time to achieve Level speed depends on speed preceding the Level movement command and is determined by an acceleration profile.
3.3.2 The actual velocity at Level speed depends on the drive equipment installed.
3.4 Drive(Slow,d), where d is not Stop, shall move the elevator at a slow speed in the direction d.
3.4.1 The time to achieve Slow speed depends on speed preceding the Slow movement command and is determined by an acceleration profile.
3.4.2 The actual velocity at Slow speed depends on the drive equipment installed.
3.5 Drive(Fast,d), where d is not Stop, shall move the elevator at maximum possible speed in the direction d as determined by a velocity profile.
3.6 CarPosition shall be updated according to integration of Car speed as determined by Drive() commands and an acceleration profile.
3.7 Cable slip may cause CarPosition to change while the Drive is commanded to Stop,

4. DoorMotor[b, r]

Replication:

- Each Car has four DoorMotor[b, r], with each controlled by DoorControl[b, r]. DoorMotor[b, r] keeps track of the actual position of the door. Each Door[b,r] contributes from 0% to 50% of the total DoorPosition[b,r].

Instantiation:

- All DoorMotor[b, r] are at Stop at initialization.

Assumptions:

None

Input Interface:

- DoorMotor[b, r]

Output Interface:

- DoorPosition[b, r] (internal simulator variable)

Velocity Profile:

- Time to open the door fully is 2 seconds or less.
- Time to close the door fully when commanded to Close is 2 seconds or less.
- Time to close the door fully when commanded to Nudge is 20 seconds or less if the doors are not blocked.

State:

- D_position: double precision value in units of percent of total door opening width. Since there are two doors, the range of each door is [0,50]. 0 represents fully closed. 50 represents fully open.

CONSTRAINTS:

4.1 D_position shall be within the range [0,50] regardless of DoorMotor[b, r] commands.
4.2 The DoorMotors[b, r] themselves shall not activate DoorReversal[b, r] sensors.
4.3 DoorMotors[b, r] shall operate properly even if transitioned between Open and Close (or Nudge) in either direction without an intermediate Stop command.

BEHAVIORS:

4.4 DoorMotor[b, r](Stop) shall stop changes in door position within 100 msec.
4.5 DoorMotor[b, r](Open) and DoorMotor[b, r](Close) shall cause door[b, r] to Open and Close respectively according to a velocity profile.
4.6 DoorMotor[b,r](Nudge) shall cause the door to close at a speed that is slow enough to avoid injuring passengers according to a velocity profile.
4.7 If the door is blocked by a passenger and the DoorMotor[b,r] is commanded to Nudge, the door shall stop when it makes contact with the passenger.
4.6 D_position shall be kept updated by a physical model to indicate current positions of simulated doors.

5. DoorControl[b, r]

Replication:

- Each Car has four DoorControllers[b, r].

Instantiation:

- DoorControllers[b, r] shall command doors[b, r] to close at initialization.

Assumptions:

- DoorControllers[b, r] do not command doors to open unless at least one AtFloor [f, b] sensor is true.
- DoorControllers[b, r] do not command doors to open when car is moving faster than Level Speed.

Input Interface:

- mAtFloor[f, b]
- mDriveSpeed
- mDesiredFloor
- mDesiredDwell[b]
- mDoorClosed[b, r]
- mDoorOpened[b, r]
- mDoorReversal[b, r]
- mCarCall[f, b]
- mHallCall[f, b, d]
- mCarWeight

Output Interface:

- DoorMotor[b, r]
- mDoorMotor[b, r]

State:

- Dwell, long integer with number of msec desired for door dwell during current cycle.
- CountDown: a count-down timer for Door Dwell[b] (implemented in simulation by scheduling a future task execution at time of expiration)

CONSTRAINTS:

5.1 DoorClosed[b, *] shall be True when there is no mAtFloor[f, b] that is True.
5.2 Any DoorReversal[b,*] shall not be True for more than an accumulated time of 50 msec without causing all DoorControllers[b,*] to perform an Open or Nudge command.
5.3 Doors should keep moving in desired direction unless commanded otherwise, subject to the constraints of the door object.
5.4 All doors should be commanded to identical positions at all times.
5.5 If CarWeight(x) >= MaxCarCapacity, the doors shall open completely until the car is no longer overloaded.

TIME-TRIGGERED BEHAVIORS:

5.6 If any mAtFloor[f, b] is True and f equals mDesiredFloor.f, and mDriveSpeed = (0, d) or mDriveSpeed = (s, Stop) then:
    5.6.1 DoorMotor[b, r] shall be commanded to Open;
    5.6.2 CountDown shall be set to Dwell.
5.7 If mDoorOpened[b, r] is True, then
    5.7.1 DoorMotor[b,r] shall be commanded to Stop.
    5.7.2 CountDown shall be decremented.
5.8 If mDoorClosed[b, r] is True, then DoorMotor[b,r] should be commanded to Stop.
5.9 When CountDown <= 0, DoorMotor[b,r] should be commanded to Nudge.
5.10 If mCarWeight(g) >= MaxCarCapacity, and mDoorOpened[b, r] is False, DoorMotor[b, r] shall be commanded to Open.
5.11 Dwell shall be set to an appropriate value based on mDesiredDwell[b].
5.12 mDoorMotor[b,r] shall be set to the current value of DoorMotor[b,r]

6. DriveControl

Replication:

- There is one DriveControl, which controls the elevator Drive (the main motor moving Car Up and Down). For simplicity we will assume this node never fails, although the system could be implemented with two such nodes, one per each of the Drive windings.

Instantiation:

- DriveControl initializes to Stopping the Drive.

Assumptions:

- DriveControl does not command Drive to move faster than LevelSpeed when any DoorClosed[b, r] is False.
- DriveControl only commands Drive to stop when car is level with a floor.

Input Interface:

- DriveSpeed
- mAtFloor[f, b]
- mLevel[d]
- mCarLevelPosition
- mDoorClosed[b, r]
- mDoorMotor[b, r]
- mEmergencyBrake
- mDesiredFloor
- mHoistwayLimit[d]
- mCarWeight

Output Interface:

- Drive
- mDrive
- mDriveSpeed

State:

- DesiredDirection = {Up, Down, Stop} computed desired direction based on comparing current floor position with Floor desired by Dispatcher. This is implicitly computed and used as a macro in the behavior descriptions.
- CurrentFloor, is a shorthand notation for the value of whichever mAtFloor[f, b] is True, if any. If CurrentFloor is invalid it has a mnemonic value of None.
- CommitPoint[f](v); v = {Reached, NotReached} calculated based on car's acceleration profile, current speed, and distance from the floor f, to determine whether or not the car can still stop at the next floor.

CONSTRAINTS:

6.1 Drive shall have been commanded to Stop whenever any mDoorClosed[b, r] is False.
6.2 Drive shall have been commanded to Stop whenever any mDoorMotor[b, r] is commanded to Open.
6.3 The commanded value of Drive shall either be the same as or "adjacent to" the value of DriveSpeed.
6.3.1 In the following table, any entry with an 'X' means that that Drive command is considered adjacent to the corresponding value of {DriveSpeed}:

	Drive Speed Values						
Drive Command Values	FastSpeed >= DriveSpeed >= SlowSpeed, UP	SlowSpeed >= DriveSpeed > LevelSpeed, UP	LevelSpeed >= DriveSpeed > 0, UP	DriveSpeed == 0, STOP	LevelSpeed >= DriveSpeed > 0, DOWN	SlowSpeed >= DriveSpeed > 0, DOWN	FastSpeed >= DriveSpeed >= SlowSpeed, DOWN
Fast, Up	X						
Slow, Up	X	X	X	X			
Level, Up		X	X	X			
Stop, Stop			X	X	X		
Level, Down				X	X	X	
Slow, Down				X	X	X	X
Fast, Down							X

6.4 Drive should be Stopped whenever mEmergencyBrake is activated.
6.5 If mCarWeight(x) >= MaxCarCapacity, the drive shall not be commanded to {Slow, d} or {Fast, d}until the car is no longer overloaded.

EVENT-TRIGGERED BEHAVIORS:

7. LanternControl[d]

Replication:

- Two controllers, one for each lantern {Up, Down} mounted in the Car. Each controller controls two lightbulbs in parallel (one by each of the Car's front and back doors), and actuates each front/back pair of bulbs as a single actuator.

Instantiation:

- Lanterns are Off at initialization.

Assumptions:

- CarLanterns[d] are never commanded to be On at the same time.

Input Interface:

- mDoorClosed[b, r]
- mDesiredFloor
- mAtFloor[f, b]

Output Interface:

- CarLantern[d]
- mCarLantern[d]

State:

- DesiredDirection = {Up, Down, Stop} computed desired direction based on comparing CurrentFloor with Floor desired by Dispatcher. This is implicitly computed and used as a macro in the behavior descriptions.

CONSTRAINTS:

7.1 Both CarLanterns[d] shall not be On at the same time.

EVENT-TRIGGERED BEHAVIORS:

8. HallButtonControl[f, b, d]

Replication:

- There are two HallButtonControllers[f, b, d] per hallway, one for each of the Up and Down HallCall buttons. (Topmost floor does not have an Up button; bottom floor does not have a Down button.) These accept HallCall button presses as well as control HallLight feedback lights.

Instantiation:

- All HallCalls are False at initialization.
- All HallLights are off at initialization.

Assumptions:

None

Input Interface:

- mAtFloor[f, b]
- mDesiredFloor
- mDoorClosed[b, r]
- HallCall[f, b, d]

Output Interface:

- HallLight[f, b, d]
- mHallLight[f, b, d]
- mHallCall[f, b, d]

State:

- CurrentFloor, is a shorthand notation for the value f of whichever mAtFloor[f, b] is True, if any. If CurrentFloor is invalid it has a mnemonic value of None.

CONSTRAINTS:

None

EVENT-TRIGGERED BEHAVIORS:

9. CarButtonControl[f, b]

Replication:

- There is one CarButtonController per hallway [f, b], with all controllers located in the Car. These accept CarCall button presses as well as control CarLight feedback lights.

Instantiation:

- All CarCalls are False at initialization.
- All CarLights are off at initialization.

Assumptions:

None

Input Interface:

- mAtFloor[f, b]
- mDesiredFloor
- mDoorClosed[b, r]
- CarCall[f, b]

Output Interface:

- CarLight[f, b]
- mCarLight[f, b]
- mCarCall[f, b]

State:

- CurrentFloor, is a shorthand notation for the value f of whichever mAtFloor[f, b] is True, if any. If CurrentFloor is invalid it has a mnemonic value of None.

CONSTRAINTS:

None

EVENT-TRIGGERED BEHAVIORS:

10. CarPositionControl

Replication:

- There is one CarPositionControl instance in the car, which feeds values to the CarPositionIndicator.

Instantiation:

- The Car is initialized on the first Floor (Lobby).

Assumptions:

None

Input Interface:

- mAtFloor[f, b]
- mCarLevelPosition
- mDesiredFloor
- mDriveSpeed

Output Interface:

- mCarPositionIndicator
- CarPositionIndicator

State:

- CurrentFloor, is a shorthand notation for the value f of whichever mAtFloor[f, b] is True, if any. If CurrentFloor is invalid it has a mnemonic value of None.
- CommitPoint[f](v); v = {Reached, NotReached} calculated based on car's acceleration profile, current speed, and distance from the floor f, to determine whether or not the car can still stop at the next floor.

CONSTRAINTS:

10.1 The Car can be at only one position at a time.
10.2 The CarPositionIndicator shall display the current floor whenever any of the doors are open.
10.3 The floor indicated by the car position indicator shall only change by one floor in either direction per update cycle, and should be a close approximation to the car's actual position. The direction of change shall be in the same direction the Drive is moving. By "close approximation" we mean within stopping distance in the direction of motion.

TIME-TRIGGERED BEHAVIORS:

10.4 Whenever any mAtFloor[f, b] is True, CarPositionIndicator shall be commanded to display CurrentFloor.

11. Dispatcher

Replication:

- There is one Dispatcher in the system, corresponding with the Car.

Instantiation:

- The Dispatcher is initialized to send the car to the Lobby, have the Lobby as the desired destination, and have a preferred direction of "Stopped" (i.e., no preferred direction).

Assumptions:

- Dispatcher never sets DesiredFloor.f to be above MaxFloor or below Lobby (1).

Input Interface:

- mAtFloor[f, b]
- mDoorClosed[b, r]
- mHallCall[f, b, d]
- mCarCall[f, b]
- mCarWeight

Output Interface:

- mDesiredFloor
- mDesiredDwell

State:

- Target: an integer Floor number for desired Floor, initialized to Lobby = 1.

CONSTRAINTS:

11.1 Target shall be a valid Floor number from 1 .. MaxFloor inclusive.
11.2 The desired direction d of mDesiredFloor(f, b, d) shall not be Up when f = MaxFloor .
11.3 The desired direction d of mDesiredFloor(f, b, d) shall not be Down when f = 1.

TIME-TRIGGERED BEHAVIORS:

11.4 mDesiredFloor.f shall always be set to Target.
11.5 mDesiredFloor.d shall always be set to Stop.
11.6 Whenever any mDoorClosed [b, r] is False, then
  11.6.1 Target shall be set equal to ((f mod MaxFloors) + 1)
  11.6.2 mDesiredFloor.b shall be set to b, where f, b is whichever mAtFloor[f, b] is True
11.7 If all mAtFloor[f, b] are False AND any mDoorClosed [b, r] is False (which means doors are not closed between floors!); then
  11.7.1 Target shall be set to 1
  11.7.2 mDesiredFloor.b shall be set to None .
11.8 If two mAtFloor[f, b] values are True with the same value f, then mDesiredFloor.b shall be set to Both.
11.9 mDesiredDwell shall always be set to a constant appropriate value for door open dwell.