# ECE 18-649
# Final Project Report

December 9, 2015

Group # 12

Shepard Emerson
Daniel Gorziglia
Daniel Haddox
Tom Eliot



All

# Outline

- Project Statistics
- Controller Design, Door Control
  - Scenarios
  - Sequence Diagrams
  - Requirements
  - Statechart
  - Code
  - Testing
  - System Perspective
- Lessons Learned
- Open Issues

# Project Statistics

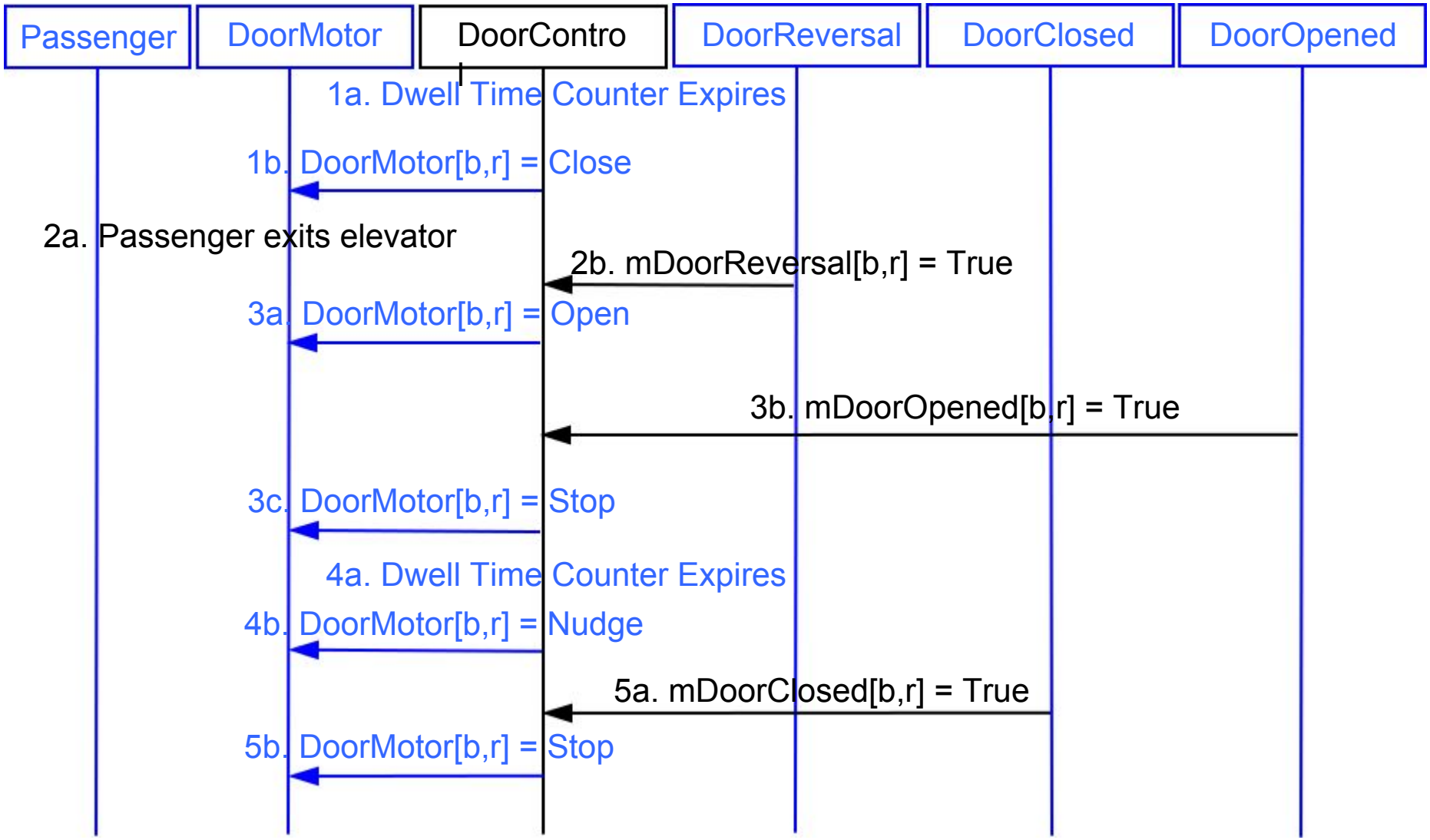
| | **Mid Semester** | **Final Project** |
|---|---|---|
| Scenarios and SDs | 18 | 21 |
| Lines of Requirements | 37 | 48 |
| Statecharts | 21 States, 26 Arcs | 28 states, 45 Arcs |
| Lines of non-comment code | 1870 | 3589 |
| Test files | 33 | 43 + 40 + more |
| Git Commits | 612 | 980 |
| Peer Reviews | 60 (47 defects found) | 103 (60 defects found) |
| Defects found via test | 20 found, all fixed | 35 found, all fixed |

# DoorControl: Scenarios and Sequence Diagrams

**Most Relevant Scenarios:**

- 4A: Passenger in elevator as it arrives, then exits
  - Required for doors to open at destination
- 5A/B: Passenger enters/exits elevator
  - Passenger gets in way of door
  - Required for the doors to reverse
- 7C: Elevator doors close on hallway
  - Dwell count expires and doors close
  - Hall/CarButtonControl and LanternControl lights turn off

# Sequence Diagram 5B



# DoorControl Design - Requirements

## State Variables

- DwellTime - long integer with number of msec desired for door dwell during current cycle.
- CountDown - a countdown timer for door dwell time
- DoorHasReversed - Boolean value indicating that DoorControl has attempted to close the door but a door reversal has occurred, initialized to False

## 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.

# DoorControl Design - Requirements (1)

## Time-Triggered Requirements

5.6 If any mAtFloor[f, b] is True and mCarCall[f,b] is True and mDriveSpeed is stop and mDesiredFloor(b) is equal to b or both, 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] shall be commanded to Stop and DoorHasReversed shall be set to False.

5.9 If CountDown <= 0 and DoorHasReversed is False, DoorMotor[b, r] shall be commanded to Close.

# DoorControl Design - Requirements (2)

## Time-Triggered Requirements

5.9 If CountDown <= 0 and DoorHasReversed is False, DoorMotor[b, r] shall be commanded to Close.

5.10 If CountDown <= 0 and DoorHasReversed is True, DoorMotor[b, r] shall be commanded to Nudge.

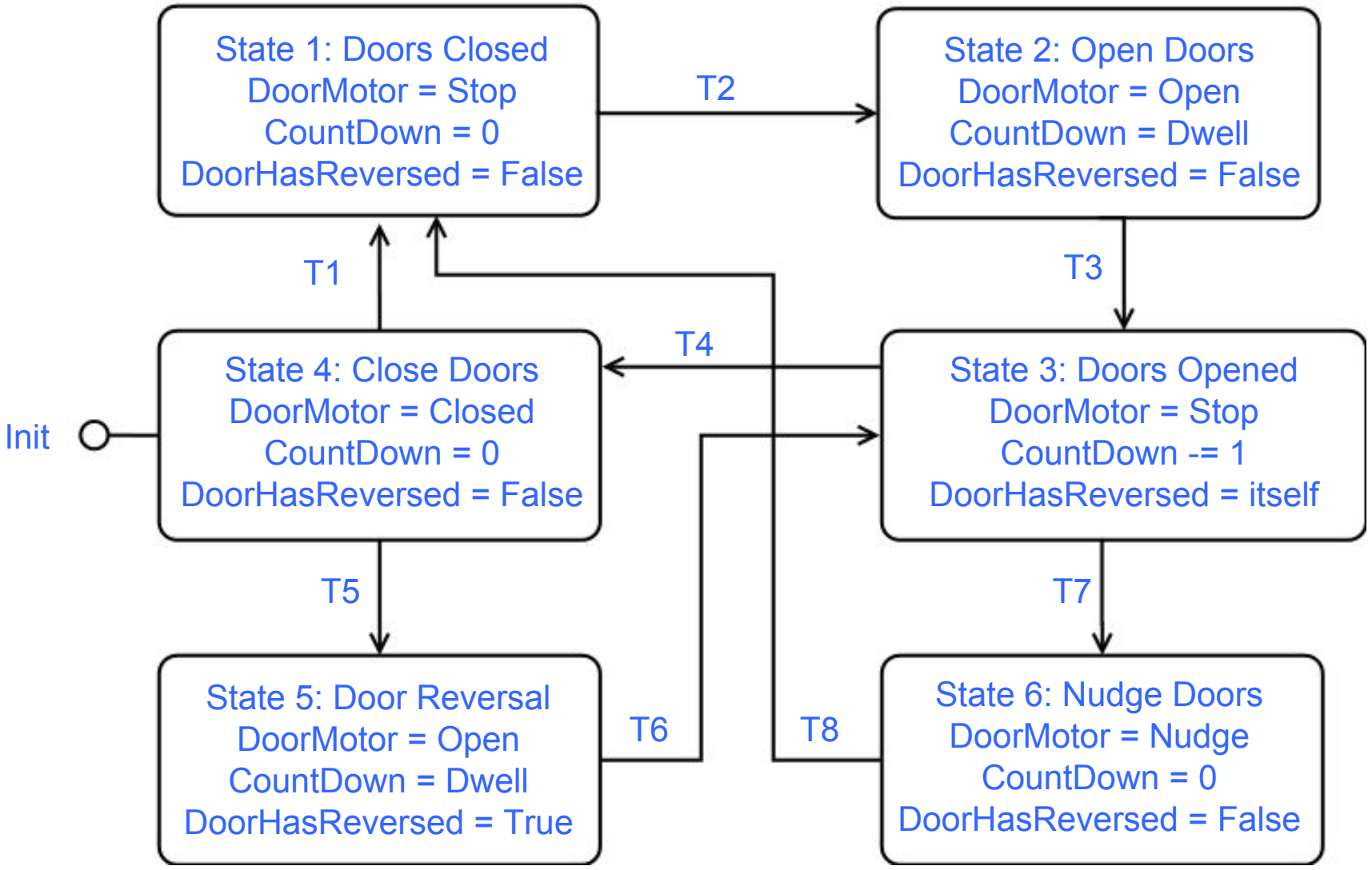
5.11 If mDoorReversal[b, r] is True, DoorHasReversed shall be set to True and DoorMotor[b,r] shall be commanded to Open.

5.12 If mCarWeight(g) >= MaxCarCapacity, and mDoorOpened[b, r] is False, DoorMotor[b, r] shall be commanded to Open.

5.13 Dwell shall be set to an appropriate value based on mDesiredDwell.

5.14 mDoorMotor[b,r] shall be set to the current value of DoorMotor [b,r]

# DoorControl Design - Statechart




For all: mDoorMotor[b,*] = DoorMotor[b,*]  Dwell = mDesiredDwell[b]

# DoorControl Design - Statechart Transitions

| Transition | Condition |
|---|---|
| 5.T.1 | mDoorClosed[b,r] == True |
| 5.T.2 | (mAtFloor[f,b] == True AND mDesiredFloor.f == f AND (mDersiredFloor.b == b OR mDesiredFloor.b == BOTH) AND (mDriveSpeed == (0,stop)) |
| 5.T.3 | mDoorOpened[b,r] == True |
| 5.T.4 | CountDown <= 0 AND DoorHasReversed == False AND mCarWeight[g] < MaxCarCapacity |
| 5.T.5 | mDoorReversal[b,r] == True OR mCarWeight[g] >= MaxCarCapacity |
| 5.T.6 | mDoorOpened[b,r] == True |
| 5.T.7 | CountDown <= 0 AND DoorHasReversed == True AND mCarWeight[g] < MaxCarCapacity |
| 5.T.8 | mDoorClosed[b,r] == True |

# DoorControl Implementation

- Controller Instantiation
- Mailbox and message translator setup
- timerExpired() state machine
  - Set outputs
  - Transition logic

# DoorControl Testing

- 2 Unit Tests, 18 passed assertions each
  - Duplicated unit test to test transition ORs
- 10 Integration tests
  - 3B, 4A, 5A, 5B, 7A, 7B, 7C, 8A, 9A, 10A
- Acceptance tests
  - Doors initially opened twice on each floor
  - Synchronization issues / time wasted
  - Required changes to multiple other controllers
    - Timing in clearing calls and commanding doors

# DoorControl Testing

- Runtime Requirement Monitors
  - R-T7: Open doors only if pending calls
  - R-T10: Only nudge doors if reversal occurred.
  - Extremely useful for testing entire system

# Door Control - System Perspective

- Dispatcher has control over doors (mDF.h)
- Doors opening inhibits dispatcher changes

- Doors opening (!closed) turns off calls
- Doors opening turns on lanterns
- Doors closed allows drive to move

- Must tune dwell time for uppeak acceptance
  - Multiple reversals considered but not used

# Lessons Learned

- Can be challenging to collaborate on design and architecture.
  - Everyone has their own approach
- Automating acceptance testing helped us discover bugs.
  - Writing a good script can make debugging much easier.
- Take breaks when working

# Open Issues

- Heavy architecture changes to dispatcher have not been propagated through the documentation
- Still finding small edge cases

# Thank you

Questions?