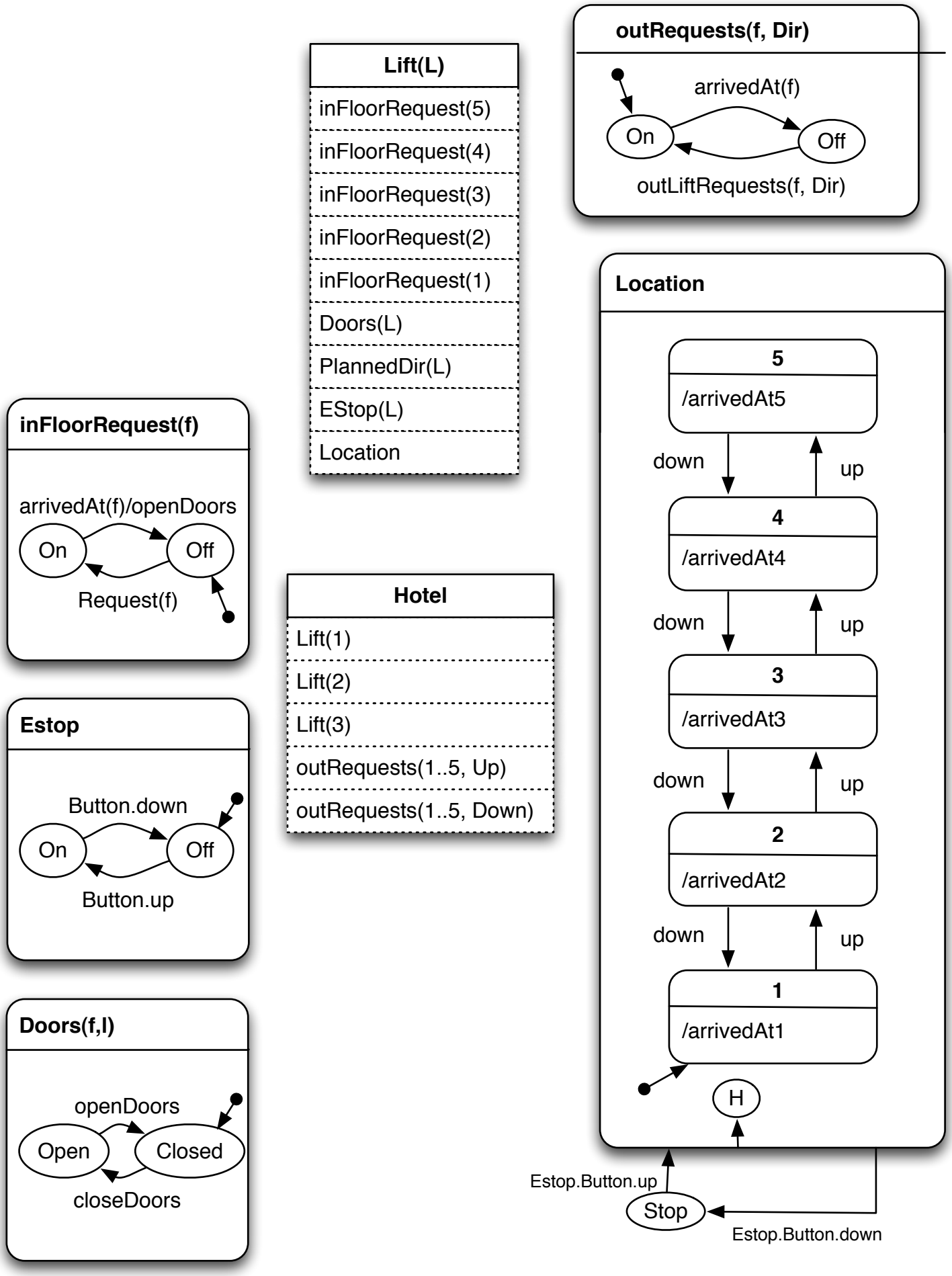
Eric Stokes
David Pick
Pete Brousalis

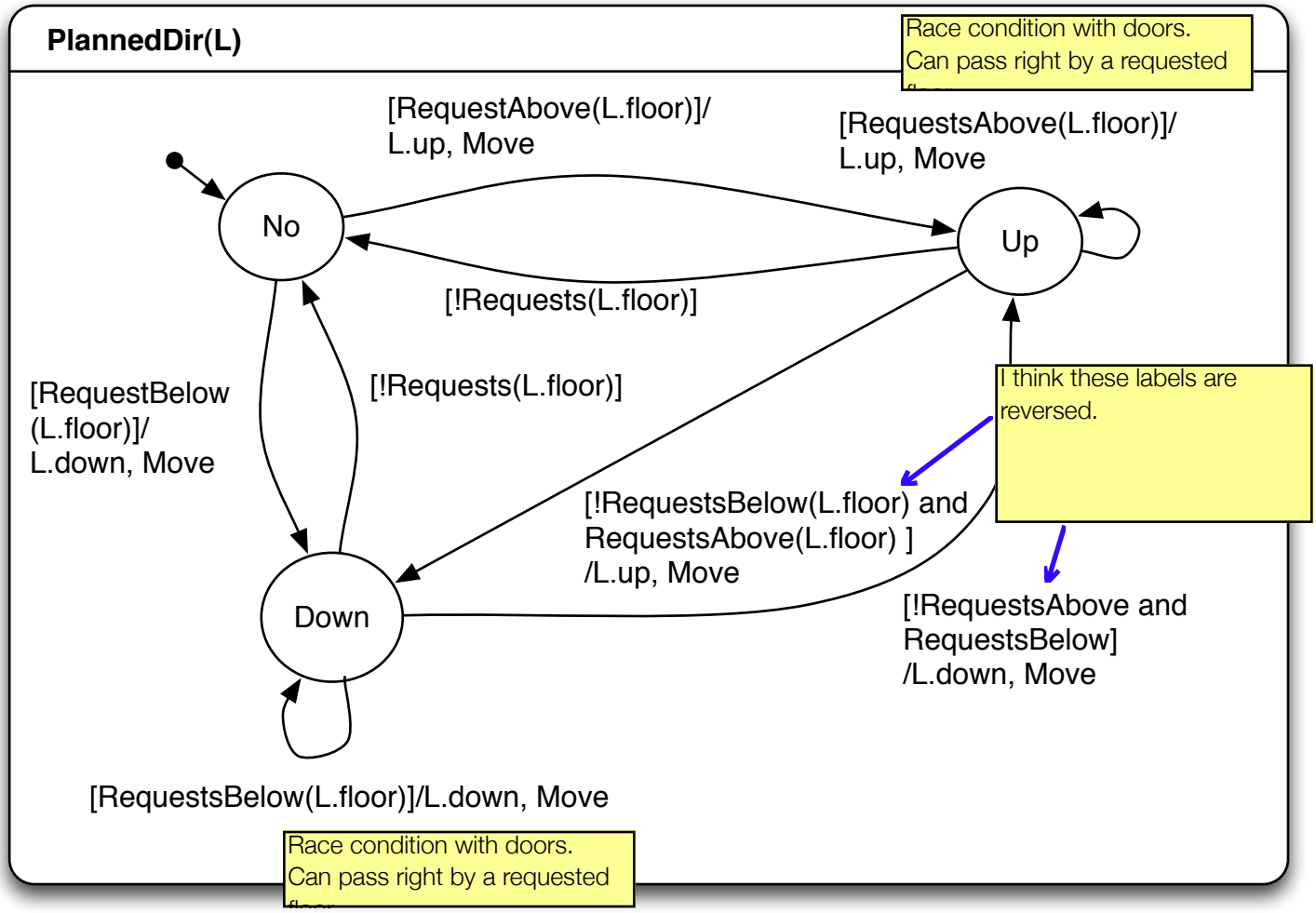
Elevator Project: Milestone 4
State Chart

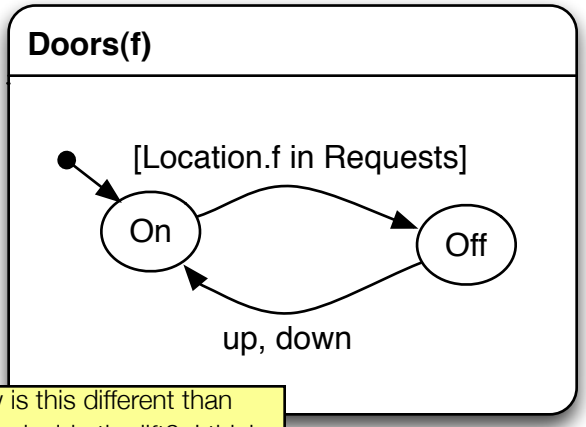
Introduction

As we have progressed through the project, we have evolved our elevator simulation from a basic to advanced alloy model. We then implemented this alloy model in Java using JML to prove it. Finally we are producing a State Chart depicting the elevators and how they behave. By creating the state chart, we can actually progress through all possible positions and possibilities that could occur with the elevator system. For both convenience and readability, we have used several state aliases so that we can represent larger portions of the system in a more generic manner. We did add a hotel submachine so we can represent the whole system. From that one master state, we have several submachines that operate in an orthogonal pattern. These submachines are the lifts and the floors. We decided to create the floor machine to handle the up and down requests from buttons pressed on that floor. There are three lifts handled in our model, which effectively behave independently, each lift contains doors, and emergency stop segment. We then also have two different types of requests, those that are inside an elevator and those that are on floors.

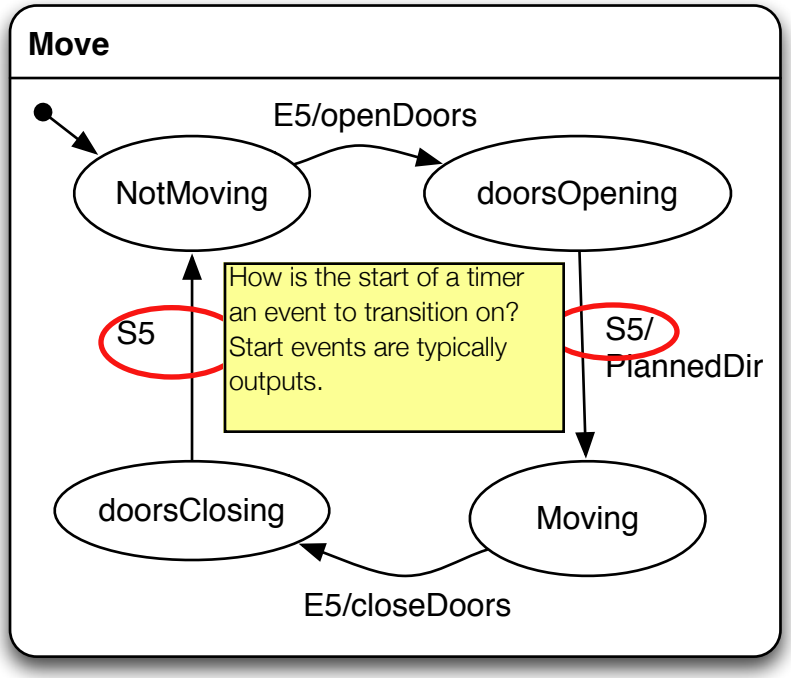
The control algorithm is implemented with a combination of the PlannedDir, move and Doors sub charts in combination with several pieces of Alloy code. This control algorithm allows us to move lifts between floors and ensures that the doors are not moving while the elevator is moving and vice versa. All elevators will simultaneously service a floor, due to the fact we do not have data as to how an actual elevator system performs this behavior. The state charts are included below.


Lift(L)
inFloorRequest(5)
inFloorRequest(4)
inFloorRequest(3)
inFloorRequest(2)
inFloorRequest(1)
Doors(L)
PlannedDir(L)
EStop(L)
Location
outRequests(f, Dir)
arrivedAt(f)
On
Off
outLiftRequests(f, Dir)
inFloorRequest(f)
arrivedAt(f)/openDoors
On
Off
Request(f)
Hotel
Lift(1)
Lift(2)
Lift(3)
outRequests(1..5, Up)
outRequests(1..5, Down)
Location
5
/arrivedAt5
down
up
4
/arrivedAt4
down
up
3
/arrivedAt3
down
up
2
/arrivedAt2
down
up
1
/arrivedAt1
H
Estop
Button.down
On
Off
Button.up
Doors(f,l)
openDoors
Open
Closed
closeDoors
Estop.Button.up
Stop
Estop.Button.down





How is this different than doors inside the lift? I think you just wanted the Doors on the previous page.



Missing up/down arrows.

You meant &&. (I think you mis-read my whiteboard ampersands.)

Alloy

```
RequestsAbove(f) = some f, f' | f' > f ++ in outLiftRequests(f', Up).ON

RequestsBelow(f) = some f, f' | f' < f ++ in outLiftRequests(f', Down).ON

outLiftRequests(f, dir) = f in outLiftRequests(f, dir).On

arrivedAt(f) = some y:{1..3} | in Lift(y).Floor(f) ++ in Lift(y).Doors.Open

requests(f) = RequestsAbove(f) ++ RequestsBelow(f)
```

You used this as an event, not as a guard. No need to define it like this.

P4 Milestone

Grade: 84

Criteria (weight)	5 Exemplary	3 Satisfactory	1 Needs Improvement	Weighted Score
Organization of Report (x1)	Report is well organized with appropriate sections.	Report is mostly well organized with some sections.	Report is not well organized. There is no obvious logic to the order.	
Clarity and Conciseness of Prose (x2)	Written description is clear and unambiguous. It is not unnecessarily wordy.	Written description is mostly clear and unambiguous. Occasional instances of unclear or awkward writing.	Written description is unclear and ambiguous, or else missing altogether.	
Professionalism (x2)	Report presents a professional tone. It could be shared with a "real-world" customer without changes.	Report largely presents a professional tone. It could be shared with a "real-world" customer with minor revisions.	Report is unprofessional. The majority of the prose would have to be rewritten before sharing the report with a "real-world" customer.	
Depth of Analysis (x4)	Model covers all important corner cases. It demonstrates a deep understanding of the problem.	Model covers many important corner cases. Some cases might be treated in an unusual manner, but such treatment is documented.	Model treats few or no corner cases. It demonstrates just a superficial understanding of the problem.	
Logical Correctness (x4)	Model is free of logical and semantic errors. Formal model matches prose description.	Model is mostly free of logical and semantic errors. Such errors are limited to a very few different sorts.	Model has many logical errors of a variety of kinds.	
Clarity of Formalism (x4)	Diagram is well-labeled and at an appropriate level of abstraction so that a customer familiar with the problem domain could readily understand the specification.	Diagram is mostly well-labeled, with at most three cryptic labels. Diagram is generally at an appropriate level of abstraction, though a customer familiar with the problem domain might need some guidance to understand the specification.	Labels are cryptic or abstraction is used to the point that the actual specification would be obscured to all but an expert in the notation.	
Conciseness of Formalism (x3)	Specification appropriately uses the abstraction features of the notation to minimize unhelpful redundancy.	Specification may include some unhelpful redundancy, but the general requirements are still readily comprehensible	Specification is highly redundant. The volume of redundancy makes comprehension of the specification very difficult.	
			Total Score:	84