

SOFTWARE SPECIFICATIONS

Elevator

Group 25

Miao Li

Table of Contents

[System Architecture 3](#_bookmark0)

[T1: Unit Test 3](#_bookmark1)

[T1.1: SystemProcessor Unit Test 3](#_bookmark2)

[T1.1.1: Test pressButton() 3](#_bookmark3)

[T1.1.2: Test open/ coseDoor() 4](#_bookmark4)

[T1.1.3: Test longerOpen() 5](#_bookmark5)

[T1.2: Controller Unit Test 8](#_bookmark11)

[T1.2.1: Test mixedTest() 8](#_bookmark12)

[T1.2.19: Test changeTimer() 20](#_bookmark30)

[T1.3: MainBoardUI Unit Test 20](#_bookmark31)

[T1.3.1: Test show() 20](#_bookmark32)

[T2: Integration Test 22](#_bookmark33)

[T2.1: SystemProcessor+SystemDB+mainBoardUI Integration 22](#_bookmark34)

[T2.2: SystemProcessor+SystemDB+mainBoardUI+PasswordUI+SettingUI Integration 23](#_bookmark35)

[T3: Functional Test 24](#_bookmark36)

[T3.1: Use Case “Get basic information” 24](#_bookmark37)

[T3.1.2: Test ‘Get changed information’ 24](#_bookmark39)

[T4: Model Checking 26](#_bookmark47)

[Full UPPAAL Model 26](#_bookmark48)

[Customer 27](#_bookmark49)

[Elevator1 27](#_bookmark50)

[Elevator2 28](#_bookmark51)

[Check Properties 28](#_bookmark52)

# System Architecture

The system architecture is shown below:

I write test functions in file `test.m`.

# T1: Unit Test

## T1.1: SystemProcessor Unit Test

### *T1.1.1: Test updateInjectionSpeed ()*

classdef test < matlab.uitest.TestCase

properties

inUI\_1

inUI\_2

outUI\_0

outUI\_1

outUI\_2

outUI\_3

end

methods (TestMethodSetup)

function launchApp(testCase)

% init properties

f = 3; % floorHeight f:[-f, 0, f, 2f] -> [-1, 1, 2, 3]

t = 0.5;%Timer Period

openTime = 3;%time pperiod openning the door

ele\_1=Elevator(1);% 1: 0 1 2

ele\_2=Elevator(2);% 2: -1 0 1 2

control=ElevatorController(f, openTime);

testCase.inUI\_1=ElevatorInsideUI(control, ele\_1, 1);

testCase.inUI\_2=ElevatorInsideUI(control, ele\_2, 2);

testCase.outUI\_0=ElevatorOutsideUI\_B1;

testCase.outUI\_1=ElevatorOutsideUI\_F1\_F2("1");

testCase.outUI\_2=ElevatorOutsideUI\_F1\_F2("2");

testCase.outUI\_3=ElevatorOutsideUI\_F3;

% set ElevatorController's properties

control.elevator\_1 = ele\_1;

control.elevator\_2 = ele\_2;

control.elevatorInUI\_1 = testCase.inUI\_1;

control.elevatorInUI\_2 = testCase.inUI\_2;

control.elevatorOutUI\_0 = testCase.outUI\_0;

control.elevatorOutUI\_1 = testCase.outUI\_1;

control.elevatorOutUI\_2 = testCase.outUI\_2;

control.elevatorOutUI\_3 = testCase.outUI\_3;

%set ElevatorOutUI's properties

testCase.outUI\_0.Process=control;

testCase.outUI\_1.Process=control;

testCase.outUI\_2.Process=control;

testCase.outUI\_3.Process=control;

%set the .mlapp files' properties

testCase.f1app.control=c;

testCase.f2app.control=c;

testCase.f3app.control=c;

testCase.e1app.control=c;

testCase.e1app.fSensor=fS;

testCase.e2app.control=c;

testCase.e2app.fSensor=fS;

%set the timer

control.Timer = timer('ExecutionMode', 'fixedRate', ... % Run timer repeatedly

'Period', t, ... % Period is 1 second

'TimerFcn', @control.timerFcn); % Specify callback function

testCase.addTeardown(@delete,testCase.inUI\_1);

testCase.addTeardown(@delete,testCase.inUI\_2);

testCase.addTeardown(@delete,testCase.outUI\_0);

testCase.addTeardown(@delete,testCase.outUI\_1);

testCase.addTeardown(@delete,testCase.outUI\_2);

testCase.addTeardown(@delete,testCase.outUI\_3);

end

end

methods (Test)

function tc1(testCase)

% press 3 and B1 at the same time from outside

% ele1 goes to 3 and ele2 goes to B1

testCase.press(testCase.outUI\_3.DownButton);

testCase.press(testCase.outUI\_0.UpButton);

pause(6);

% press again leads to a longer time

testCase.press(testCase.outUI\_3.DownButton);

pause(3);

end

function tc2(testCase)

% Then press 1 and 3 inside

% ele1 goes to 1 and ele2 goes to 3, almost the same time

testCase.press(testCase.inUI\_1.Button\_1);

testCase.press(testCase.inUI\_2.Button\_3);

pause(6);

% press OpenButton leads to longer time delay

testCase.press(testCase.inUI\_1.OpenButton);

pause(2);

% press OpenButton leads immediate dlose

testCase.press(testCase.inUI\_1.CloseButton);

end

end

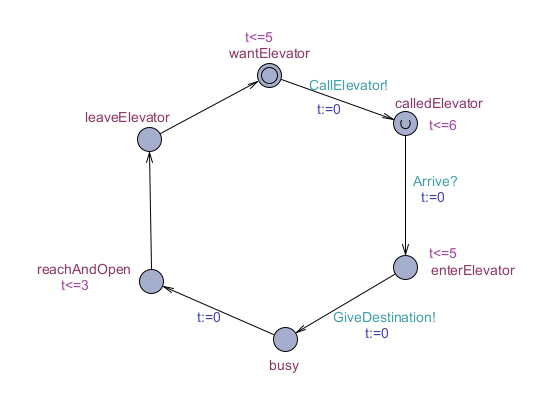
end

# T4: Model Checking

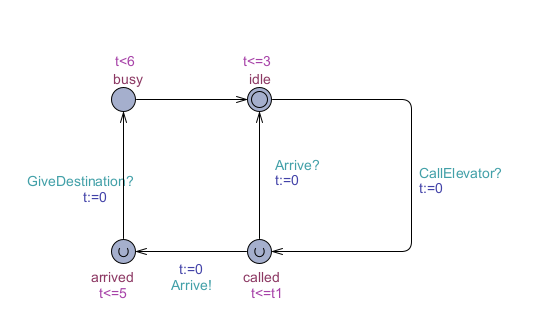
A UPPAAL model of this Elevator built for model checking.

## Full UPPAAL Model

### *Customer*

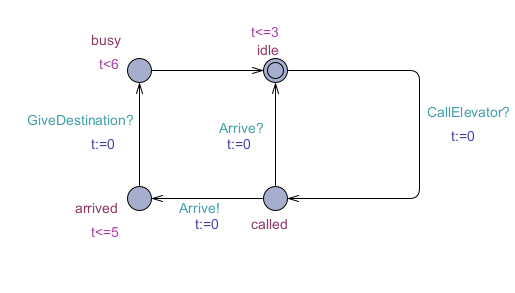


### *Elevator1*



### Just the same as the previous file.

### *Elevator2*



### *Check Properties*

### The process can last with no deadlock.