SOFTWARE SPECIFICATIONS

Elevator

Group 25
Miao Li

Table of Contents

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 ()

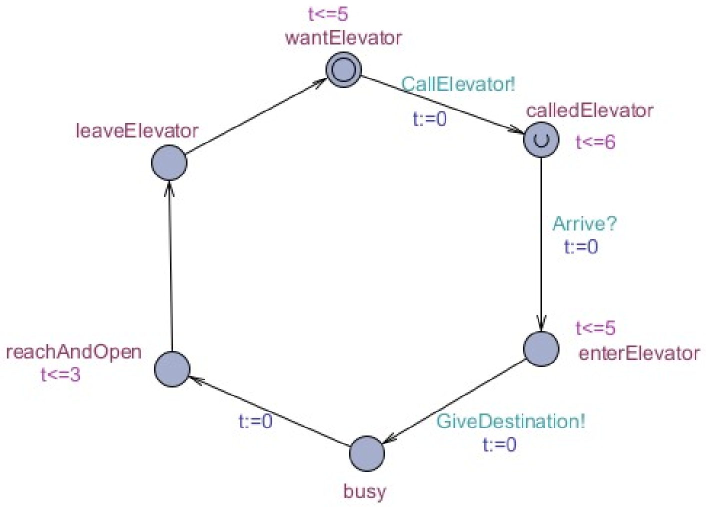
```
classdef test < matlab.uitest.TestCase
    properties
        inUI_1
        inUI_2
        outUI_0
        outUI_1
        outUI_2
        outUI_3
    end

    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

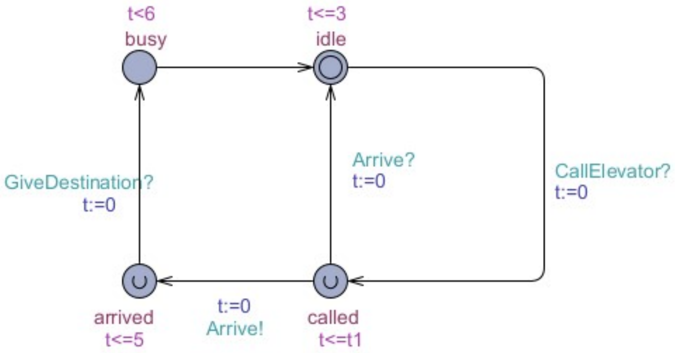
A UPPAAL model of this Elevator built for model checking.

Full UPPAAL Model

Customer

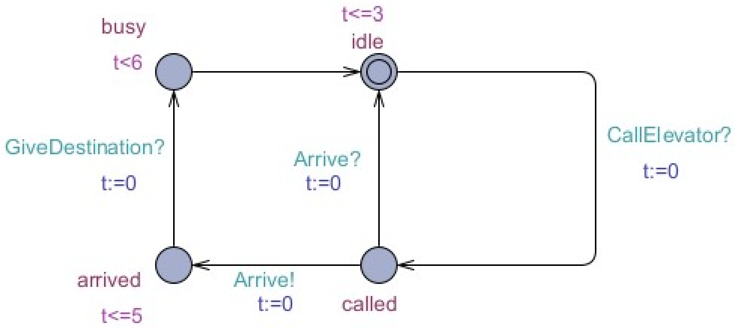


Elevator1



Just the same as the previous file.

Elevator2



Check Properties

The process can last with no deadlock.