Group X

Author: YourName

Software Specifications

Smart Restaurant System

Table of Contents

[System Architecture 2](#_Toc44468515)

[T1: Unit Test 2](#_Toc44468516)

[T1.1: InsidePanel Unit Test 2](#_Toc44468517)

[T1.2: OutsidePanel Unit Test 14](#_Toc44468518)

[T1.3: ControlUnit Unit Test 15](#_Toc44468519)

[T2: Integration Test 19](#_Toc44468520)

[T2.1: UI + InsidePanel 19](#_Toc44468521)

[T3: Functional Test 22](#_Toc44468522)

[T4: Model Checking 23](#_Toc44468523)

## System Architecture

The system architecture is shown below:

屏幕上有字

描述已自动生成

## T1: Unit Test

### T1.1: InsidePanel Unit Test

T1.1.1: testInsidePanel ()

function obj = InsidePanel()

obj.current\_floor = 1;

obj.direction = "Stationary";

obj.door\_state = "Closed";

obj.route = [0, 0, 0];

obj.pre\_direction = "Up";

obj.move\_timer = timer();

obj.door\_timer = timer();

obj.move\_time = 2;

obj.door\_time = 2; Tcover1.1.1.1

end

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.1.1 |
| Coverage Item | Tcover1.1.1.1 |
| Input |  |
| State | IP = InsidePanel; |
| Expected Output | current\_floor = 1;  direction = "Stationary";  door\_state = "Closed";  route = [0, 0, 0];  pre\_direction = "Up";  move\_time = 2;  door\_time = 2; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.2: testgetCurrentFloor ()

function curr\_flr = getCurrentFloor(obj)

curr\_flr = obj.current\_floor; Tcover1.1.2.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.2.1 |
| Coverage Item | Tcover1.1.2.1 |
| Input |  |
| State | IP = InsidePanel; |
| Expected Output | IP.current\_floor = 1; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.3: testgetPreDirection ()

function dire = getPreDirection(obj)

dire = obj.pre\_direction; Tcover1.1.3.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.3.1 |
| Coverage Item | Tcover1.1.3.1 |
| Input |  |
| State | IP = InsidePanel; |
| Expected Output | IP.pre\_direction = "Up"; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.4: testgetDirection ()

function dire = getDirection(obj)

dire = obj.direction; Tcover1.1.4.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.4.1 |
| Coverage Item | Tcover1.1.4.1 |
| Input |  |
| State | IP = InsidePanel; |
| Expected Output | IP.direction = "Stationary"; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.5: testgetDoorState ()

function state = getDoorState(obj)

state = obj.door\_state; Tcover1.1.5.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.5.1 |
| Coverage Item | Tcover1.1.5.1 |
| Input |  |
| State | IP = InsidePanel; |
| Expected Output | IP. door\_state = "Closed"; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.6: testopenDoor ()

function [] = openDoor(obj)

obj.door\_state = "Opened";

if get(obj.door\_timer, 'Running')

stop(obj.door\_timer);

end

obj.updateDisp(); Tcover1.1.6.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.6.1 |
| Coverage Item | Tcover1.1.6.1 |
| Input |  |
| State |  |
| Expected Output | ele1. door\_state = "Opened"; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.7: testcloseDoor ()

function [] = closeDoor(obj, ~, ~)

obj.door\_state = "Closed";

obj.updateDisp();

if get(obj.door\_timer, 'Running') % Stop the timer and proceed

stop(obj.door\_timer);

end

obj.checkRoute(); Tcover1.1.7.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.7.1 |
| Coverage Item | Tcover1.1.7.1 |
| Input |  |
| State |  |
| Expected Output | ele1. door\_state ="Closed"; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.8: testupdateDisp ()

function [] = updateDisp(obj)

% Inner app disp update

obj.app.CurrentFloor.Value = mat2str(obj.current\_floor);

obj.app.Direction.Value = obj.direction;

obj.app.DoorState.Value = obj.door\_state;

% Should not check current floor, meaningless

if obj.direction == "Stationary"

switch obj.current\_floor

case 1

obj.app.Floor1Check.Value = false;

case 2

obj.app.Floor2Check.Value = false;

case 3

obj.app.Floor3Check.Value = false;

end

end

% Enable corresponding checkbox

switch obj.current\_floor

case 1

obj.app.Floor1Check.Enable = false;

obj.app.Floor2Check.Enable = true;

obj.app.Floor3Check.Enable = true;

case 2

obj.app.Floor1Check.Enable = true;

obj.app.Floor2Check.Enable = false;

obj.app.Floor3Check.Enable = true;

case 3

obj.app.Floor1Check.Enable = true;

obj.app.Floor2Check.Enable = true;

obj.app.Floor3Check.Enable = false;

end

% Model disp update

switch obj.current\_floor

case 1

obj.model.Image.Layout.Row = 3; % Upside down

if obj.door\_state == "Closed"

obj.model.Image.ImageSource = 'ele\_closed.jpg';

else

obj.model.Image.ImageSource = 'ele\_opened.jpg';

end

case 2

obj.model.Image.Layout.Row = 2; % Upside down

if obj.door\_state == "Closed"

obj.model.Image.ImageSource = 'ele\_closed.jpg';

else

obj.model.Image.ImageSource = 'ele\_opened.jpg';

end

case 3

obj.model.Image.Layout.Row = 1; % Upside down

if obj.door\_state == "Closed"

obj.model.Image.ImageSource = 'ele\_closed.jpg';

else

obj.model.Image.ImageSource = 'ele\_opened.jpg';

end

end

% Outer app update

obj.controller.updateOutDisp(); Tcover1.1.8.1-4

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.8.1 |
| Coverage Item | Tcover1.1.8.1 |
| Input |  |
| State | ele1.direction = "Stationary";  ele1.current\_floor = 1; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor1CheckValue = false;  Floor1CheckEnable = 'off';  Floor2CheckEnable = 'on';  Floor3CheckEnable = 'on';  LayoutRow = 3; |

|  |  |
| --- | --- |
|  | Test Case T1.1.8.2 |
| Coverage Item | Tcover1.1.8.2 |
| Input |  |
| State | ele1.direction = "Stationary";  ele1.current\_floor = 2; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor2CheckValue = false;  Floor1CheckEnable = 'on';  Floor2CheckEnable = 'off';  Floor3CheckEnable = 'on';  LayoutRow = 2; |

|  |  |
| --- | --- |
|  | Test Case T1.1.8.3 |
| Coverage Item | Tcover1.1.8.3 |
| Input |  |
| State | ele1.direction = "Stationary";  ele1.current\_floor = 3; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor3CheckValue = false;  Floor1CheckEnable = 'on';  Floor2CheckEnable = 'on';  Floor3CheckEnable = 'off';  LayoutRow = 1; |

|  |  |
| --- | --- |
|  | Test Case T1.1.8.4 |
| Coverage Item | Tcover1.1.8.4 |
| Input |  |
| State | ele1.direction = "Up";  ele1.current\_floor = 3; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor1CheckEnable = 'on';  Floor2CheckEnable = 'on';  Floor3CheckEnable = 'off';  LayoutRow = 1; |

* Test coverage: 4/4=100%
* Test result: 4 passed

T1.1.9: testaddDesFloor()

function [] = addDesFloor(obj, floor\_num)

obj.route(floor\_num) = 1;

obj.checkRoute(); Tcover1.1.9.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.9.1 |
| Coverage Item | Tcover1.1.9.1 |
| Input | 1 |
| State | ele1.current\_floor = 2;  ele1.pre\_direction = "Up"; |
| Expected Output | ele1.route(1)=1; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.10: testdelDesFloor ()

function [] = delDesFloor(obj, floor\_num)

obj.route(floor\_num) = 0;

obj.checkRoute(); Tcover1.1.10.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.10.1 |
| Coverage Item | Tcover1.1.10.1 |
| Input | 1 |
| State | ele1.current\_floor = 2;  ele1.pre\_direction = "Up"; |
| Expected Output | ele1.route(1)=0； |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.11: testcheckRoute ()

function [] = checkRoute(obj)

% If the outside panel of this floor is checked

if obj.route(obj.current\_floor) == 1

obj.delDesFloor(obj.current\_floor);

stop(obj.door\_timer);

obj.openDoor();

obj.door\_timer = timer('StartDelay', obj.door\_time, 'TimerFcn', @obj.closeDoor);

start(obj.door\_timer);

end

if obj.current\_floor == 1 % If currently located on the first floor,

% just check higher two floors

for i = 2:3

if obj.route(i) == 1

obj.goUp();

return

end

end

elseif obj.current\_floor == 3 % If currently located on the highest floor,

% just check lower two floors

for i = 1:2

if obj.route(i) == 1

obj.goDown();

return

end

end

else % obj.current\_floor == 2

if obj.pre\_direction == "Up"

for i = (obj.current\_floor+1):3

if(obj.route(i) == 1)

obj.goUp();

return

end

end

elseif obj.pre\_direction == "Down"

for i = 1:(obj.current\_floor-1)

if(obj.route(i) == 1)

obj.goDown();

return

end

end

end

end

% Naive search

for i = 1:3

if obj.route(i) == 1

if i > obj.current\_floor

obj.goUp();

return

elseif i < obj.current\_floor

obj.goDown();

return

end

end

end Tcover1.1.11.1-4

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.11.1 |
| Coverage Item | Tcover1.1.11.1 |
| Input |  |
| State | ele1.current\_floor = 1;  ele1.pre\_direction = "Up";  ele1.route(1) = 0;  ele1.route(2) = 1;  ele1.route(3) = 1; |
| Expected Output | ele1.direction="Up"； |

|  |  |
| --- | --- |
|  | Test Case T1.1.11.2 |
| Coverage Item | Tcover1.1.11.2 |
| Input |  |
| State | ele1.current\_floor = 3;  ele1.pre\_direction = "Up";  ele1.route(1) = 0;  ele1.route(2) = 1;  ele1.route(3) = 0; |
| Expected Output | ele1.direction="Down"； |

|  |  |
| --- | --- |
|  | Test Case T1.1.11.3 |
| Coverage Item | Tcover1.1.11.3 |
| Input |  |
| State | ele1.current\_floor = 2;  ele1.pre\_direction = "Up";  ele1.route(1) = 0;  ele1.route(2) = 0;  ele1.route(3) = 1; |
| Expected Output | ele1.direction="Up"； |

|  |  |
| --- | --- |
|  | Test Case T1.1.11.4 |
| Coverage Item | Tcover1.1.11.4 |
| Input |  |
| State | ele1.current\_floor = 2;  ele1.pre\_direction = "Down";  ele1.route(1) = 1;  ele1.route(2) = 0;  ele1.route(3) = 0; |
| Expected Output | ele1.direction="Down"； |

* Test coverage: 4/4=100%
* Test result: 4 passed

T1.1.12: testgoUp ()

function goUp(obj)

% NEVER move when door is opened

if(obj.getDoorState) ~= "Closed"

return;

end

% Shut the door when moving

obj.app.OpenDoor.Enable = false;

% Set app display

obj.pre\_direction = "Up";

obj.direction = "Up";

disp("GOUP!");

obj.updateDisp();

% Renew move timer

stop(obj.move\_timer);

obj.move\_timer = timer('StartDelay', obj.move\_time, 'TimerFcn', @obj.arrive);

start(obj.move\_timer); Tcover1.1.12.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.12.1 |
| Coverage Item | Tcover1.1.12.1 |
| Input |  |
| State |  |
| Expected Output | ele1.pre\_direction="Up"  ele1.direction="Up" |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.13: testgoDown ()

function goDown(obj)

% NEVER move when door is opened

if(obj.getDoorState) ~= "Closed"

return;

end

% Shut the door when moving

obj.app.OpenDoor.Enable = false;

% Set app display

obj.pre\_direction = "Down";

obj.direction = "Down";

disp("GODOWN!");

obj.updateDisp();

% Renew move timer

stop(obj.move\_timer);

obj.move\_timer = timer('StartDelay', obj.move\_time, 'TimerFcn', @obj.arrive);

start(obj.move\_timer); Tcover1.1.13.1

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.13.1 |
| Coverage Item | Tcover1.1.13.1 |
| Input |  |
| State |  |
| Expected Output | ele1.pre\_direction= "Down"  ele1.direction= "Down" |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.1.14: testarrive ()

function [] = arrive(obj, ~, ~)

stop(obj.move\_timer);

% Update the current\_floor according to the direction

if obj.direction == "Up"

obj.current\_floor = obj.current\_floor + 1;

obj.updateDisp();

elseif obj.direction == "Down"

obj.current\_floor = obj.current\_floor - 1;

obj.updateDisp();

end

% If one floor is unchecked during moving and other floors

% is not checked, then stop the ele on next floor

haveChoice = false;

for i = 1:3

if(obj.route(i) == 1)

haveChoice = true;

end

end

if(haveChoice == false)

obj.direction = "Stationary";

obj.updateDisp();

obj.checkRoute();

return;

end

% If current\_floor is checked, stop; Ohterwise, move

if obj.route(obj.current\_floor) == 1

% Enable open door if stationary

obj.app.OpenDoor.Enable = true;

% Clear current floor from the route

obj.route(obj.current\_floor) = 0;

obj.direction = "Stationary";

obj.openDoor();

% Renew door timer

stop(obj.door\_timer);

obj.door\_timer = timer('StartDelay', obj.door\_time, 'TimerFcn', @obj.closeDoor);

start(obj.door\_timer);

else

obj.checkRoute();

end Tcover1.1.14.1-3

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.14.1 |
| Coverage Item | Tcover1.1.14.1 |
| Input |  |
| State | ele1.direction = "Up";  ele1.current\_floor = 1;  ele1.route(1) = 0;  ele1.route(2) = 1;  ele1.route(3) = 1; |
| Expected Output | ele1.current\_floor=2  ele1.direction="Stationary" |

|  |  |
| --- | --- |
|  | Test Case T1.1.14.2 |
| Coverage Item | Tcover1.1.14.2 |
| Input |  |
| State | ele1.direction = "Up";  ele1.current\_floor = 1;  ele1.route(1) = 1;  ele1.route(2) = 0;  ele1.route(3) = 1; |
| Expected Output | ele1.current\_floor=2  ele1.direction= "Up" |

|  |  |
| --- | --- |
|  | Test Case T1.1.14.3 |
| Coverage Item | Tcover1.1.14.3 |
| Input |  |
| State | ele1.direction = "Down";  ele1.current\_floor = 2;  ele1.route(1) = 0;  ele1.route(2) = 0;  ele1.route(3) = 0; |
| Expected Output | ele1.current\_floor= 1  ele1.direction= "Stationary" |

* Test coverage: 3/3=100%
* Test result: 3 passed

### T1.2: OutsidePanel Unit Test

T1.2.1: testOutsidePanel ()

function obj = OutsidePanel()

obj.ele1\_floor = 1;

obj.ele1\_dir = "Stationary";

obj.ele2\_floor = 1;

obj.ele2\_dir = "Stationary"; Tcover1.2.1.1

end

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.1.1 |
| Coverage Item | Tcover1.2.1.1 |
| Input |  |
| State | OP = OutsidePanel; |
| Expected Output | ele1\_floor = 1;  ele1\_dir = "Stationary";  ele2\_floor = 1;  ele2\_dir = "Stationary"; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.2.2: testOutsidePanel ()

function obj = OutsidePanel()

obj.ele1\_floor = 1;

obj.ele1\_dir = "Stationary";

obj.ele2\_floor = 1;

obj.ele2\_dir = "Stationary"; Tcover1.2.2.1

end

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.2.1 |
| Coverage Item | Tcover1.2.2.1 |
| Input |  |
| State | OP = OutsidePanel; |
| Expected Output | ele1\_floor = 1;  ele1\_dir = "Stationary";  ele2\_floor = 1;  ele2\_dir = "Stationary"; |

* Test coverage: 1/1=100%
* Test result: 1 passed

T1.2.3: testupdateDisp ()

function [] = updateDisp(obj, ele1\_flr, ele1\_dir, ele2\_flr, ele2\_dir)

obj.app.Ele1Floor.Value = mat2str(ele1\_flr);

obj.app.Ele2Floor.Value = mat2str(ele2\_flr);

obj.app.Ele1Dir.Value = ele1\_dir;

obj.app.Ele2Dir.Value = ele2\_dir; Tcover1.2.3.1

end

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.2.3.1 |
| Coverage Item | Tcover1.2.3.1 |
| Input |  |
| State | ele1\_flr = 1;  ele2\_flr = 2;  ele1\_dir = "Up";  ele2\_dir = "Up"; |
| Expected Output | flr1.app.Ele1Floor.Value=cellstr('1')  flr1.app.Ele2Floor.Value=cellstr('2')  flr1.app.Ele1Dir.Value=cellstr(Ele1DirValue)  flr1.app.Ele2Dir.Value=cellstr(Ele2DirValue) |

* Test coverage: 1/1=100%
* Test result: 1 passed

### T1.3: ControlUnit Unit Test

T1.3.1: testgetEleFloor ()

function floor = getEleFloor(obj, ele\_num)

switch ele\_num

case 1

floor = obj.ele1.getCurrentFloor();

case 2

floor = obj.ele2.getCurrentFloor();

end Tcover1.3.1.1-2

end

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.3.1.1 |
| Coverage Item | Tcover1.3.1.1 |
| Input |  |
| State | ele\_num = 1;  flr = controller.ele1.getCurrentFloor(); |
| Expected Output | Floor=flr |

|  |  |
| --- | --- |
|  | Test Case T1.3.1.2 |
| Coverage Item | Tcover1.3.1.2 |
| Input |  |
| State | ele\_num = 2;  flr = controller.ele1.getCurrentFloor(); |
| Expected Output | Floor=flr |

* Test coverage: 2/2=100%
* Test result: 2 passed

T1.3.2: testdispatchEle ()

function floor = getEleFloor(obj, ele\_num)

% Ele1 currently stops at the floor, call ele1

if(obj.ele1.getCurrentFloor() == floor && obj.ele1.getDirection() == "Stationary")

obj.ele1.addDesFloor(floor);

return;

% Ele2 currently stops at floor, call ele2

elseif(obj.ele2.getCurrentFloor() == floor && obj.ele2.getDirection() == "Stationary")

obj.ele2.addDesFloor(floor);

return;

% Already in (at least) one ele's route, do nothing, just wait

% for the ele to arrive

elseif(obj.ele1.route(floor) == 1 || obj.ele2.route(floor) == 1)

return;

% ele1 is under the floor and it was going up, call ele1

elseif(obj.ele1.getCurrentFloor() < floor && obj.ele1.getPreDirection() == "Up")

obj.ele1.addDesFloor(floor);

return;

% ele2 is under the floor and it was going up, call ele2

elseif(obj.ele2.getCurrentFloor() < floor && obj.ele2.getPreDirection() == "Up")

obj.ele2.addDesFloor(floor);

return;

% ele1 is over the floor and it was going down, call ele1

elseif(obj.ele1.getCurrentFloor() > floor && obj.ele1.getPreDirection() == "Down")

obj.ele1.addDesFloor(floor);

return;

% ele2 is over the floor and it was going down, call ele2

elseif(obj.ele2.getCurrentFloor() > floor && obj.ele2.getPreDirection() == "Down")

obj.ele2.addDesFloor(floor);

return;

% choose the closer one

elseif((abs(obj.ele2.getCurrentFloor()-floor)<abs(obj.ele1.getCurrentFloor()-floor)) &&...

(obj.ele2.getPreDirection()~=direction))

obj.ele2.addDesFloor(floor);

return;

else

obj.ele1.addDesFloor(floor);

end Tcover1.3.2.1-2

end

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.3.2.1 |
| Coverage Item | Tcover1.3.2.1 |
| Input |  |
| State | floor = 2;  controller.ele1.current\_floor = 2;  controller.ele1.direction = "Stationary";  direction = "Up"; |
| Expected Output | controller.ele1.route(2)=0 |

|  |  |
| --- | --- |
|  | Test Case T1.3.2.2 |
| Coverage Item | Tcover1.3.2.2 |
| Input |  |
| State | floor = 2;  controller.ele2.current\_floor = 2;  controller.ele2.direction = "Stationary";  direction = "Up"; |
| Expected Output | controller.ele2.route(2)=0 |

|  |  |
| --- | --- |
|  | Test Case T1.3.2.3 |
| Coverage Item | Tcover1.3.2.3 |
| Input |  |
| State | floor = 3;  controller.ele1.current\_floor = 2;  controller.ele2.current\_floor = 1;  controller.ele1.route(floor) = 0;  controller.ele2.route(floor) = 0;  controller.ele1.pre\_direction = "Up"; |
| Expected Output | controller.ele1.route(3)=1 |

|  |  |
| --- | --- |
|  | Test Case T1.3.2.4 |
| Coverage Item | Tcover1.3.2.4 |
| Input |  |
| State | floor = 2;  controller.ele1.current\_floor = 3;  controller.ele2.current\_floor = 1;  controller.ele1.route(floor) = 0;  controller.ele2.route(floor) = 0;  controller.ele2.pre\_direction = "Up"; |
| Expected Output | controller.ele2.route(2)=1 |

|  |  |
| --- | --- |
|  | Test Case T1.3.2.5 |
| Coverage Item | Tcover1.3.2.5 |
| Input |  |
| State | floor = 2;  controller.ele1.current\_floor = 3;  controller.ele2.current\_floor = 1;  controller.ele1.route(floor) = 0;  controller.ele2.route(floor) = 0;  controller.ele1.pre\_direction = "Down";  controller.ele2.pre\_direction = "Down"; |
| Expected Output | controller.ele1.route(2)=1 |

|  |  |
| --- | --- |
|  | Test Case T1.3.2.6 |
| Coverage Item | Tcover1.3.2.6 |
| Input |  |
| State | floor = 2;  controller.ele1.current\_floor = 1;  controller.ele2.current\_floor = 3;  controller.ele1.route(floor) = 0;  controller.ele2.route(floor) = 0;  controller.ele1.pre\_direction = "Down";  controller.ele2.pre\_direction = "Down"; |
| Expected Output | controller.ele2.route(2)=1 |

|  |  |
| --- | --- |
|  | Test Case T1.3.2.7 |
| Coverage Item | Tcover1.3.2.7 |
| Input |  |
| State | floor = 1;  controller.ele1.current\_floor = 2;  controller.ele2.current\_floor = 3;  controller.ele1.route(floor) = 0;  controller.ele2.route(floor) = 0;  controller.ele1.pre\_direction = "Stationary";  controller.ele2.pre\_direction = "Stationary";  direction = "Up"; |
| Expected Output | controller.ele1.route(1)=1 |

|  |  |
| --- | --- |
|  | Test Case T1.3.2.8 |
| Coverage Item | Tcover1.3.2.8 |
| Input |  |
| State | floor = 1;  controller.ele1.current\_floor = 3;  controller.ele2.current\_floor = 3;  controller.ele1.route(floor) = 0;  controller.ele2.route(floor) = 0;  controller.ele1.pre\_direction = "Stationary";  controller.ele2.pre\_direction = "Stationary";  direction = "Up"; |
| Expected Output | controller.ele1.route(1)=1 |

* Test coverage: 8/8=100%
* Test result: 8 passed

## T2: Integration Test

### T2.1: UI + InsidePanel

T2.1.1: testupdateDisp ()

function [] = updateDisp(obj)

% Inner app disp update

obj.app.CurrentFloor.Value = mat2str(obj.current\_floor);

obj.app.Direction.Value = obj.direction;

obj.app.DoorState.Value = obj.door\_state;

% Should not check current floor, meaningless

if obj.direction == "Stationary"

switch obj.current\_floor

case 1

obj.app.Floor1Check.Value = false;

case 2

obj.app.Floor2Check.Value = false;

case 3

obj.app.Floor3Check.Value = false;

end

end

% Enable corresponding checkbox

switch obj.current\_floor

case 1

obj.app.Floor1Check.Enable = false;

obj.app.Floor2Check.Enable = true;

obj.app.Floor3Check.Enable = true;

case 2

obj.app.Floor1Check.Enable = true;

obj.app.Floor2Check.Enable = false;

obj.app.Floor3Check.Enable = true;

case 3

obj.app.Floor1Check.Enable = true;

obj.app.Floor2Check.Enable = true;

obj.app.Floor3Check.Enable = false;

end

% Model disp update

switch obj.current\_floor

case 1

obj.model.Image.Layout.Row = 3; % Upside down

if obj.door\_state == "Closed"

obj.model.Image.ImageSource = 'ele\_closed.jpg';

else

obj.model.Image.ImageSource = 'ele\_opened.jpg';

end

case 2

obj.model.Image.Layout.Row = 2; % Upside down

if obj.door\_state == "Closed"

obj.model.Image.ImageSource = 'ele\_closed.jpg';

else

obj.model.Image.ImageSource = 'ele\_opened.jpg';

end

case 3

obj.model.Image.Layout.Row = 1; % Upside down

if obj.door\_state == "Closed"

obj.model.Image.ImageSource = 'ele\_closed.jpg';

else

obj.model.Image.ImageSource = 'ele\_opened.jpg';

end

end

% Outer app update

obj.controller.updateOutDisp(); Tcover1.1.8.1-4

end

* Coverage Criteria:Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T1.1.8.1 |
| Coverage Item | Tcover1.1.8.1 |
| Input |  |
| State | ele1.direction = "Stationary";  ele1.current\_floor = 1; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor1CheckValue = false;  Floor1CheckEnable = 'off';  Floor2CheckEnable = 'on';  Floor3CheckEnable = 'on';  LayoutRow = 3; |

|  |  |
| --- | --- |
|  | Test Case T1.1.8.2 |
| Coverage Item | Tcover1.1.8.2 |
| Input |  |
| State | ele1.direction = "Stationary";  ele1.current\_floor = 2; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor2CheckValue = false;  Floor1CheckEnable = 'on';  Floor2CheckEnable = 'off';  Floor3CheckEnable = 'on';  LayoutRow = 2; |

|  |  |
| --- | --- |
|  | Test Case T1.1.8.3 |
| Coverage Item | Tcover1.1.8.3 |
| Input |  |
| State | ele1.direction = "Stationary";  ele1.current\_floor = 3; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor3CheckValue = false;  Floor1CheckEnable = 'on';  Floor2CheckEnable = 'on';  Floor3CheckEnable = 'off';  LayoutRow = 1; |

|  |  |
| --- | --- |
|  | Test Case T1.1.8.4 |
| Coverage Item | Tcover1.1.8.4 |
| Input |  |
| State | ele1.direction = "Up";  ele1.current\_floor = 3; |
| Expected Output | CurrentFloorValue = mat2str(ele1.current\_floor);  DirectionValue = ele1.direction;  DoorStateValue = ele1.door\_state;  Floor1CheckEnable = 'on';  Floor2CheckEnable = 'on';  Floor3CheckEnable = 'off';  LayoutRow = 1; |

* Test coverage: 4/4=100%
* Test result: 4 passed

## T3: Functional Test

T3.1: Use Case “Call elevator” (UI test)

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.1.1 |
| Coverage Item | Tcover3.1.1 |
| Input | floor1.UpButton |
| State |  |
| Expected Output | Elevator comes |

|  |  |
| --- | --- |
|  | Test Case T3.1.2 |
| Coverage Item | Tcover3.1.2 |
| Input | Floor2.DownButton |
| State |  |
| Expected Output | Elevator comes |

* Test coverage: 2/2=100%
* Test result: 2 passed

T3.2: Use Case “move” (UI test)

* Coverage Criteria: Branch coverage
* Test case

|  |  |
| --- | --- |
|  | Test Case T3.2.1 |
| Coverage Item | Tcover3.2.1 |
| Input | Elevator1. Floor3Check |
| State |  |
| Expected Output | move |

|  |  |
| --- | --- |
|  | Test Case T3.2.2 |
| Coverage Item | Tcover3.2.2 |
| Input | Elevator1. Floor3Check  Elevator1. Floor3Check |
| State |  |
| Expected Output | Chosen cancelled |

* Test coverage: 2/2=100%
* Test result: 2 passed

T3.3: Use Case “Control door” (UI test)

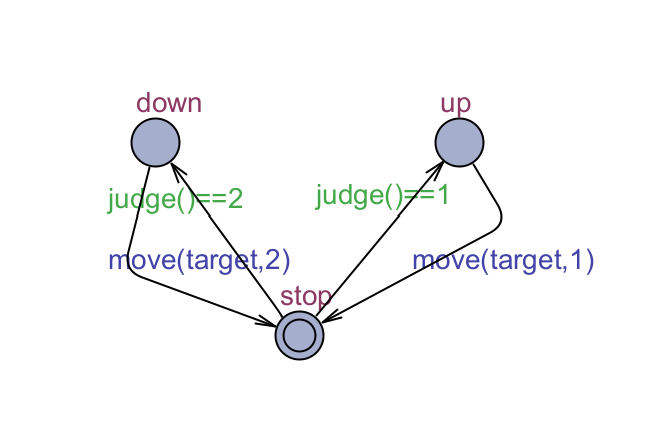
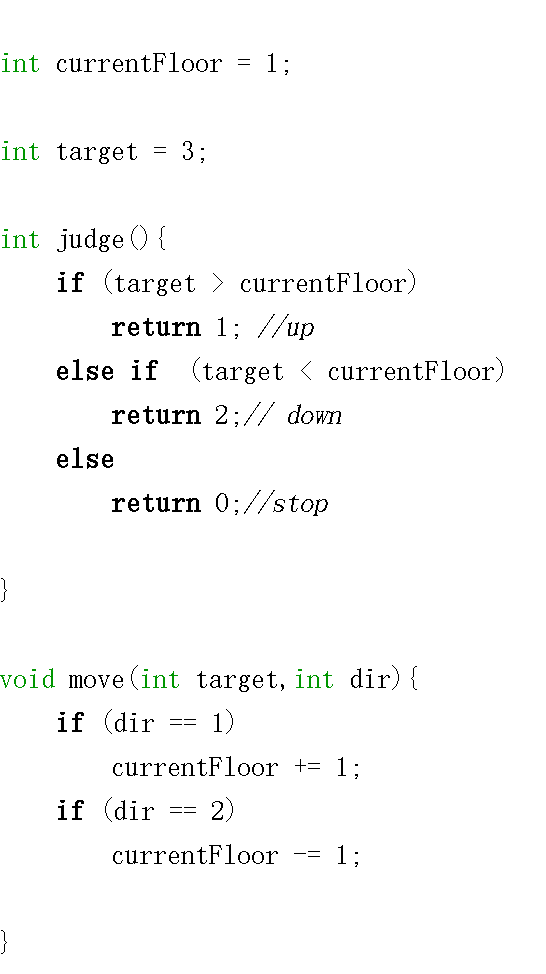
* Coverage Criteria: Branch coverage
* Test case

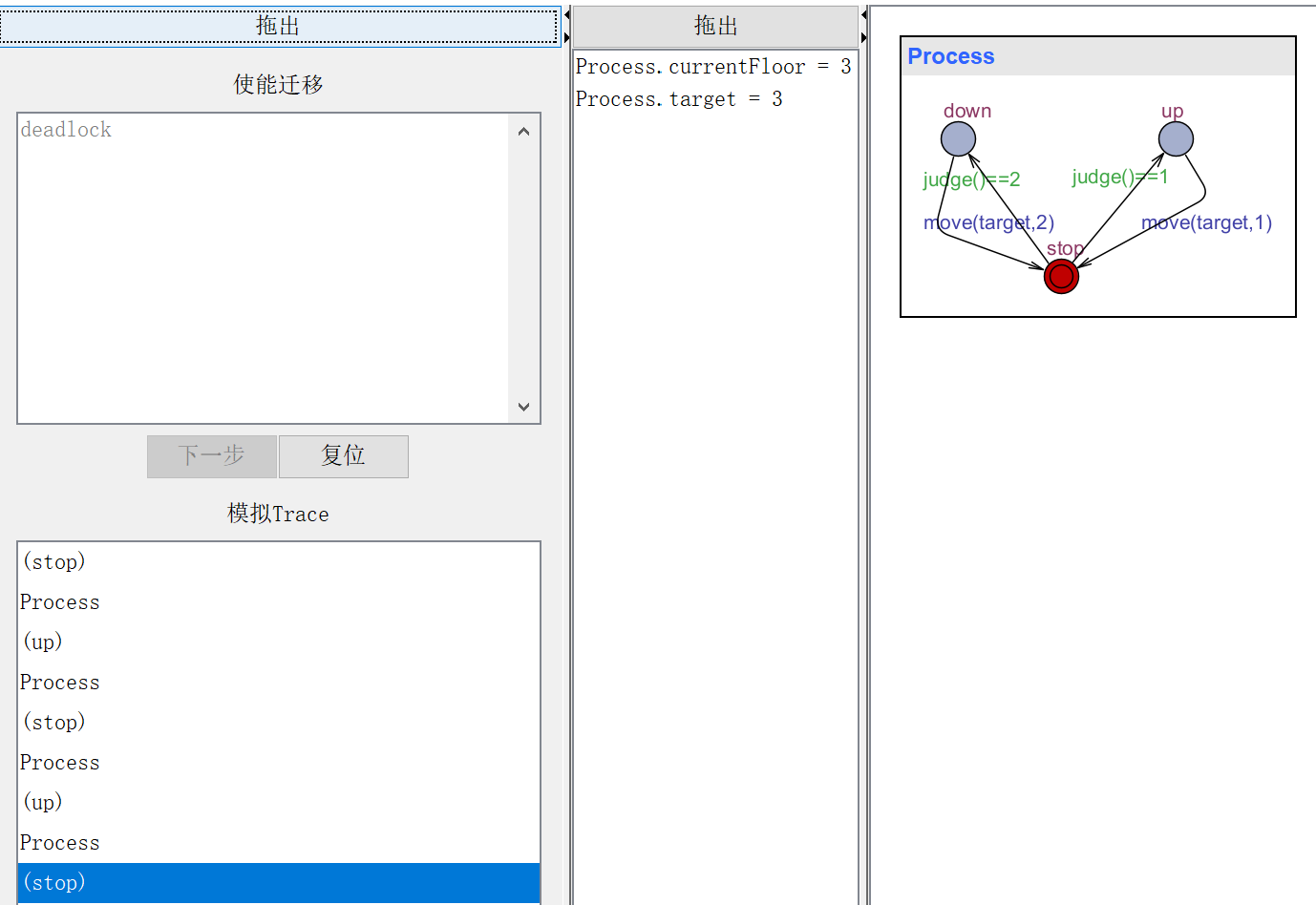
|  |  |
| --- | --- |
|  | Test Case T3.3.1 |
| Coverage Item | Tcover3.3.1 |
| Input | Elevator1. CloseDoor |
| State |  |
| Expected Output | Door close |

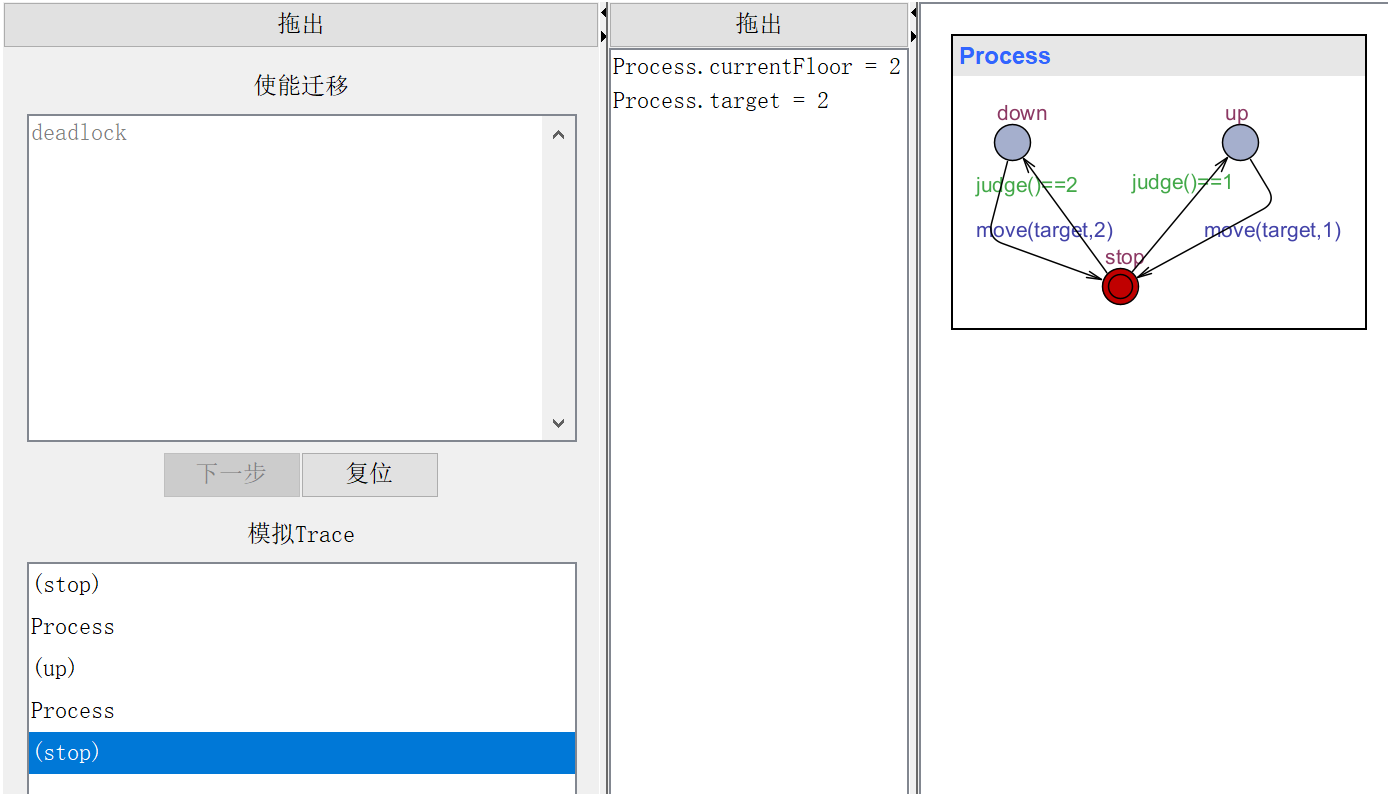
|  |  |
| --- | --- |
|  | Test Case T3.3.2 |
| Coverage Item | Tcover3.3.2 |
| Input | Elevator1. OpenDoor |
| State |  |
| Expected Output | Door open |

* Test coverage: 2/2=100%
* Test result: 2 passed

## T4: Model Checking

T4.1 Can always get to the target floor





proved!