고급 소프트웨어공학

ELEVATOR

MODEL CHECKER: SPIN

건국대학교

IT융합정보보호학과

신민용

smy11go@konkuk.ac.kr

안정현

aa9922@konkuk.ac.kr

KU KONKUK UNIVERSITY

Contents

Introduction

- 20층 건물의 6대의 엘리베이터

	저층용 엘리베이터	고층용 엘리베이터
개수	3대	3대
해당 엘리베이터	A,B,C	D,E,F
운행 층	1~10층	10~20층

- 룰(Rules)

- 반드시 1층에서 1~20층까지의 버튼 입력을 받음

- 각 6대의 아래 항목을 고려하여 엘리베이터 매칭

1. 엘리베이터의 목표 층수(goal_floor[elevator])
2. 엘리베이터 사용자의 현재 층수(user_floor)
3. 엘리베이터의 현재 층수(elevator_floor)

Introduction

- 상세사항

❖1층에서 2~20층까지의 버튼으로 입력받음 ->

1~9: 저층용(A/B/C), 10~20: 고층용(D/E/F) 매칭 ->

저층용/고층용 엘레베이터 중 가장 가까운 엘레베이터에 메시지 보냄

❖ 2~9, 11~20층에서 UP/DOWN 버튼으로 입력받음 ->

1. UP: 저층용/고층용 엘레베이터 중 '엘레베이터 목적지(눌린 버튼 중 최고층) > 사용자 현재 층 > 엘레베이터 현재 층'인 엘레베이터에 메시지 보냄

2. DOWN: 저층용/고층용 엘레베이터 중 '엘레베이터 목적지(눌린 버튼 중 최저층) < 사용자 현재 층 < 엘레베이터 현재 층'인 엘레베이터에 메시지 보냄

☞ '엘레베이터 목적지 > 사용자 현재 층 > 엘레베이터 현재 층'인 엘레베이터가 없으면, |엘레베이터 목적지 - 사용자 현재 층|이 가장 작은 엘레베이터에 메시지 보냄

Introduction

- 상세사항

❖10층에서 UP/DOWN 버튼으로 입력받음 ->

1. UP: 고층용 엘레베이터 중 |엘레베이터 목적지 - 사용자 현재 층|이 가장 작은 엘레베이터에 메시지 보냄
2. DOWN: 저층용 엘레베이터 중 |엘레베이터 목적지 - 사용자 현재 층|이 가장 작은 엘레베이터에 메시지 보냄

❖ 엘레베이터 A~F에서 1~20층까지의 버튼으로 입력받음 ->

엘레베이터 문 닫고 가던 방향으로 가면서 차례대로 문 열었다 닫음

☞ 다른 목적지 없으면 방금 눌린 목적지로 바로 향함

Introduction

- 상세사항

- lh(Low or High)!low/high, goal_user
- a_up_down!up/down, goal_user,

  b_up_down!up/down, goal_user,

  ...,

  f_up_down!up/down, goal_user
- a_chan!up/down/open/close, goal_user,

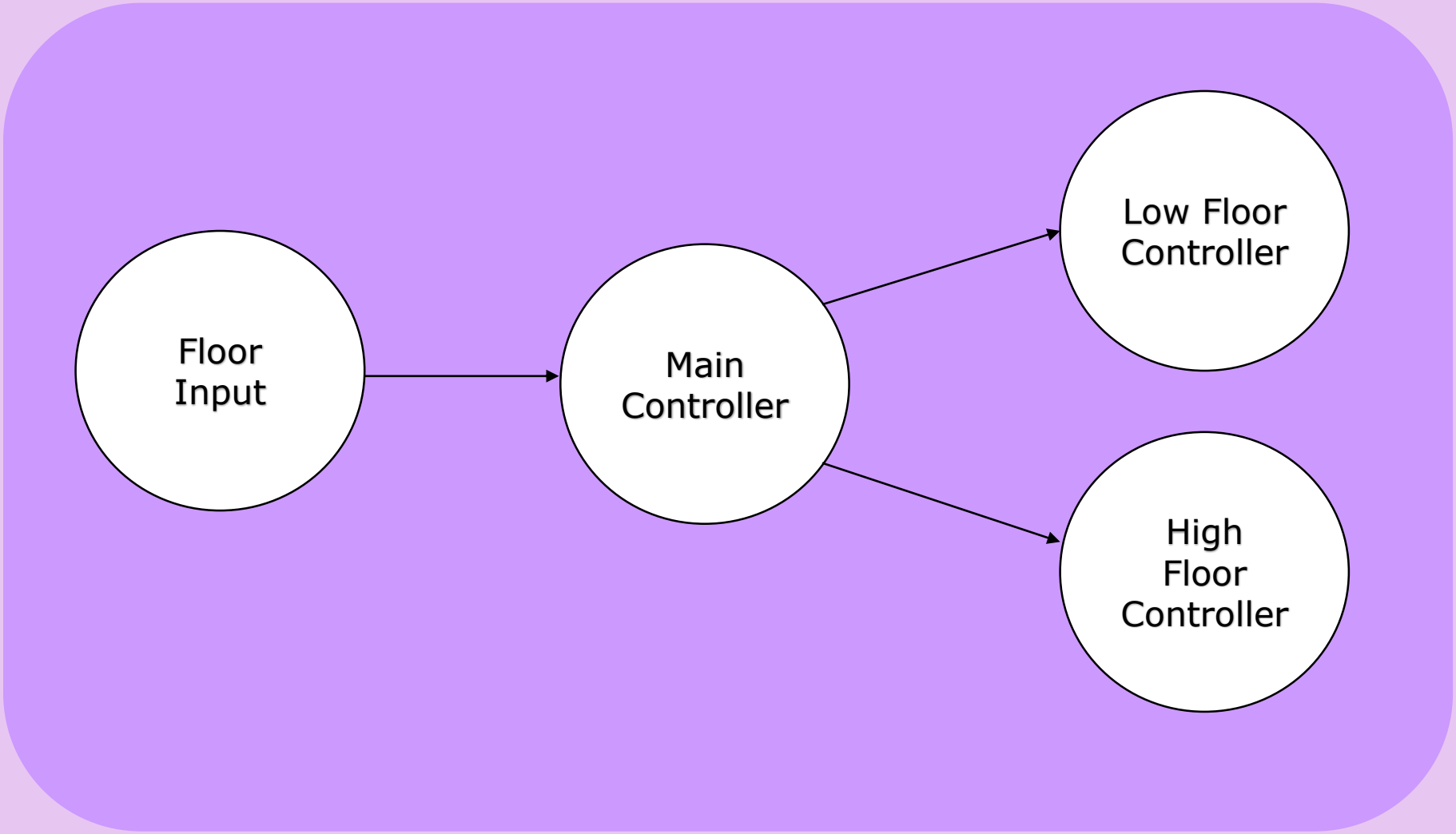
  b_chan!up/down/open/close, goal_user,

  ...,

  f_chan!up/down/open/close, goal_user

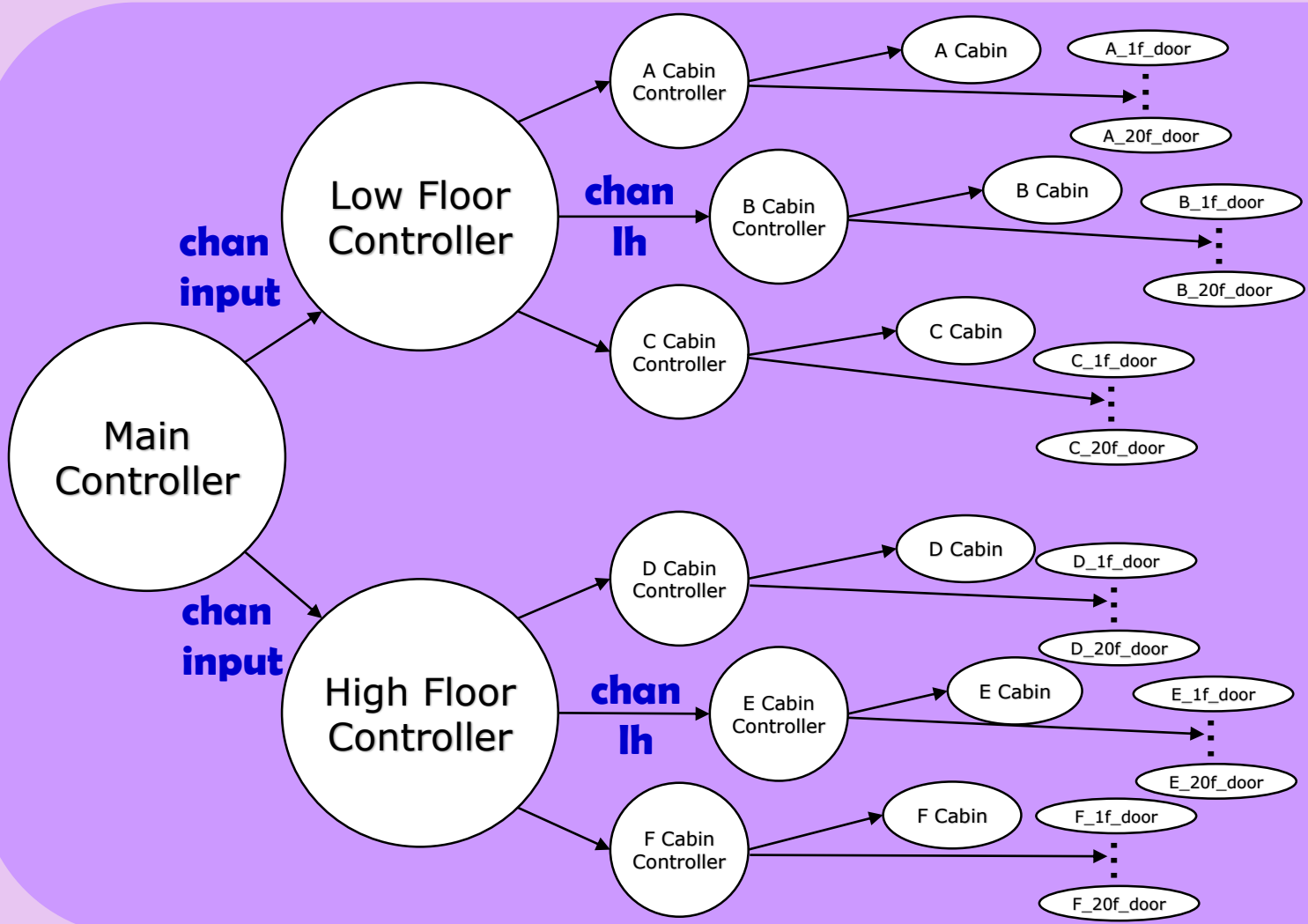
Introduction

- 메시징(Messaging)

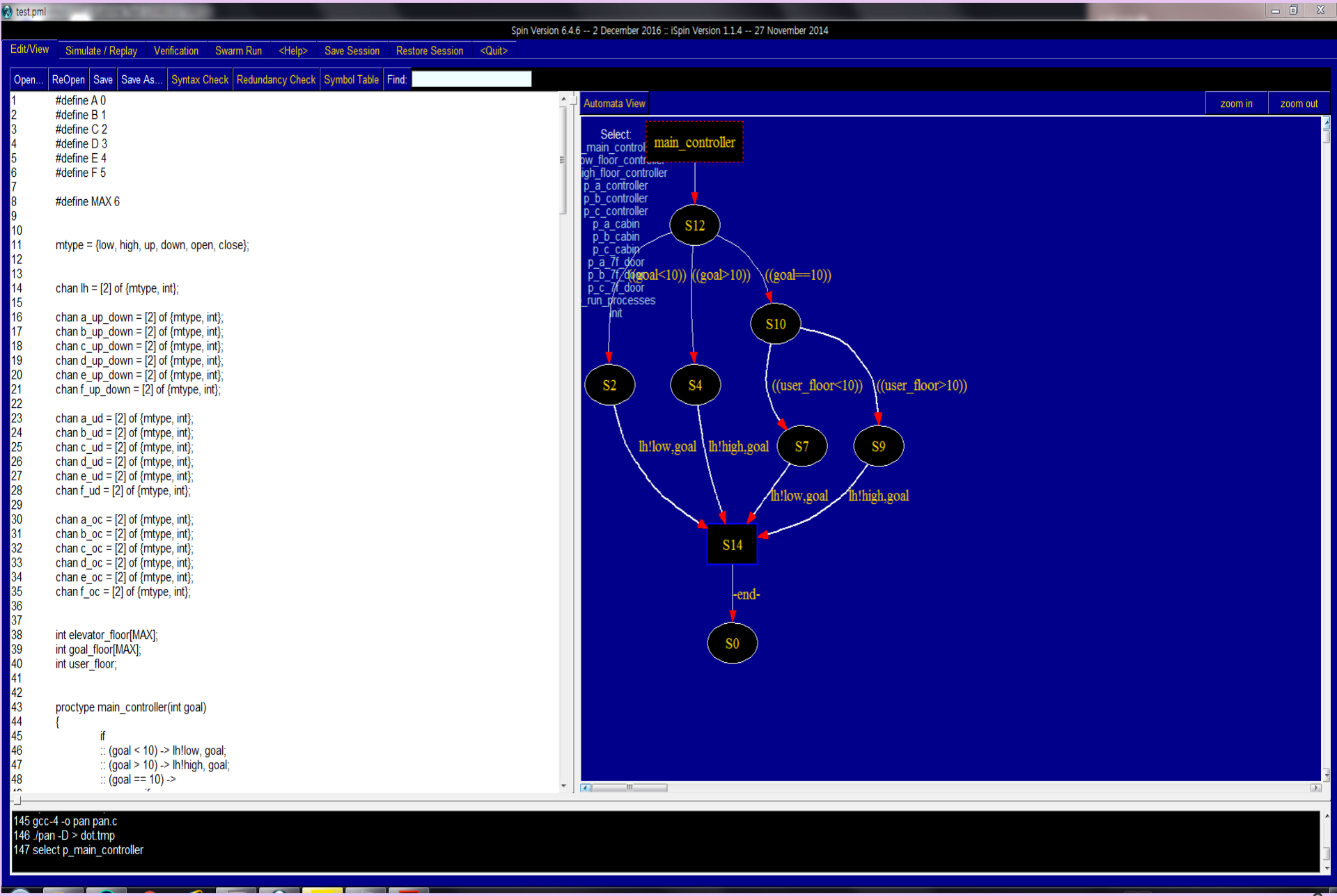


Introduction

- 메시징(Messaging)



Automata


test.pml
Spin Version 6.4.6 -- 2 December 2016 :: iSpin Version 1.1.4 -- 27 November 2014
Edit/View
Simulate / Replay
Verification
Swarm Run
<Help>
Save Session
Restore Session
<Quit>
Open...
ReOpen
Save
Save As...
Syntax Check
Redundancy Check
Symbol Table
Find:
1 #define A 0
2 #define B 1
3 #define C 2
4 #define D 3
5 #define E 4
6 #define F 5
7
8 #define MAX 6
9
10
11 mtype = {low, high, up, down, open, close};
12
13
14 chan lh = [2] of {mtype, int};
15
16 chan a_up_down = [2] of {mtype, int};
17 chan b_up_down = [2] of {mtype, int};
18 chan c_up_down = [2] of {mtype, int};
19 chan d_up_down = [2] of {mtype, int};
20 chan e_up_down = [2] of {mtype, int};
21 chan f_up_down = [2] of {mtype, int};
22
23 chan a_ud = [2] of {mtype, int};
24 chan b_ud = [2] of {mtype, int};
25 chan c_ud = [2] of {mtype, int};
26 chan d_ud = [2] of {mtype, int};
27 chan e_ud = [2] of {mtype, int};
28 chan f_ud = [2] of {mtype, int};
29
30 chan a_oc = [2] of {mtype, int};
31 chan b_oc = [2] of {mtype, int};
32 chan c_oc = [2] of {mtype, int};
33 chan d_oc = [2] of {mtype, int};
34 chan e_oc = [2] of {mtype, int};
35 chan f_oc = [2] of {mtype, int};
36
37
38 int elevator_floor[MAX];
39 int goal_floor[MAX];
40 int user_floor;
41
42
43 proctype main_controller(int goal)
44 {
45 if
46 :: (goal < 10) -> lh!low, goal;
47 :: (goal > 10) -> lh!high, goal;
48 :: (goal == 10) ->
Automata View
zoom in
zoom out
Select:
main_controller
p_a_controller
p_b_controller
p_c_controller
p_a_cabin
p_b_cabin
p_c_cabin
p_a_7f_door
p_b_7f_door
p_c_7f_door
run_processes
init
S12
((goal<10))
((goal>10))
((goal==10))
S10
S2
S4
((user_floor<10))
((user_floor>10))
lh!low,goal
lh!high,goal
S7
S9
lh!low,goal
lh!high,goal
S14
-end-
S0
145 gcc-4 -o pan pan.c
146 ./pan -D > dot.tmp
147 select p_main_controller

Automata

test.pml

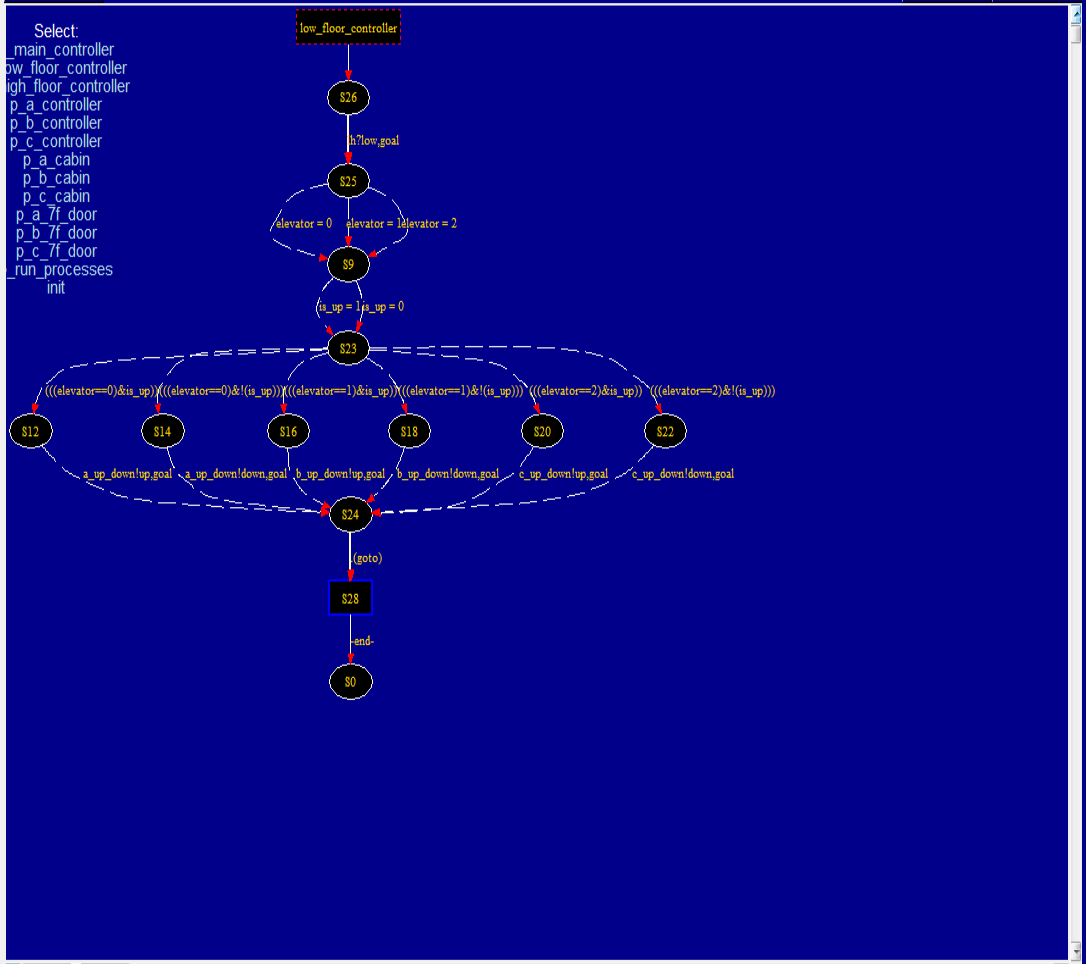
Spin Version 6.4.6 -- 2 December 2016 :: iSpin Version 1.1.4 -- 27 November 2014

Edit/View | Simulate / Replay | Verification | Swarm Run | <Help> | Save Session | Restore Session | <Quit>

Open... | ReOpen | Save | Save As... | Syntax Check | Redundancy Check | Symbol Table | Find:

```
15
16      chan a_up_down = [2] of {mtype, int};
17      chan b_up_down = [2] of {mtype, int};
18      chan c_up_down = [2] of {mtype, int};
19      chan d_up_down = [2] of {mtype, int};
20      chan e_up_down = [2] of {mtype, int};
21      chan f_up_down = [2] of {mtype, int};
22
23      chan a_ud = [2] of {mtype, int};
24      chan b_ud = [2] of {mtype, int};
25      chan c_ud = [2] of {mtype, int};
26      chan d_ud = [2] of {mtype, int};
27      chan e_ud = [2] of {mtype, int};
28      chan f_ud = [2] of {mtype, int};
29
30      chan a_oc = [2] of {mtype, int};
31      chan b_oc = [2] of {mtype, int};
32      chan c_oc = [2] of {mtype, int};
33      chan d_oc = [2] of {mtype, int};
34      chan e_oc = [2] of {mtype, int};
35      chan f_oc = [2] of {mtype, int};
36
37
38      int elevator_floor[MAX];
39      int goal_floor[MAX];
40      int user_floor;
41
42
43      proctype main_controller(int goal)
44      {
45              if
46              :: (goal < 10) -> lh!low, goal;
47              :: (goal > 10) -> lh!high, goal;
48              :: (goal == 10) ->
49                      if
50                      :: (user_floor < 10) -> lh!low, goal;
51                      :: (user_floor > 10) -> lh!high, goal;
52                      fi;
53              fi;
54      }
55
56      proctype low_floor_controller(int goal)
57      {
58              int distance = goal - user_floor;
59              int i = 0;
60              int elevator = -1;
61              bit is_up = 0;
62              int temp = -1;
```

Automata View | zoom in | zoom out



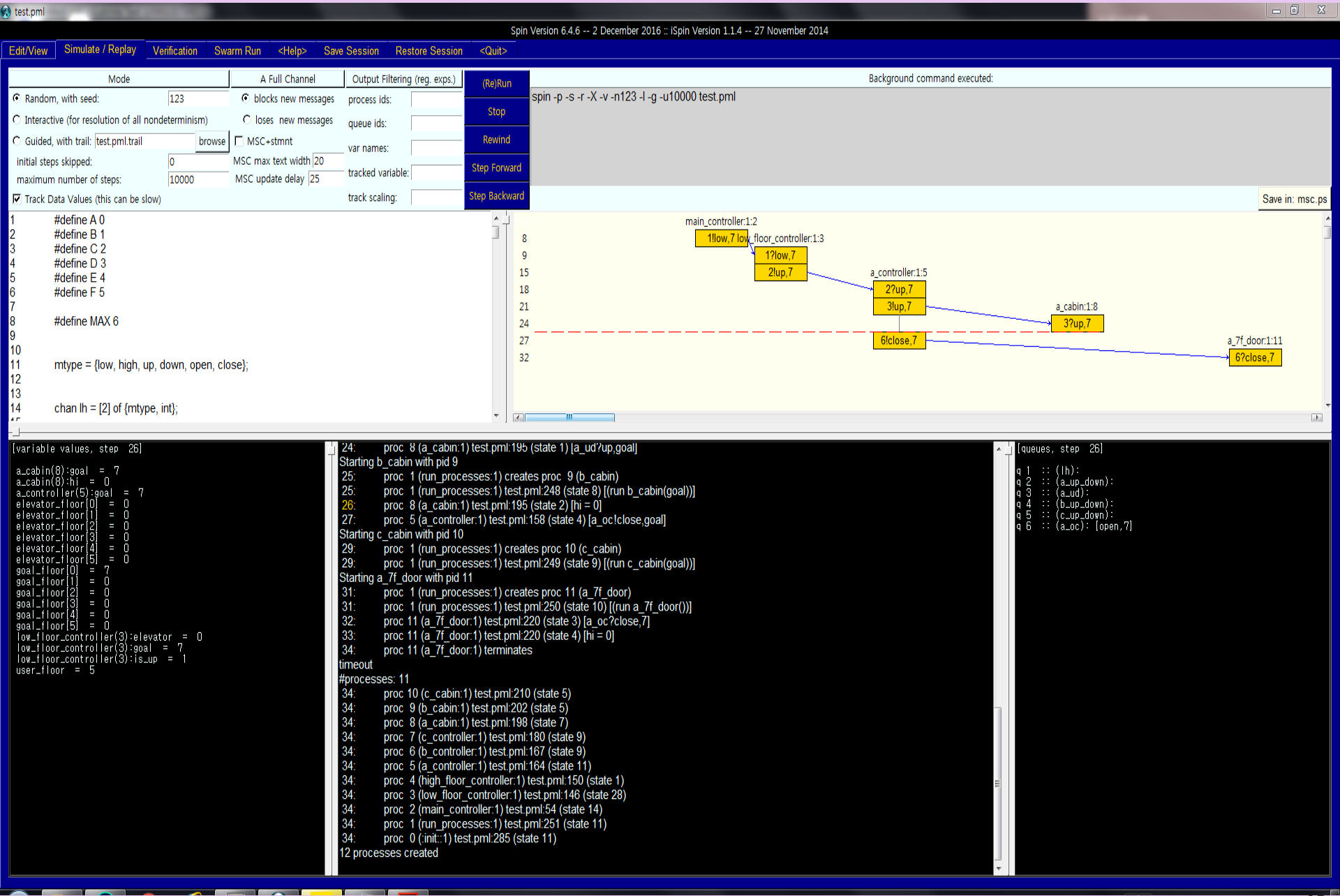

```
153 gcc-4 -o pan pan.c
154 ./pan -D > dot.tmp
155 select p_low_floor_controller
```

Error Source code

```
if
        :: (distance > 0) -> is_up = 1;
        :: (distance < 0) -> is_up = 0;
        fi;

        if
        :: (is_up) -> //up
           do
           :: (i > C) -> break;
           :: else ->
              if
              :: (!goal_floor[i]) ->
                 if //elevator_floor[i] < user_floor < goal_floor[i]
                 :: ((goal_floor[i] > user_floor) && (user_floor >
elevator_floor[i])) ->
                    elevator = i;
                    break;
                 fi;
              fi;
              i++;
           od;
        :: (!is_up) -> //down
           do
           :: (i > C) -> break;
           :: else ->
              if
              :: (!goal_floor[i]) ->
:: ((elevator_floor[i] > user_floor) && (user_floor > goal_floor[i])) ->
                    elevator = i;
                    break;
                 fi;
              fi;
              i++;
           od;
        fi;

        if //find a elevator near the user
        :: (elevator == -1) ->
           i = 0;
           distance = 20;
           do
           :: (i > C) -> break;
           :: else ->
              temp = goal_floor[i] - user_floor;
              if
              :: (temp < 0) -> temp = temp - (2*temp);
              fi;
              if
              :: (distance > temp) ->
                 distance = temp;
                 elevator = i;
              fi;
              i++;
           od;
        fi;
```

```
bit f_button[FLOOR_MAX];
bit e_button[ELEVATOR_MAX][FLOOR_MAX];

mtype = {button_press}
```

Automata


test.pml
Spin Version 6.4.6 -- 2 December 2016 :: iSpin Version 1.1.4 -- 27 November 2014
Edit/View
Simulate / Replay
Verification
Swarm Run
<Help>
Save Session
Restore Session
<Quit>
Mode
Random, with seed: 123
Interactive (for resolution of all nondeterminism)
Guided, with trail: test.pml.trail
browse
initial steps skipped: 0
maximum number of steps: 10000
Track Data Values (this can be slow)
A Full Channel
blocks new messages
loses new messages
MSC+stmnt
MSC max text width 20
MSC update delay 25
Output Filtering (reg. exps.)
process ids:
queue ids:
var names:
tracked variable:
track scaling:
(Re)Run
Stop
Rewind
Step Forward
Step Backward
Background command executed:
spin -p -s -r -X -v -n123 -l -g -u10000 test.pml
Save in: msc.ps
1 #define A 0
2 #define B 1
3 #define C 2
4 #define D 3
5 #define E 4
6 #define F 5
7
8 #define MAX 6
9
10
11 mtype = {low, high, up, down, open, close};
12
13
14 chan lh = [2] of {mtype, int};
main_controller:1:2
1!low,7
low_floor_controller:1:3
1?low,7
2!up,7
a_controller:1:5
2?up,7
3!up,7
a_cabin:1:8
3?up,7
6!close,7
a_7f_door:1:11
6?close,7
[variable values, step 26]
a_cabin(8):goal = 7
a_cabin(8):hi = 0
a_controller(5):goal = 7
elevator_floor[0] = 0
elevator_floor[1] = 0
elevator_floor[2] = 0
elevator_floor[3] = 0
elevator_floor[4] = 0
elevator_floor[5] = 0
goal_floor[0] = 7
goal_floor[1] = 0
goal_floor[2] = 0
goal_floor[3] = 0
goal_floor[4] = 0
goal_floor[5] = 0
low_floor_controller(3):elevator = 0
low_floor_controller(3):goal = 7
low_floor_controller(3):is_up = 1
user_floor = 5
24: proc 8 (a_cabin:1) test.pml:195 (state 1) [a_ud?up,goal]
Starting b_cabin with pid 9
25: proc 1 (run_processes:1) creates proc 9 (b_cabin)
25: proc 1 (run_processes:1) test.pml:248 (state 8) [(run b_cabin(goal))]
26: proc 8 (a_cabin:1) test.pml:195 (state 2) [hi = 0]
27: proc 5 (a_controller:1) test.pml:158 (state 4) [a_oc!close,goal]
Starting c_cabin with pid 10
29: proc 1 (run_processes:1) creates proc 10 (c_cabin)
29: proc 1 (run_processes:1) test.pml:249 (state 9) [(run c_cabin(goal))]
Starting a_7f_door with pid 11
31: proc 1 (run_processes:1) creates proc 11 (a_7f_door)
31: proc 1 (run_processes:1) test.pml:250 (state 10) [(run a_7f_door())]
32: proc 11 (a_7f_door:1) test.pml:220 (state 3) [a_oc?close,7]
33: proc 11 (a_7f_door:1) test.pml:220 (state 4) [hi = 0]
34: proc 11 (a_7f_door:1) terminates
timeout
#processes: 11
34: proc 10 (c_cabin:1) test.pml:210 (state 5)
34: proc 9 (b_cabin:1) test.pml:202 (state 5)
34: proc 8 (a_cabin:1) test.pml:198 (state 7)
34: proc 7 (c_controller:1) test.pml:180 (state 9)
34: proc 6 (b_controller:1) test.pml:167 (state 9)
34: proc 5 (a_controller:1) test.pml:164 (state 11)
34: proc 4 (high_floor_controller:1) test.pml:150 (state 1)
34: proc 3 (low_floor_controller:1) test.pml:146 (state 28)
34: proc 2 (main_controller:1) test.pml:54 (state 14)
34: proc 1 (run_processes:1) test.pml:251 (state 11)
34: proc 0 (:init::1) test.pml:285 (state 11)
12 processes created
[queues, step 26]
q 1 :: (lh):
q 2 :: (a_up_down):
q 3 :: (a_ud):
q 4 :: (b_up_down):
q 5 :: (c_up_down):
q 6 :: (a_oc): [open,7]

Property

1. 버튼이 눌리면 언젠가는 버튼의 불이 꺼진다.

- ltl p1 {([]main_controller:f_button==1)->
  (<>&main_controller:f_button==0)};

2. 층수가 입력이 되면 항상 저층부 혹은 고층부로 매칭된다.

- ltl p2 {((main_controller:goal=N)->lh!low) ||
  ((main_controller:goal=N)->lh!high)};

Thank you

Question & Answer