Smart Elevator Modeling with UPPAAL

정세진, 손준익

Contents

Introduction

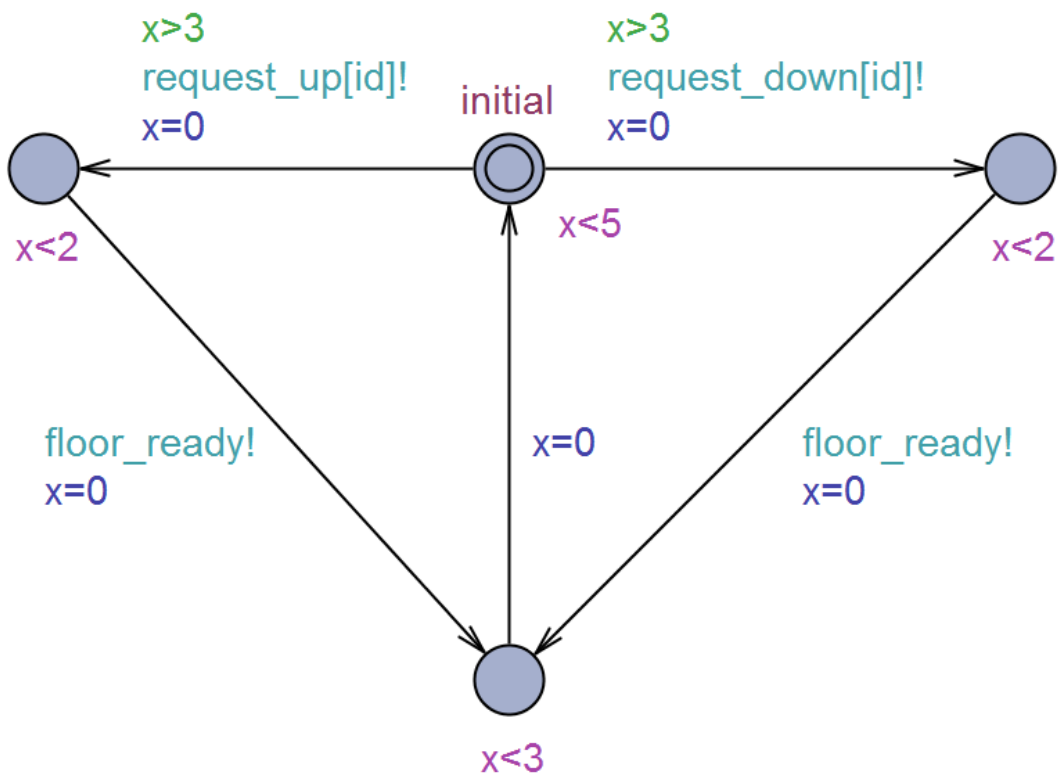
- 대상 : 스마트 엘리베이터

- 제약사항
  - 동작하는 엘리베이터는 2대로 구성
  - 건물은 총 4층으로 구성

- 모델
  - Cabin, Floor, Controller

Modeling with UPPAAL

- Automata – Floor



- request_up
  : 위층으로 이동 요청

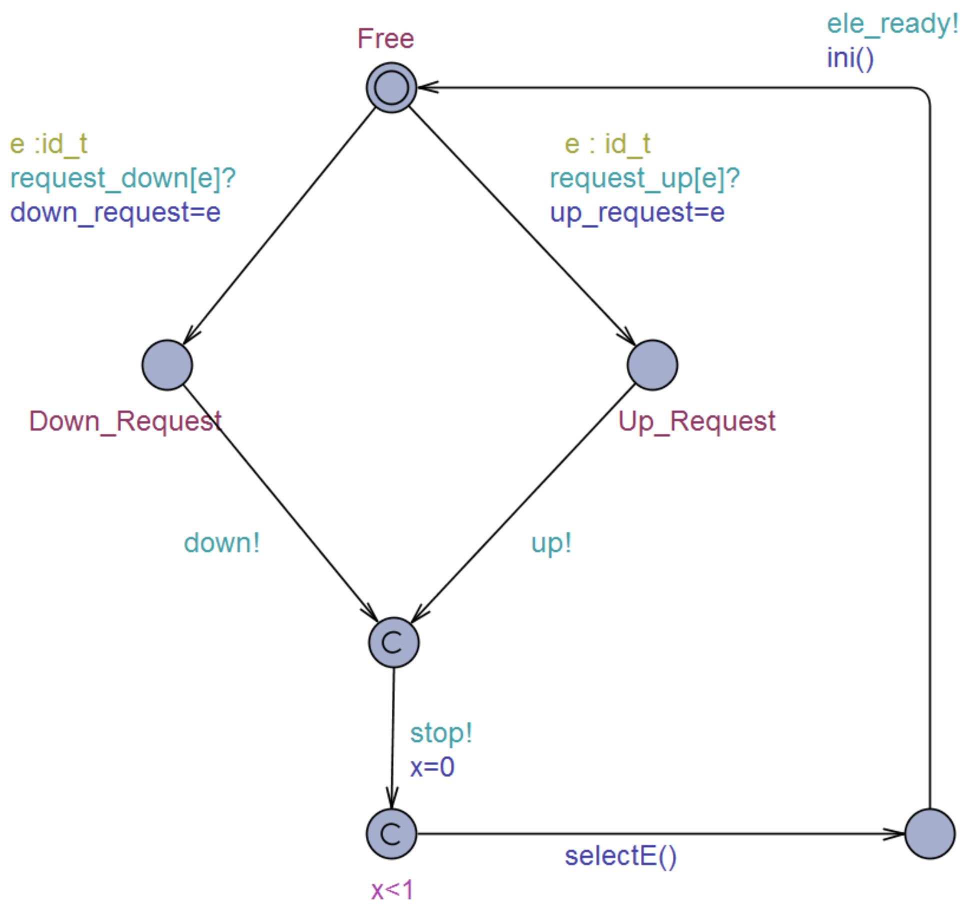
- request_down
  : 아래층으로 이동 요청

- id : Floor 모델의 현재 층수

- floor_ready
  : 선택된 엘리베이터와
  통신하는 채널

Modeling with UPPAAL

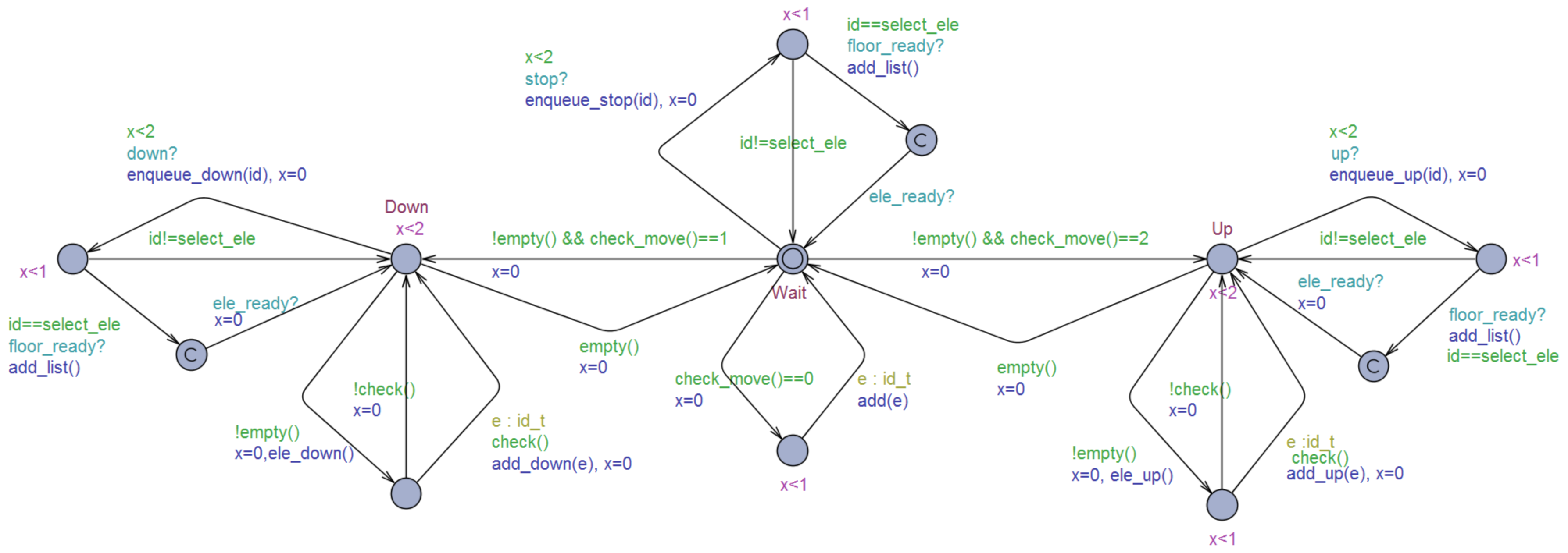
- Automata – Controller



- up_request
  : 위 층으로의 이동을 요청한 층
- up, down, stop
  : broad cast channel로 가능한 엘리베이터와 통신
- selectE()
  : 요청한 층과 가장 가까운 엘리베이터를 선택해주는 함수
- init()
  : 정보 초기화 하는 함수
- ele_ready
  : 선택된 엘리베이터와 통신하는 채널

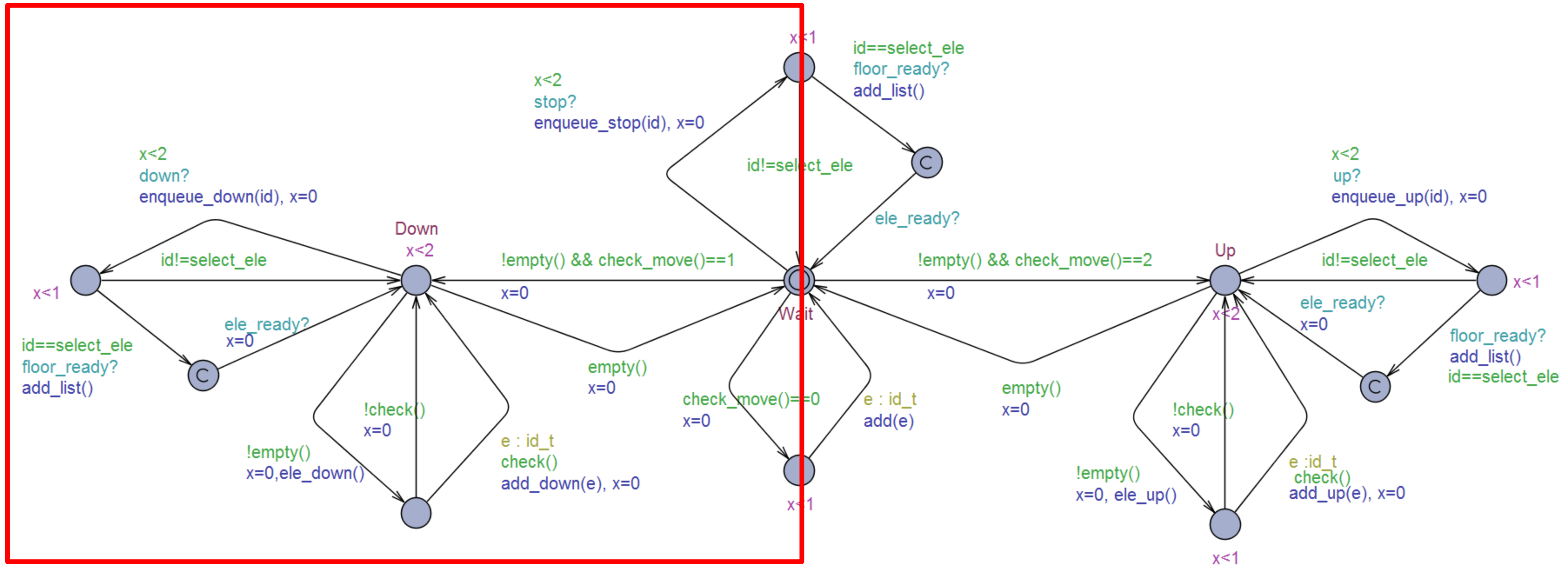
Modeling with UPPAAL

- Automata – Elevator



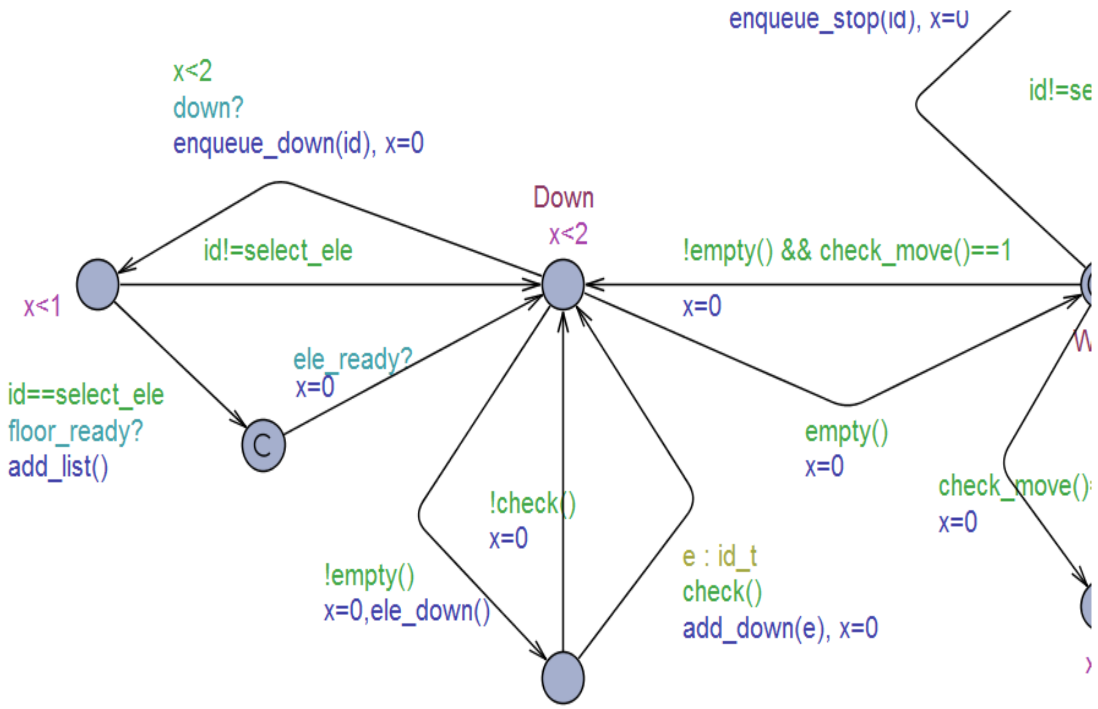
Modeling with UPPAAL

- Automata – Elevator



Modeling with UPPAAL

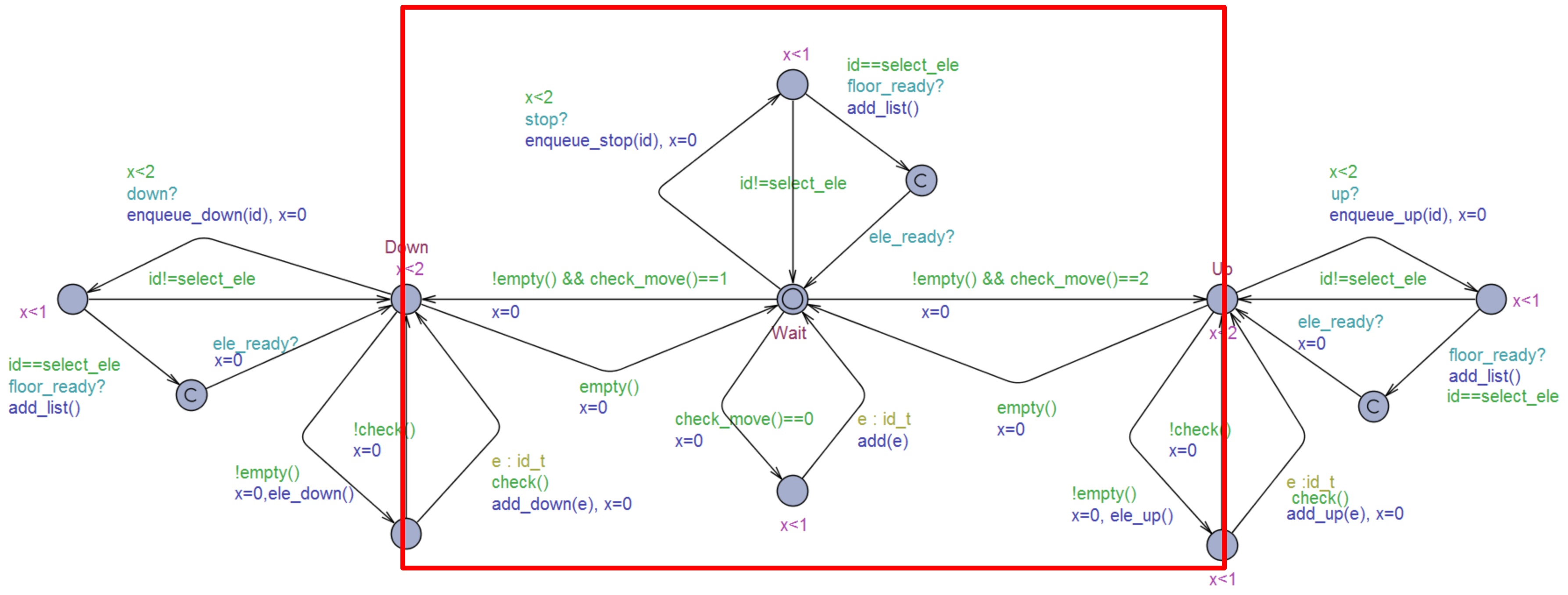
- Automata – Elevator Move Down



- empty()
  : 현재 가야하는 층이 있는지 확인
- check_move()
  : 가야하는 층이 현재 위치에서
  아래 층인지 위 층인지 확인
- check()
  : 현재 층이 요청이 들어온 층인지 확인
- add_list()
  : 요청 온 층을 엘리베이터의 버퍼에 추가
- ele_down()
  : 엘리베이터를 한층 씩 내리는 함수
- add_down()
  : 현재 층보다 아래층을 입력으로 받아
  엘리베이터의 버퍼에 추가

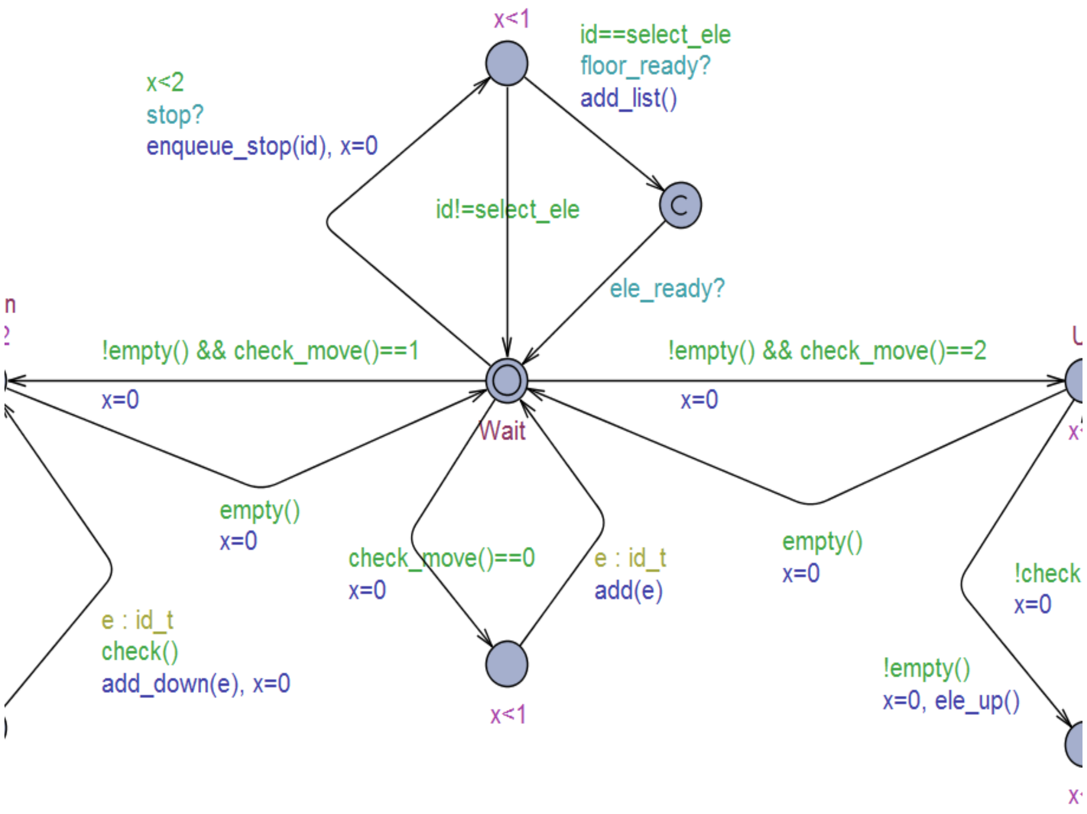
Modeling with UPPAAL

- Automata – Elevator



Modeling with UPPAAL

- Automata – Elevator Stop



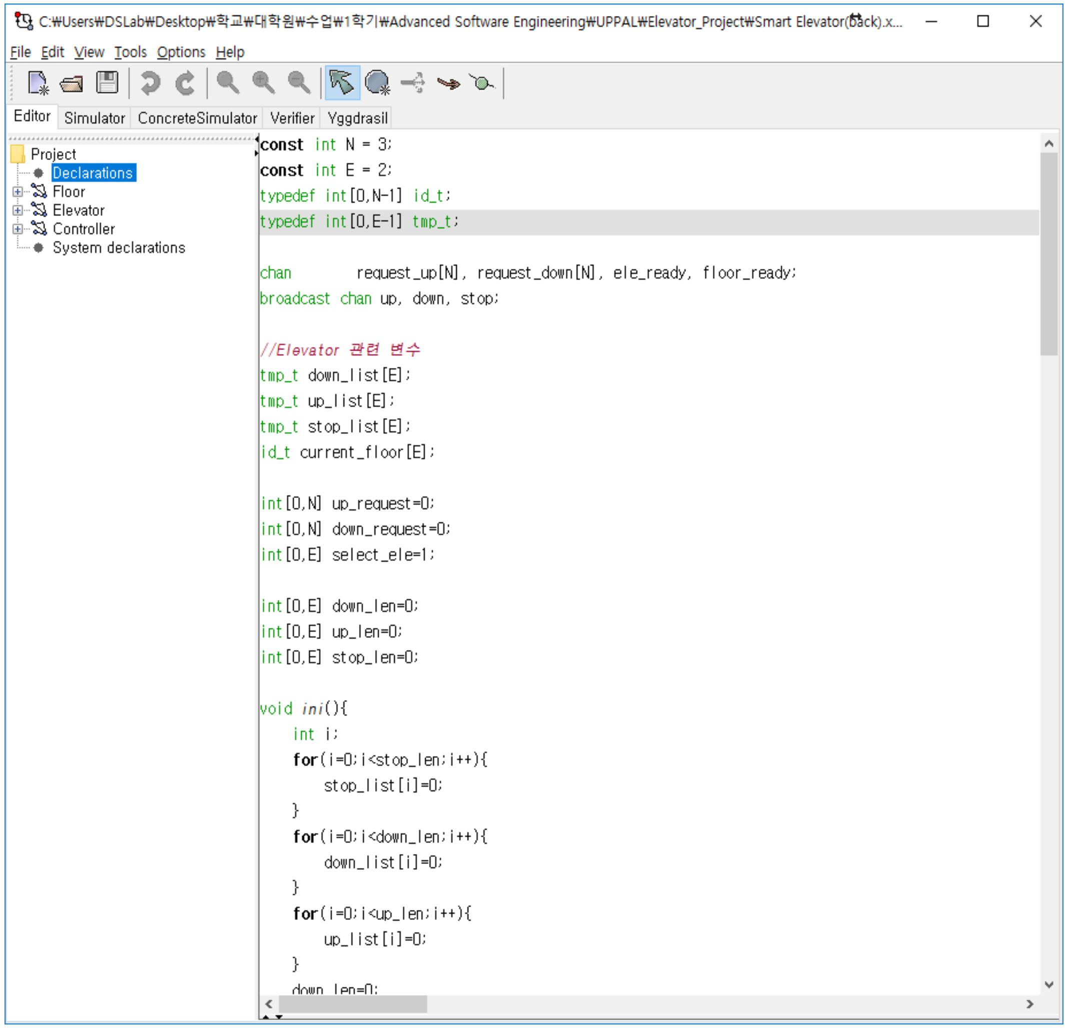
- empty()
  : 현재 가야하는 층이 있는지 확인

- check_move()
  : 가야하는 층이 현재 위치에서
  아래 층인지 위 층인지 확인

- add()
  : 요청 온 층을 엘리베이터의 버퍼에 추가

Modeling with UPPAAL

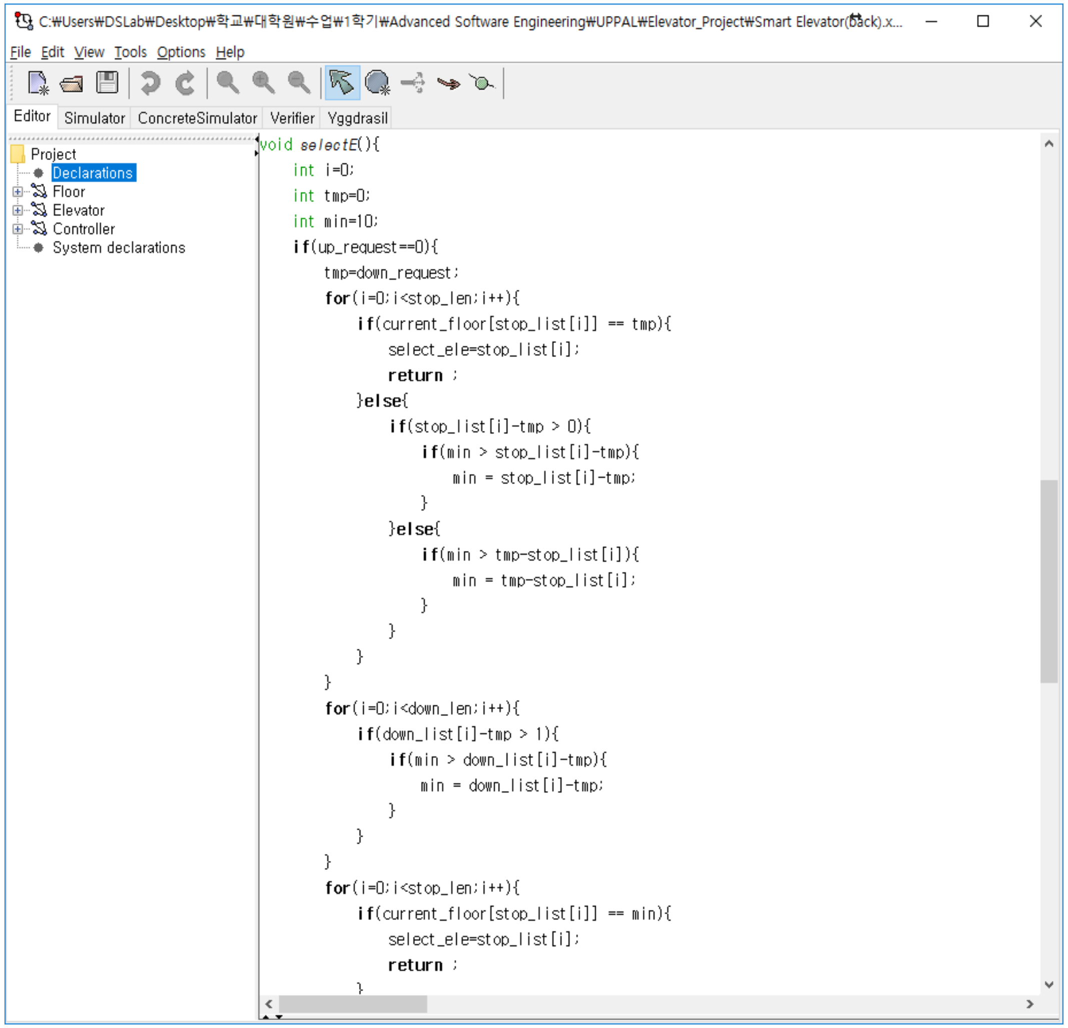
- Global Declarations


C:₩Users₩DSLab₩Desktop₩학교₩대학원₩수업₩1학기₩Advanced Software Engineering₩UPPAL₩Elevator_Project₩Smart Elevator(back).x...
File Edit View Tools Options Help
Editor Simulator ConcreteSimulator Verifier Yggdrasil
Project
Declarations
Floor
Elevator
Controller
System declarations

```
const int N = 3;
const int E = 2;
typedef int[0,N-1] id_t;
typedef int[0,E-1] tmp_t;

chan        request_up[N], request_down[N], ele_ready, floor_ready;
broadcast chan up, down, stop;

//Elevator 관련 변수
tmp_t down_list[E];
tmp_t up_list[E];
tmp_t stop_list[E];
id_t current_floor[E];

int[0,N] up_request=0;
int[0,N] down_request=0;
int[0,E] select_ele=1;

int[0,E] down_len=0;
int[0,E] up_len=0;
int[0,E] stop_len=0;

void ini(){
    int i;
    for(i=0;i<stop_len;i++){
        stop_list[i]=0;
    }
    for(i=0;i<down_len;i++){
        down_list[i]=0;
    }
    for(i=0;i<up_len;i++){
        up_list[i]=0;
    }
    down_len=0;
```

Modeling with UPPAAL

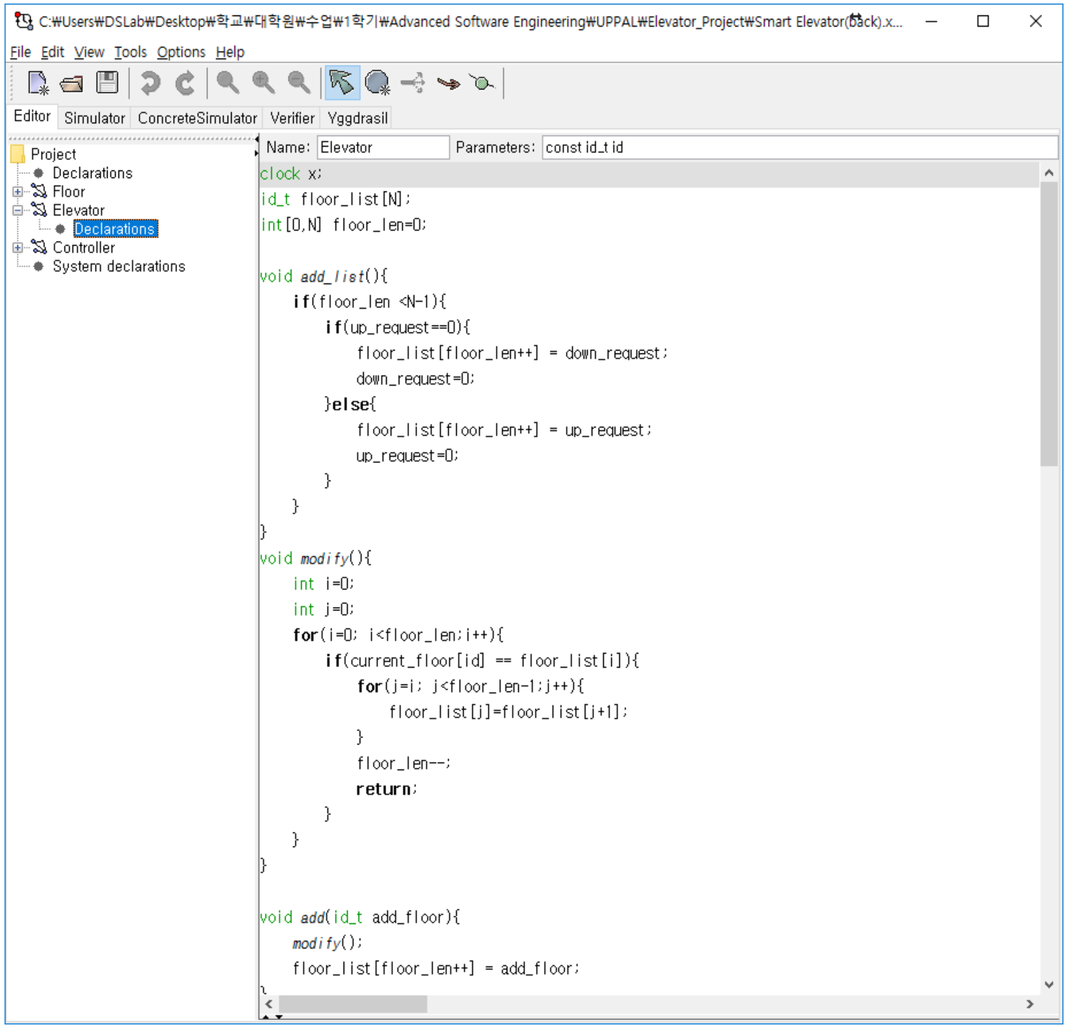
- Global Declarations


C:₩Users₩DSLab₩Desktop₩학교₩대학원₩수업₩1학기₩Advanced Software Engineering₩UPPAL₩Elevator_Project₩Smart Elevator(back).x...
File Edit View Tools Options Help
Editor Simulator ConcreteSimulator Verifier Yggdrasil
Project
Declarations
Floor
Elevator
Controller
System declarations

```
void selectE(){
    int i=0;
    int tmp=0;
    int min=10;
    if(up_request==0){
        tmp=down_request;
        for(i=0;i<stop_len;i++){
            if(current_floor[stop_list[i]] == tmp){
                select_ele=stop_list[i];
                return ;
            }else{
                if(stop_list[i]-tmp > 0){
                    if(min > stop_list[i]-tmp){
                        min = stop_list[i]-tmp;
                    }
                }else{
                    if(min > tmp-stop_list[i]){
                        min = tmp-stop_list[i];
                    }
                }
            }
        }
        for(i=0;i<down_len;i++){
            if(down_list[i]-tmp > 1){
                if(min > down_list[i]-tmp){
                    min = down_list[i]-tmp;
                }
            }
        }
        for(i=0;i<stop_len;i++){
            if(current_floor[stop_list[i]] == min){
                select_ele=stop_list[i];
                return ;
            }
```

Modeling with UPPAAL

- Elevator Declarations

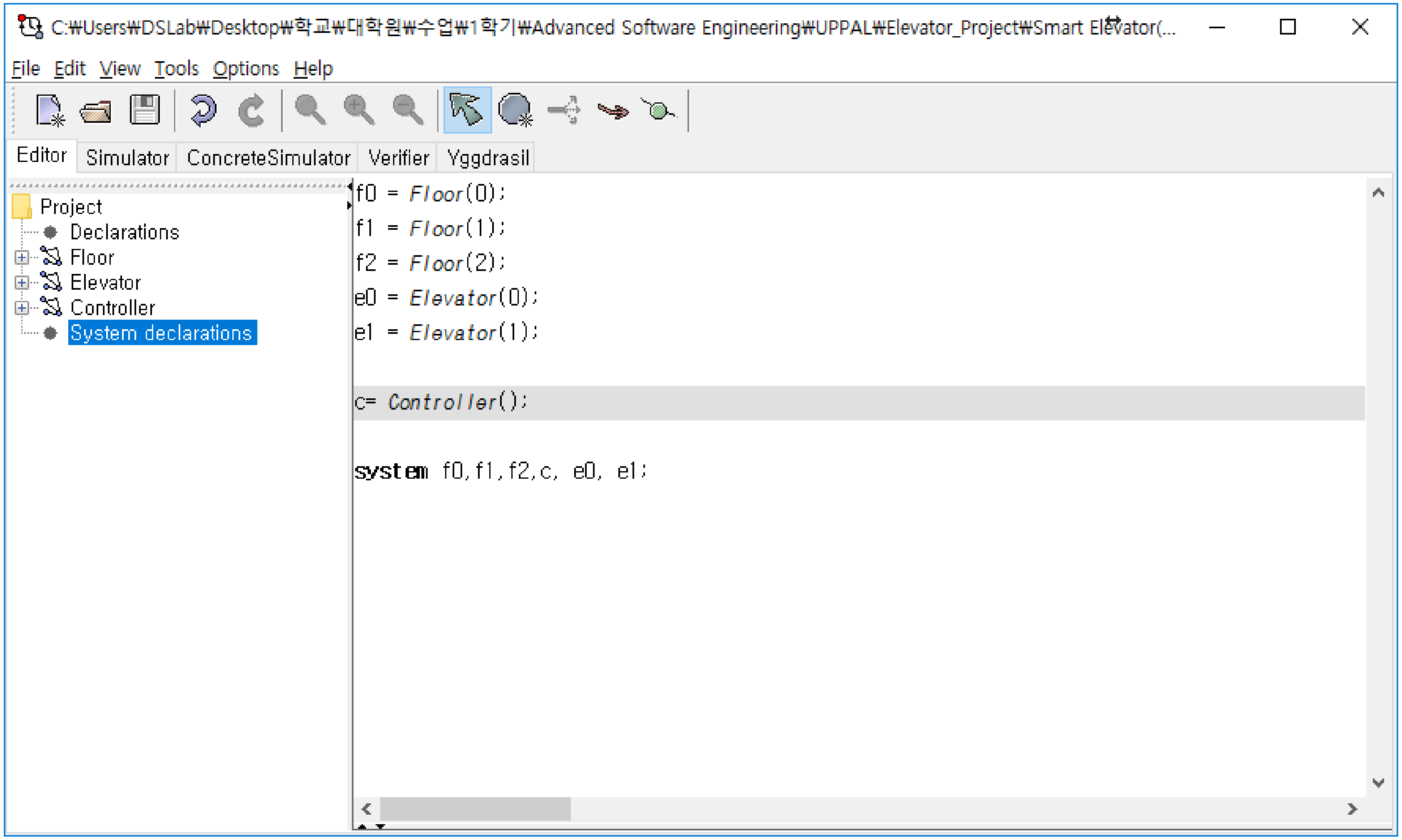

C:₩Users₩DSLab₩Desktop₩학교₩대학원₩수업₩1학기₩Advanced Software Engineering₩UPPAL₩Elevator_Project₩Smart Elevator(back).x...
File Edit View Tools Options Help
Editor Simulator ConcreteSimulator Verifier Yggdrasil
Project
Declarations
Floor
Elevator
Declarations
Controller
System declarations

Name: Elevator Parameters: const id_t id

```
clock x;
id_t floor_list[N];
int[0,N] floor_len=0;

void add_list(){
    if(floor_len <N-1){
        if(up_request==0){
            floor_list[floor_len++] = down_request;
            down_request=0;
        }else{
            floor_list[floor_len++] = up_request;
            up_request=0;
        }
    }
}
void modify(){
    int i=0;
    int j=0;
    for(i=0; i<floor_len;i++){
        if(current_floor[id] == floor_list[i]){
            for(j=i; j<floor_len-1;j++){
                floor_list[j]=floor_list[j+1];
            }
            floor_len--;
            return;
        }
    }
}

void add(id_t add_floor){
    modify();
    floor_list[floor_len++] = add_floor;
```

Modeling with UPPAAL

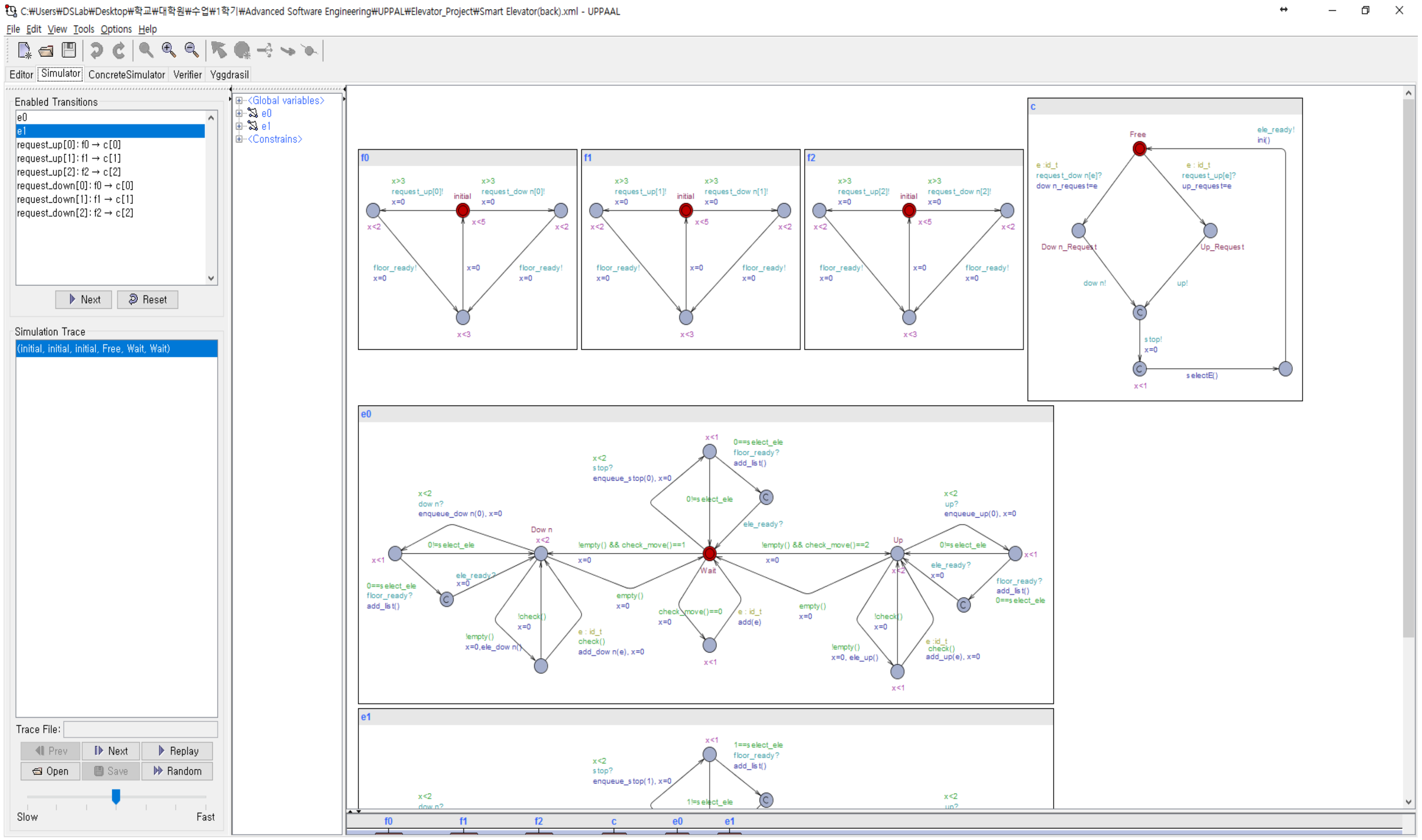
- System Declarations


C:₩Users₩DSLab₩Desktop₩학교₩대학원₩수업₩1학기₩Advanced Software Engineering₩UPPAL₩Elevator_Project₩Smart Elevator(...
File Edit View Tools Options Help
Editor Simulator ConcreteSimulator Verifier Yggdrasil
Project
Declarations
Floor
Elevator
Controller
System declarations

```
f0 = Floor(0);
f1 = Floor(1);
f2 = Floor(2);
e0 = Elevator(0);
e1 = Elevator(1);

c= Controller();

system f0,f1,f2,c, e0, e1;
```

Modeling with UPPAAL

- Simulator



Enabled Transitions:

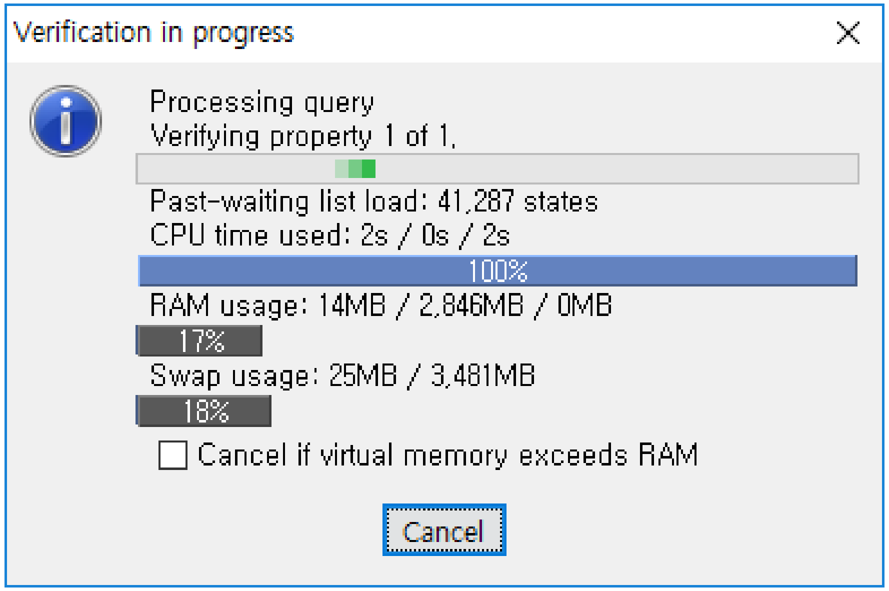
```
e0
e1
request_up[0]: f0 → c[0]
request_up[1]: f1 → c[1]
request_up[2]: f2 → c[2]
request_down[0]: f0 → c[0]
request_down[1]: f1 → c[1]
request_down[2]: f2 → c[2]
```

Simulation Trace:

```
(initial, initial, initial, Free, Wait, Wait)
```

Property

- Deadlock 확인
  - A[] not deadlock

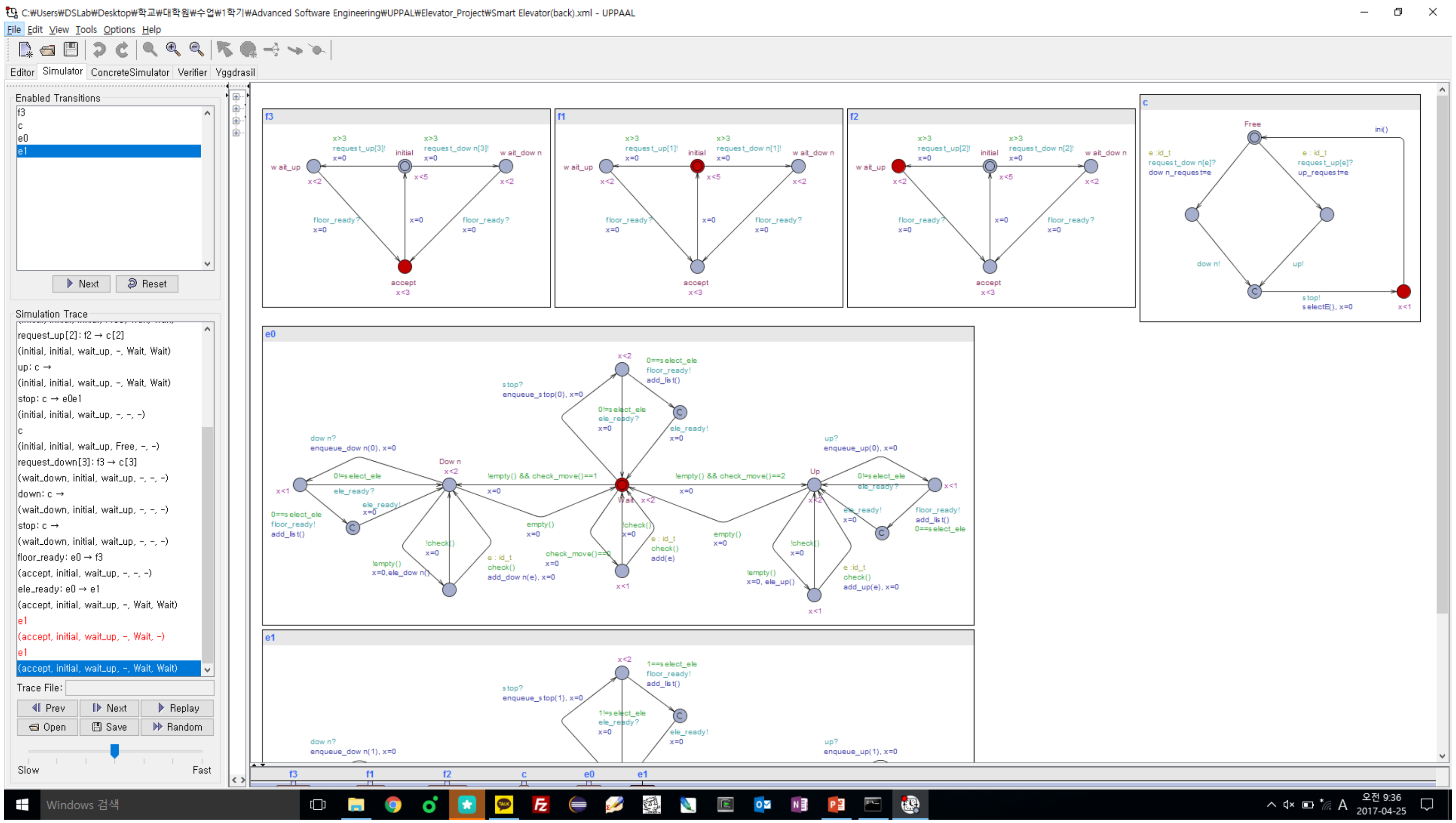


```
Verification in progress

Processing query
Verifying property 1 of 1.
Past-waiting list load: 41,287 states
CPU time used: 2s / 0s / 2s
100%
RAM usage: 14MB / 2,846MB / 0MB
17%
Swap usage: 25MB / 3,481MB
18%
```

☐ Cancel if virtual memory exceeds RAM

Property

- 각 층에서 요청이 있으면 요청이 accept 되어야 한다.
  - Floor.wait_up → Floor.accept


C:₩Users₩DSLab₩Desktop₩학교₩대학원₩수업₩1학기₩Advanced Software Engineering₩UPPAL₩Elevator_Project₩Smart Elevator(back).xml - UPPAAL
File Edit View Tools Options Help
Editor Simulator ConcreteSimulator Verifier Yggdrasil
Enabled Transitions
Next Reset
Simulation Trace
Trace File:
Prev Next Replay
Open Save Random
Slow Fast
f3 f1 f2 c e0 e1
wait_up initial wait_down accept
Free
Down Wait Up

Enabled Transitions:

```
f3
c
e0
e1
```

Simulation Trace:

```
request_up[2]: f2 → c[2]
(initial, initial, wait_up, -, Wait, Wait)
up: c →
(initial, initial, wait_up, -, Wait, Wait)
stop: c → e0e1
(initial, initial, wait_up, -, -, -)
c
(initial, initial, wait_up, Free, -, -)
request_down[3]: f3 → c[3]
(wait_down, initial, wait_up, -, -, -)
down: c →
(wait_down, initial, wait_up, -, -, -)
stop: c →
(wait_down, initial, wait_up, -, -, -)
floor_ready: e0 → f3
(accept, initial, wait_up, -, -, -)
ele_ready: e0 → e1
(accept, initial, wait_up, -, Wait, Wait)
e1
(accept, initial, wait_up, -, Wait, -)
e1
(accept, initial, wait_up, -, Wait, Wait)
```