

스마트 엘리베이터

- SMV 정형검증 -

HPE lab

남 현 우

2017.05.02

Contents

- 소개
 - 시스템 구성
 - 요구사항
- Automata
- Property
- 검증 결과

스마트 엘리베이터?

- 예) NHN 빌딩

- 키오스크 단말기

- 승객은 키오스크에서 목표 층 입력
- 최적 탑승 엘리베이터로 안내

- 승강기

- 열림/닫힘/비상벨 버튼
- 현재 층 디스플레이
- 이외 버튼은 없음



그림1. 목표 층 입력



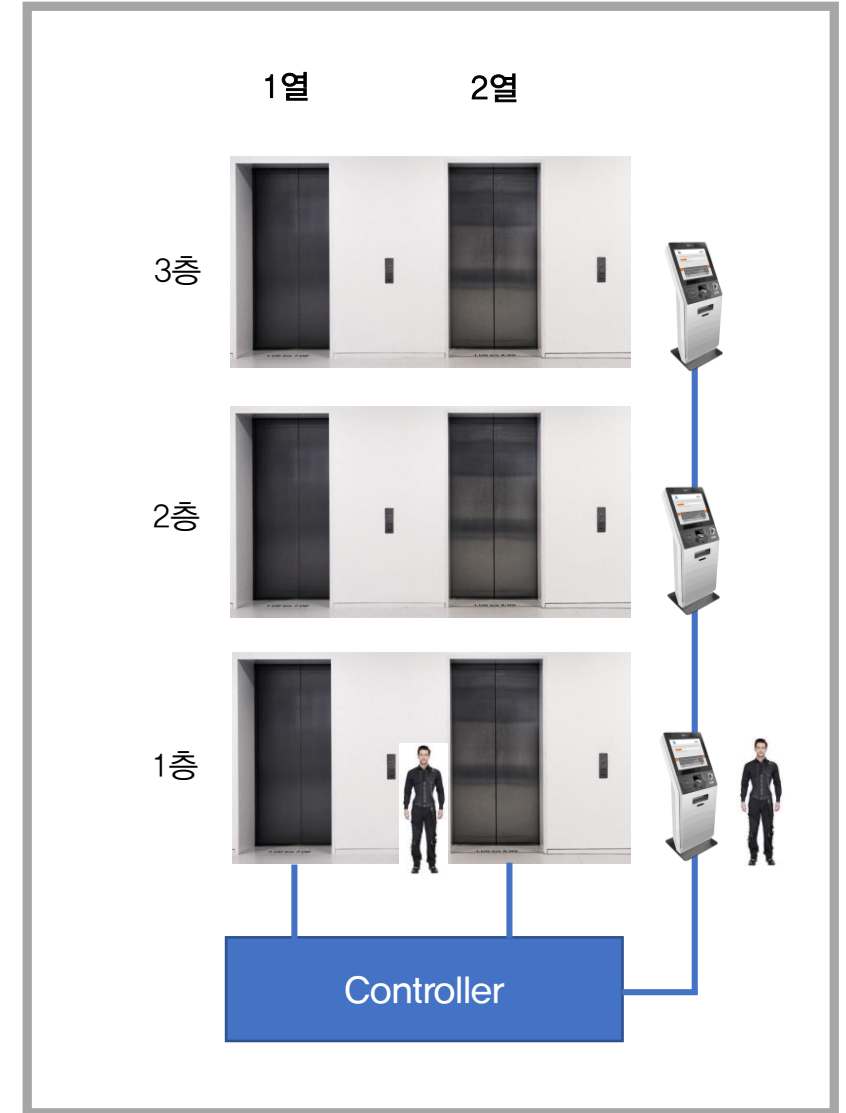
그림2. 탑승 엘리베이터 안내



그림3. 승강기 내부 버튼

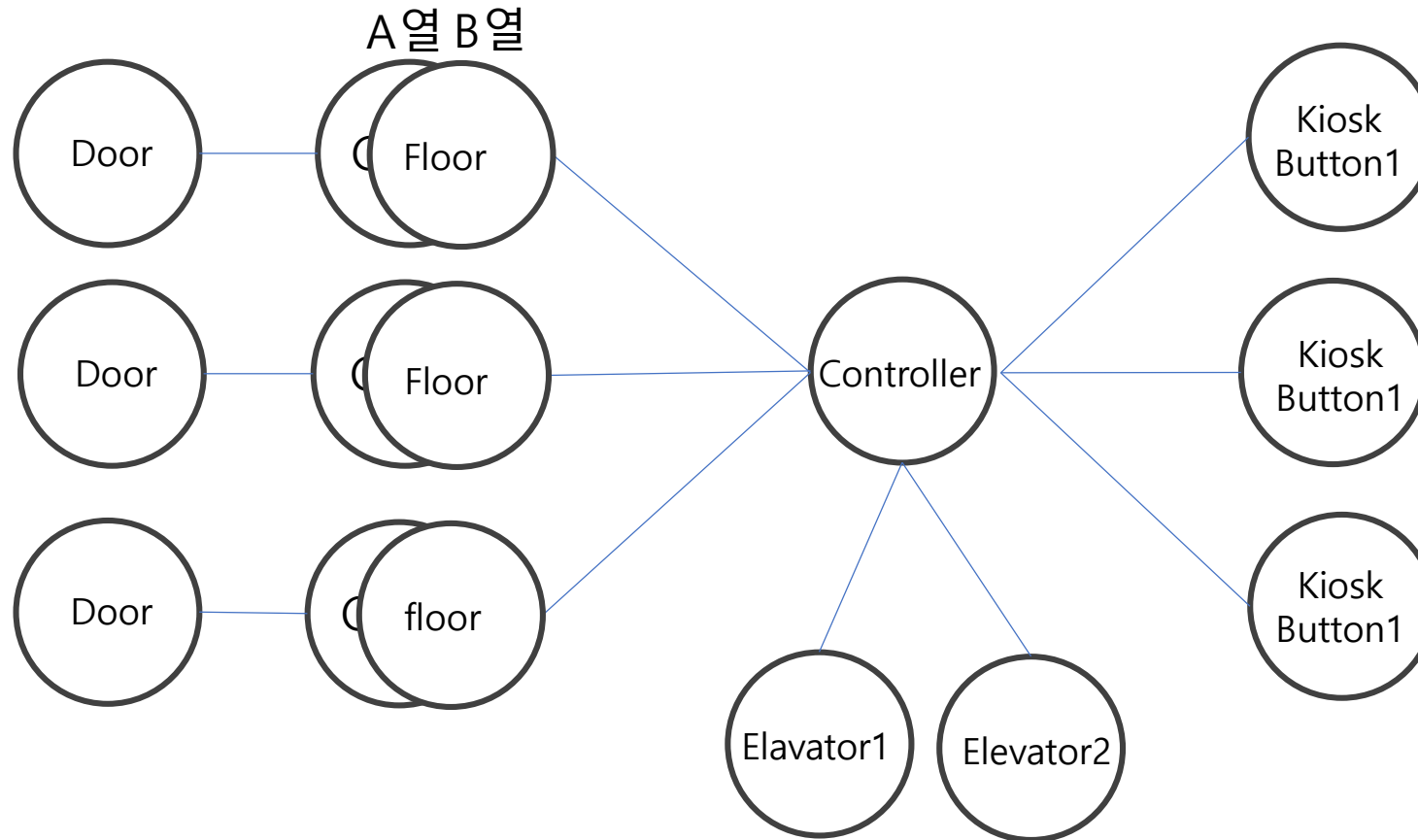
시스템 구성

- Elements
 - KioskButton (1층~3층)
 - Elevator1, Elevator2
 - Controller
 - Floor
 - Door
 - Button
- State Variable
 - State1(idle, down, up), State1
 - Direction1(up, down), Direction1
 - TargetFloor(1~3)
 - Kiosk Button(1~3)



시스템 구성

- 스마트 엘리베이터

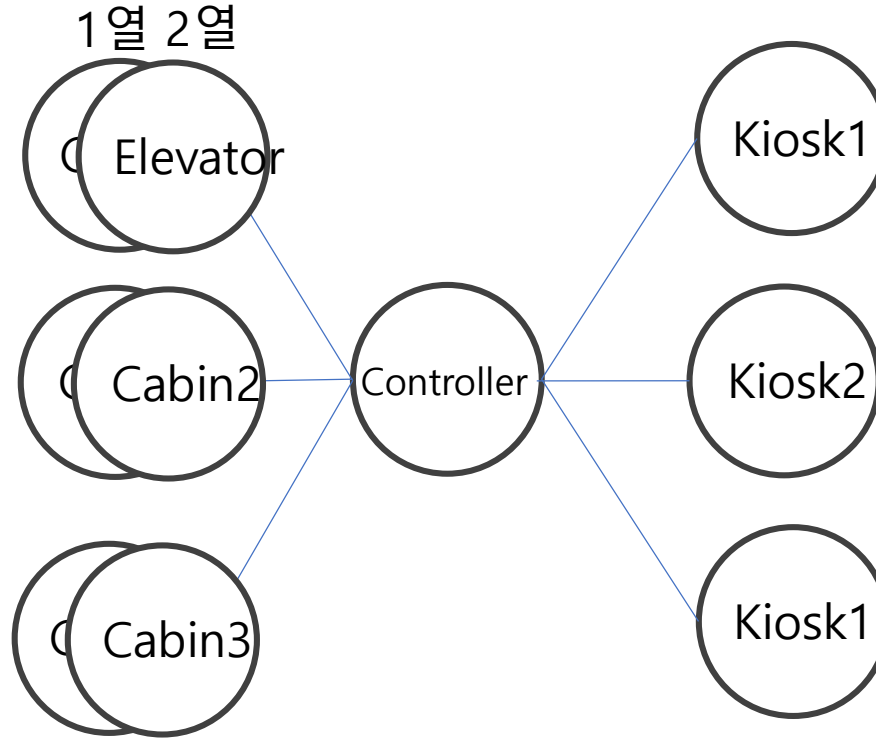


요구사항

- 사용자는 키오스크를 이용해 원하는 목표 층 정보를 입력한다.
- 엘리베이터 내부에서는 열기/닫기 버튼을 통해 문을 조작 가능하다.
- 각 층의 키오스크에서 엘리베이터를 호출할 경우 언젠간 서비스 이용 가능해야 한다.
- 엘리베이터는 탑승 후 목표 층까지 도착 가능해야 한다.
- 사용자의 요청이 없을 경우 엘리베이터는 움직이지 않는다.
- 엘리베이터가 운행하는 동안 문이 열리면 안된다.
- 엘리베이터가 가장 아래층이나 꼭대기 층에 있을때, idle 상태이거나 가능한 방향(up/down) 상태이어야만 한다.

시스템 구성

- 스마트 엘리베이터



Automata

- Elevator Module

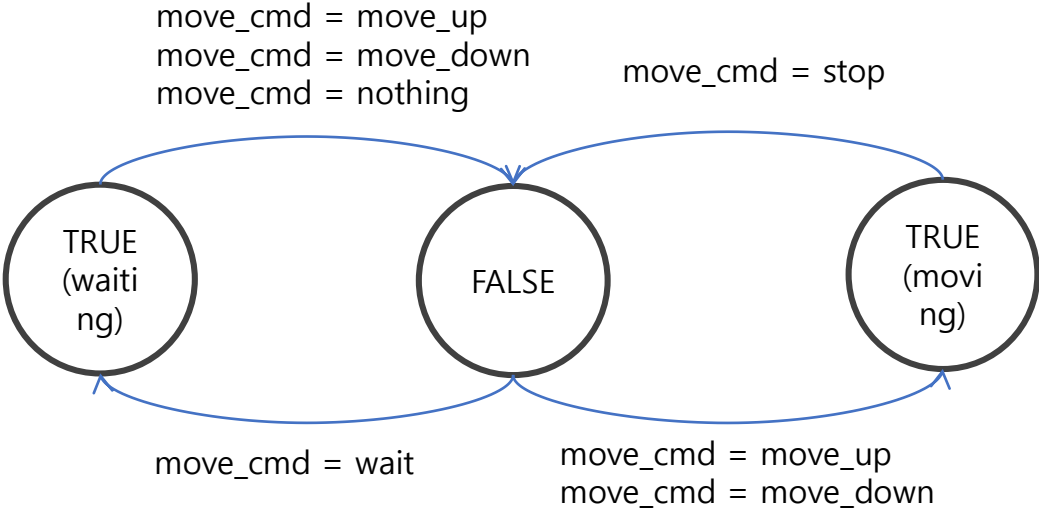


Fig. waiting & moving

Automata

- Elevator Module

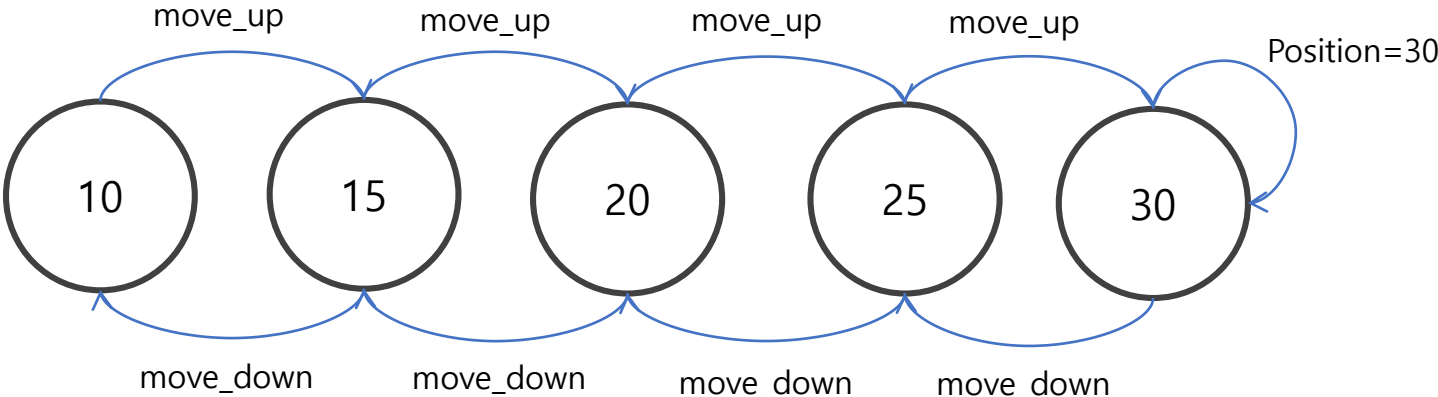


Fig. position

Automata

- Elevator Module

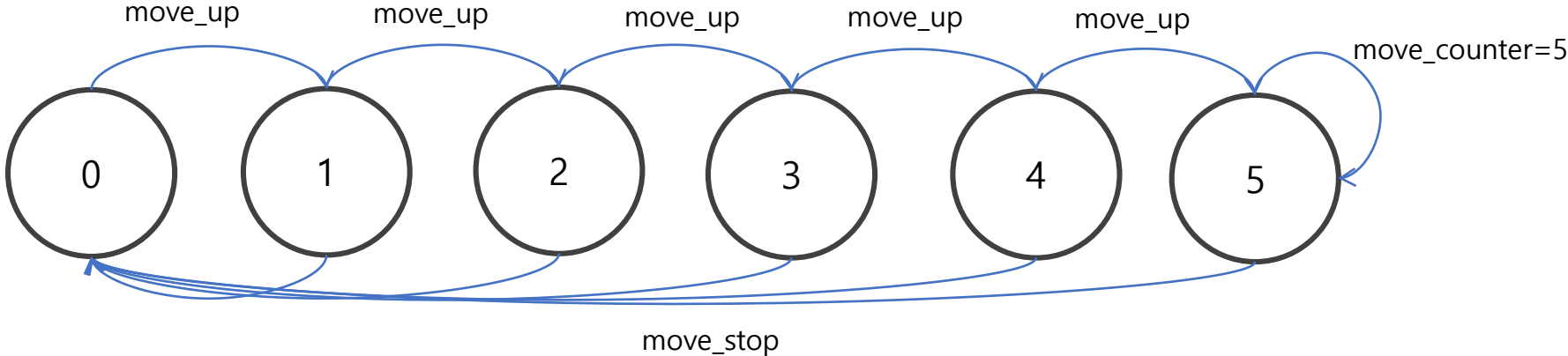


Fig. move_counter

Automata

- Button Module

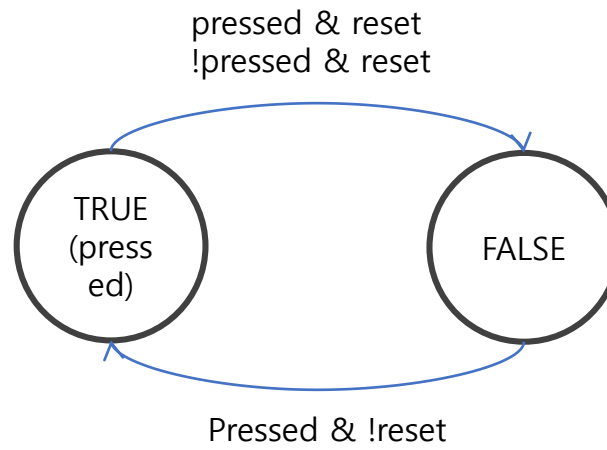


Fig. Button

Automata

- Door Module

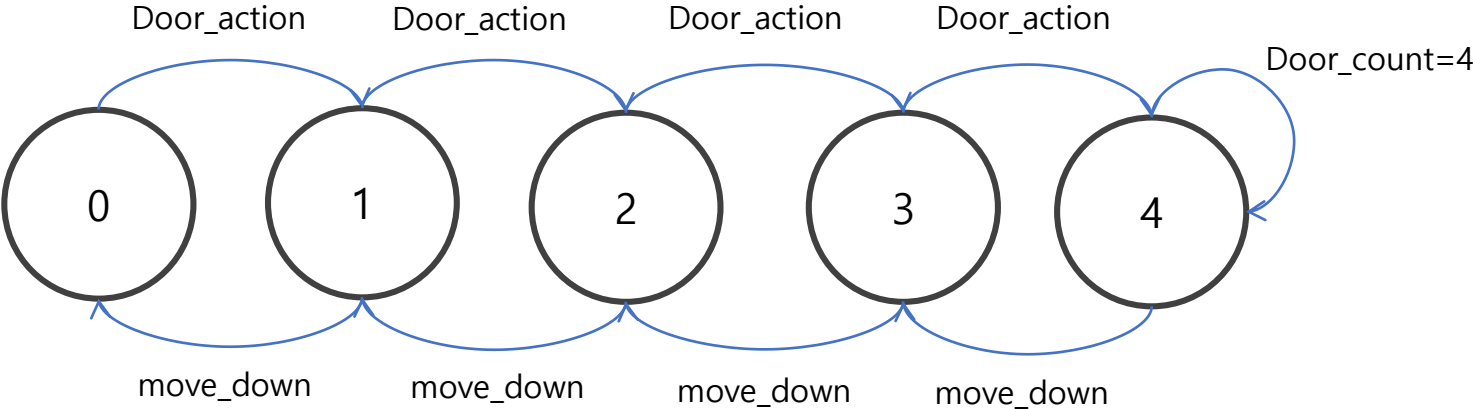


Fig. door_counter

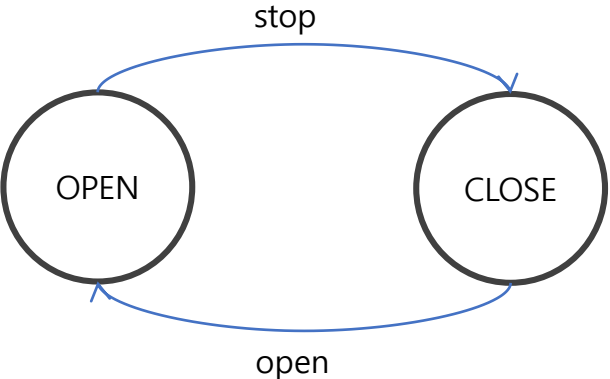
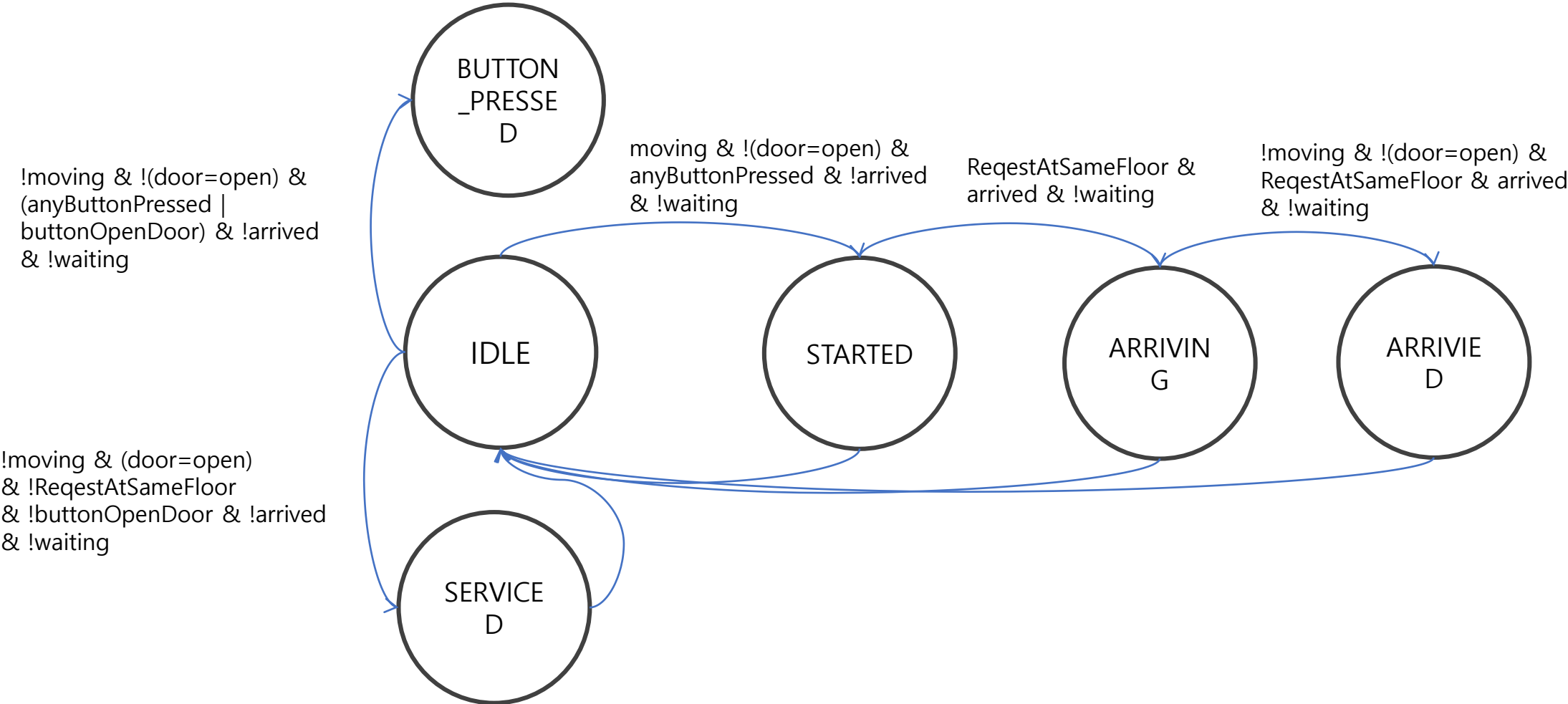


Fig. door_action

Automata

- Controller Module



Property

-- 1. 각 층의 키오스크에서 엘리베이터 호출시 최종적으로 서비스 이용 가능해야 한다.

SPEC AG (Kiosk1btn.pressed -> AF(ctrl.Kiosk1btnServiced))

SPEC AG (Kiosk2btn.pressed -> AF(ctrl.Kiosk2btnServiced))

SPEC AG (Kiosk3btn.pressed -> AF(ctrl.Kiosk3btnServiced))

-- 2. 키오스크에서 입력한 목표층에 대해 최종적으로 서비스 가능해야 한다.

SPEC AG (elevator1.target1floor -> AF(ctrl.kioskButton1Serviced))

SPEC AG (elevator1.target2floor -> AF(ctrl.kioskButton2Serviced))

SPEC AG (elevator1.target3floor -> AF(ctrl.kioskButton3Serviced))

SPEC AG (elevator1.buttonOpenDoorPressed -> AF(door1.status=open))

SPEC AG (elevator2.target1floor -> AF(ctrl.kioskButton1Serviced))

SPEC AG (elevator2.target2floor -> AF(ctrl.kioskButton2Serviced))

SPEC AG (elevator2.target3floor -> AF(ctrl.kioskButton3Serviced))

SPEC AG (elevator2.buttonOpenDoorPressed -> AF(door1.status=open))

-- 3. 엘리베이터가 동작하는 동안 문은 열리면 안된다.

SPEC AG !(elevator1.moving & (door1.status=open))

SPEC AG !(elevator2.moving & (door2.status=open))

-- 4. 엘리베이터 문은 사용이 사용 요청이 있기전까지 열려있어야 한다.

SPEC AG((door1.status=open) -> A[(door1.status=open) U ctrl.anyButtonPressed])(door1.status=open))

SPEC AG((door2.status=open) -> A[(door2.status=open) U ctrl.anyButtonPressed])(door2.status=open))

-- 5. 엘리베이터 요청이 없을 경우 엘리베이터는 동작하지 않는다.

SPEC AG (!(ctrl.anyButtonPressed) -> !(elevator1.moving))

SPEC AG (!(ctrl.anyButtonPressed) -> !(elevator2.moving))

-- 6. Open 버튼을 눌렀을때, 잔여 시간동안 열려있어야 한다.

SPEC (elevator1.buttonOpenDoorPressed & door1.door_counter = 0 -> AF((door1.status=open) & door1.door_counter = 2))

SPEC (elevator2.buttonOpenDoorPressed & door2.door_counter = 0 -> AF((door2.status=open) & door2.door_counter = 2))

검증결과

Model checking results

=====

```
door1.....true
door2.....true
(AG (Kiosk1btn.pressed ->(AF ctrl.Kiosk1btnServiced))).....true
(AG (Kiosk2btn.pressed ->(AF ctrl.Kiosk2btnServiced))).....true
(AG (Kiosk3btn.pressed ->(AF ctrl.Kiosk3btnServiced))).....true
(AG (elevator1.target1floor ->(AF ctrl.kioskButton1Serviced))).....true
(AG (elevator1.target2floor ->(AF ctrl.kioskButton2Serviced))).....true
(AG (elevator1.target3floor ->(AF ctrl.kioskButton3Serviced))).....true
(AG (elevator1.buttonOpenDoorPressed ->(AF (door1.status=open)))).....true
(AG (elevator2.target1floor ->(AF ctrl.kioskButton1Serviced))).....true
(AG (elevator2.target2floor ->(AF ctrl.kioskButton2Serviced))).....true
(AG (elevator2.target3floor ->(AF ctrl.kioskButton3Serviced))).....true
(AG (elevator2.buttonOpenDoorPressed ->(AF (door1.status=open)))).....true
(AG (~(elevator1.moving&(door1.status=open)))).....true
(AG (~(elevator2.moving&(door2.status=open)))).....true
(AG ((door1.status=open) ->((A ((door1.status=open) U ctrl.anyButtonPr.....true
(AG ((door2.status=open) ->((A ((door2.status=open) U ctrl.anyButtonPr.....true
(AG ((~ctrl.anyButtonPressed) ->(~elevator1.moving))).....true
(AG ((~ctrl.anyButtonPressed) ->(~elevator2.moving))).....true
((elevator1.buttonOpenDoorPressed&(door1.door_counter=0)) ->(AF ((door.....true
((elevator2.buttonOpenDoorPressed&(door2.door_counter=0)) ->(AF ((door.....true
```

향후 과제

- 고층 건물일 경우에 대한 설계 적용
- 컨트롤러 모듈의 엘리베이터 스케줄링 적용
- 추가 Property를 도출하여 검증

감사합니다.