

OSP Stage 2030 for Fluxvator: an Elevator Control Simulator

Apr. 8th 2014

Team 3

Team Organization

200913215 기세파

201013275 강태호

201013760 이인구

Context

2031. ESSENTIAL USE CASE DEFINITION	3
2032. REFINE USE CASE DIAGRAMS	13
2033. DOMAIN MODEL DEFINITION	13
2034. GLOSSARY REVISION	14
2035. SYSTEM SEQUENCE DIAGRAMS DEFINITION	15
2036. OPERATION CONTRACTS DEFINITION	18
2037. STATE DIAGRAMS DEFINITION	28

2031. Essential Use Case Definition

Use case	1.1.1 탑승 요청
Actor	User
Purpose	층을 선택하여 엘리베이터를 상승 또는 하강 큐를 할당하는 것을 요청하고, 정해진 하중을 승차시킨다.
Overview	층을 선택하여 상승 또는 하강 요청
Type	Primary and essential
Cross Reference	3.2.1, 3.1.1
Pre-Requisite	엘리베이터가 정상상태이다.
Normal Flow	(A) : Actor, (S) : System 1. (A) : 층을 선택하여 큐 할당을 요청 2. (S) : 존재하는 큐를 확인하여 같은 큐가 있는지 확인하고 있다면 요청취소 기능 수행 3. (S) : Flux calculator 기능으로 큐를 계산 후에 큐를 할당.
Alternative Flow	N/A
Exceptional Flow	N/A

Use case	1.1.2 층 선택
Actor	User
Purpose	사용자가 층을 선택하여 그 층에 엘리베이터가 도착하고 하중을 정하여 하차시킴.
Overview	사용자가 층을 선택하고 큐를 Flux calculator 를 이용하여 할당시킨다.
Type	Primary and essential.
Cross Reference	3.1.1, 3.2.2
Pre-Requisite	엘리베이터가 정상 상태임
Normal Flow	(A) : Actor, (S) : System 1. (A) : 층을 선택하여 큐 할당을 요청 2. (S) : 이미 큐가 있는지 확인한다. 있다면 요청 취소 실행 3. (S) : 없다면 큐를 할당.
Alternative Flow	N/A
Exceptional Flow	N/A

Use case	1.1.3 문 제어요청
Actor	User
Purpose	문을 유저가 직접 제어할 수있게 해준다.
Overview	문의 열림과 닫힘 상태를 유저가 직접 조작할 수 있도록 문 열림 및 문 닫힘 기능에 요청을 보내는 기능이다.
Type	Primary and essential
Cross Reference	3.1.1, 2.1.3, 2.1.1, 2.1.2
Pre-Requisite	엘리베이터가 층에 정지해 있고 정상상태임.
Normal Flow	(A) : Actor, (S) : System 1. (S) : 엘리베이터가 운행에 따라 층에 정지 하고 문을 연다. 2. (A) : 사용자는 문이 더 열려있게 조작하거나, 문을 바로 닫을수 있도록 조작할 수 있다.
Alternative Flow	엘리베이터가 운행도 없고 문이 닫혀있을 경우 1. (A) : 사용자가 열기버튼을 열어 문을 열수 있다. 닫힘 버튼을 누르면 문이 이미 닫혀있기 때문에 아무 일도 일어나지 않는다.
Exceptional Flow	N/A

Use Case	1.1.4 요청취소 (Cancel Request)
Actor	System
Purpose	엘리베이터의 문 상태를 파악한다.
Overview	문열기와 닫기 요청을 받다 요청을 다시 받으면 3.4에 취소 요청을 보낸다.
Type	Primary and Essential
Prerequisites	시뮬레이터가 작동 중
Reference	2.1.1, 2.1.2, 3.4
Normal Flow	A: Actor S: System <ul style="list-style-type: none"> • A: 사용자는 문 닫기 열기 취소요청을 받고 3.4로 보낸다. • S: 문에 대해서 내외부요청취소를 받아 전달한 매체다.
Alternative Flow	N/A
Exception Flow	

Use Case	1.2.1 상태시뮬레이트(Config State)
Actor	System
Purpose	엘리베이터의 현재 상태(비상/정상/수리)를 파악한다.
Overview	엘리베이터 운행한 동안 운행상태를 파악하고 이루어진 문제를 생기면 실시간 3.1.1로 보낸다.
Type	Primary and Essential
Prerequisites	시뮬레이터가 작동 중
Reference	3.1.1, 3.1.2, 3.3.1, 3.3.2, 3.4
Normal Flow	A: Actor S: System <ul style="list-style-type: none"> A: 사용자는 화재 또는 정전 등 상황을 시뮬레이트 하고자 하여 기능 3.1.1을 이용해 희망하는 상황을 전달한다. S: 엘리베이터의 현재 상황을 파악하고 빠르게 전달한다.
Alternative Flow	N/A
Exception Flow	만일 유저가 여러 개의 상황을 동시에 했을 경우: 동시에 설정한 상황이 모두 비상 상황의 일종일 경우는 하나씩 3.1.1로 보낸다.

Use Case	1.2.2. 최대하중 조정 (Config MaxLoad)
Actor	System
Purpose	시뮬레이션에 사용될 엘리베이터의 최대 하중 정의
Overview	시뮬레이션 내 엘리베이터의 최대 하중을 정의하여 하중 적재시 비교에 사용하도록 한다.
Type	Primary and Essential
Prerequisites	시뮬레이션이 가동 중
Reference	1.2.2
Normal Flow	A: Actor S: System <ul style="list-style-type: none"> A: 시뮬레이션 사용자가 최대하중을 정의한다. S: 사용자의 정의에 따라 최대하중이 변경된다. S: 큐 할당을 호출하여 새로운 탑승요청을 할당 시킨다.
Alternative Flow	엘리베이터가 꽉 찬 상태에서 최대하중을 변경할 경우 <ul style="list-style-type: none"> 최대하중의 변경을 엘리베이터가 층에 정차한 뒤에 반영한다.
Exception Flow	

Use Case	2.1.1 문 열기 (Open Door)
Actor	System
Purpose	엘리베이터의 문 열기 동작
Overview	명령이 들어올 경우 엘리베이터 문을 열도록 한다.
Type	Primary and Essential
Prerequisites	엘리베이터가 층에 정차상태
Reference	3.4, 2.1.1
Normal Flow	<p>A: Actor S: System</p> <ul style="list-style-type: none"> • A: 사용자가 문 제어 요청을 한다. • S: 3.4의 문 제어 계산기가 사용자의 문 제어 요청을 반영하기 적합한 상황인지(e.g. 엘리베이터가 층에 멈춰 있는지) 확인하고 적합할 경우 문 열기 기능을 호출한다. • S: 문 열기 기능이 호출되어 문이 열린다.
Alternative Flow	
Exception Flow	

Use Case	2.1.2. 문 닫기 (Close Door)
Actor	System
Purpose	엘리베이터의 문 닫기 동작
Overview	명령이 들어올 경우 문 닫기를 수행한다.
Type	Primary and Essential
Prerequisites	엘리베이터가 층에 정차상태
Reference	3.4, 2.1.1
Normal Flow	<p>A: Actor S: System</p> <ul style="list-style-type: none"> • A: 사용자들이 문 제어 요청을 한다. • S: 3.4의 계산기가 문 제어 요청이 반영될 수 있는 상황인지(e.g. 엘리베이터가 정차 상태인지, 문이 열려 있는지) 확인하고 적합할 경우 문 열기 기능을 호출한다. • S: 기능이 호출되어 문이 닫힌다.
Alternative Flow	•
Exception Flow	

Use Case	2.1.3. 움직임 제어 (Movement Control)
Actor	System
Purpose	엘리베이터 캐빈을 움직인다.
Overview	엘리베이터 캐빈의 실제 움직임을 수행시킨다.
Type	Primary and Essential
Prerequisites	시뮬레이션이 작동 중
Reference	2.1.3, 3.2.2, 3.2.3, 3.4, 1.1.2
Normal Flow	A: Actor S: System <ul style="list-style-type: none"> S: 큐 할당이나 삭제(3.2.2, 3.2.3)을 통하여 새로운 탑승요청에 대한 운행 명령을 받는다. S: 받은 명령에 따라 엘리베이터를 이동시킨다.
Alternative Flow	
Exception Flow	

Use Case	3.1.1 상황판별 (Determine State)
Actor	System
Purpose	엘리베이터의 현재 상태(비상/정상/수리)를 파악한다.
Overview	화재감지기 등 상태를 알려줄 수 있는 시스템으로부터 정보를 받아 엘리베이터의 현재 가동 상태를 지정한다. 이 시뮬레이터 에서는 상태 시뮬레이트 기능(1.2.1)으로부터 상태를 받아 결정하도록 한다.
Type	Primary and Essential
Prerequisites	시뮬레이터가 작동 중
Reference	3.1.1, 1.2.1
Normal Flow	A: Actor S: System <ol style="list-style-type: none"> A: 사용자는 화재 또는 정전 등 상황을 시뮬레이트 하고자 하여 기능 1.2.1을 이용해 희망하는 상황을 상황 판별로 전달한다. 사용자가 정의할 수 있는 상황은 총 4가지 이다 : 정상, 비상(화재), 비상(정전), 수리 S: 엘리베이터의 현재 상황을 위에서 넘겨받은 상황으로 전환한다.
Alternative Flow	N/A
Exception Flow	만일 유저가 여러 개의 상황을 동시에 했을 경우: <ol style="list-style-type: none"> 동시에 설정한 상황이 모두 비상 상황의 일종일 경우는 더 중요한 상황을 알리도록 한다. (이 시스템에는 화재와 정전 두가지만 존재하므로 무조건적으로 화재로 적용하도록 한다.)

Use Case	3.1.2 하중판별 (Determine Load)
Actor	System
Purpose	현재 하중과 엘리베이터 허용 하중을 비교하여 하중 초과 여부를 계산한다.
Overview	현재 하중과 엘리베이터 허용 하중을 비교하여 하중여유/꽉참 두가지중 한 가지 상태를 알린다.
Type	Primary and Essential
Prerequisites	문이 열려 있음
Reference	3.1.2, 1.2.1
Normal Flow	<p>A: Actor S: System</p> <ol style="list-style-type: none"> 1. A: 사용자가 탑승 요청을 하면서 탑승시킬 하중을 정한다. 2. S: 탑승요청이 된 층에 엘리베이터가 도착하면서 탑승요청에 정한 하중과 엘리베이터 현재 하중을 더한다. 3. S: 설정된 허용 하중과 현재 하중을 비교하여 여유가 있을시 하중을 모두 추가하게 되고, 초과 시 넣을 수 있는 하중만큼의 인원만 넣어 지며 하중 상태를 꽉참으로 정의한다.
Alternative Flow	<ol style="list-style-type: none"> 1. A:사용자가 층을 선택하면서 떨어낼 하중을 정의한다. 2. S:선택한 층에 멈출 시 층선택 에서 정의있던 하중 만큼을 떨어내며, 이전에 엘리베이터 상태가 꽉참 상태였을 시 상태를 여유로 전환한다.
Exception Flow	

Use Case	3.2.1. Flux 계산 (Flux Calculation)
Actor	System
Purpose	엘리베이터 운영을 효율적으로 할 수 있도록 보장
Overview	엘리베이터 운영을 효율적으로 할 수 있도록 운영 시나리오를 계산하여 엘리베이터 큐에 할당 및 삭제를 하도록 한다.
Type	Primary and Essential
Prerequisites	엘리베이터 현재상태가 정상
Reference	3.2.1, 1.1.1, 1.1.2, 1.1.4, 2.1.3, 3.1.1, 3.1.2, 3.2.2, 3.2.3
Normal Flow	A: Actor S: System <ol style="list-style-type: none"> 1. A: 사용자들이 탑승요청을 한다. 2. S: Flux 계산이 엘리베이터의 운영 큐를 읽어들이어 운영 상황을 파악한다. 3. S: 각 엘리베이터의 큐에 새로 들어온 탑승요청을 넣는 시뮬레이팅을 하여 어느 엘리베이터에 요청을 할당하는 것이 더 효율적인지를 계산한다. 4. S: 큐 할당을 호출하여 새로운 탑승요청을 할당 시킨다.
Alternative Flow	탑승요청 계산 시 한대 이상의 엘리베이터가 하중 한계 도달 상태일 경우 <ol style="list-style-type: none"> 1. Normal Flow의 Line 2까지는 정상적으로 진행된다. 2. Normal Flow의 Line 3에서 계산을 수행 할 때, 하중 여유가 있을 때의 계산 방식과는 다르게 계산을 수행하도록 한다. 3. 계산 결과에 따라 큐 할당을 호출하여 탑승요청을 할당 시킨다.
Exception Flow	

Use Case	3.2.2. 큐 할당 (Enqueue Order)
Actor	System
Purpose	엘리베이터 운영 큐에 새로운 명령을 할당 한다.
Overview	엘리베이터 운영 큐에 새로운 명령을 할당 한다.
Type	Primary and Essential
Prerequisites	시뮬레이션 작동 중
Reference	3.2.2, 1.1.2, 1.1.4, 2.1.3, 3.2.1, 3.3.1, 3.3.2
Normal Flow	A: Actor S: System <ol style="list-style-type: none"> 1. A&S: 층 선택, Flux 계산, 화재 컨트롤, 정전 컨트롤 등 엘리베이터 운행에서 새로운 목적지 설정 명령을 입력 받는다. 2. S: 해당 명령을 큐에 할당 시킨다.
Alternative Flow	
Exception Flow	

Use Case	3.2.3. 큐 삭제 (Dequeue Order)
Actor	System
Purpose	엘리베이터 운행 큐에 새로운 명령을 할당 한다.
Overview	엘리베이터 운행 큐에 새로운 명령을 할당 한다.
Type	Primary and Essential
Prerequisites	시뮬레이션 작동 중
Reference	3.2.3, 1.1.1, 1.1.4, 2.1.3, 3.2.1, 3.3.1, 3.3.2
Normal Flow	A: Actor S: System <ol style="list-style-type: none"> 1. A&S : 요청 취소, Flux 계산, 화재 컨트롤, 정전 컨트롤 등 새로운 운행 명령이나 움직임 제어로부터 목적지 도달 알림 등을 받는다. 2. S: 해당 명령을 큐에서 제거하도록 한다.
Alternative Flow	
Exception Flow	

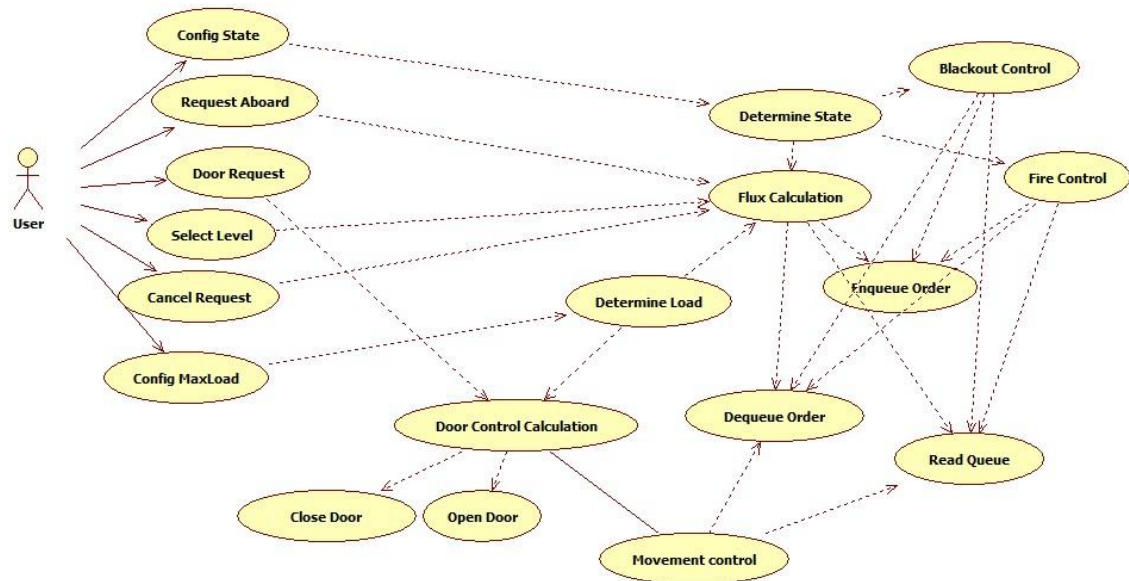
Use Case	3.2.4 큐 읽기(Read Queue)
Actor	System
Purpose	엘리베이터 큐의 내용을 읽어들인다.
Overview	운행 제어나 Flux Calculation 등에서 현재 큐의 내용을 확인할 때 이용한다.
Type	Primary and Essential
Prerequisites	시뮬레이터가 작동 중
Reference	3.2.4, 3.2.1, 2.1.3
Normal Flow	A: Actor S: System <p>S: 새로운 탑승 요청이나 층 선택 등으로 인한 Flux 계산, 혹은 단순히 엘리베이터 운행에서 다음 도착지 설정 등 큐의 내용을 확인해야 할 기능이 앞서 수행된다.</p> <p>S: 큐 읽기를 통해 큐의 내용(노드)들을 확인한다.</p> <p>S: 큐의 내용(노드)들을 요청한 기능에 돌려준다.</p>
Alternative Flow	N/A
Exception Flow	

Use case	3.3.1 화재컨트롤
Actor	System
Purpose	화재상황 발생 시에 주어진 행동을 실행한다.
Overview	화재상황 발생 시에 이미 존재하는 모든 큐를 삭제하고 상황에 따라 그에 대응하는 큐를 넣어준다. 그리고 정상상태가 될 때까지 요청들을 무시한다.
Type	Secondary
Cross Reference	1.2.1, 3.2.2, 3.2.3 1.1.1, 1.1.2, 1.1.3, 1.1.4
Pre-Requisite	상태시뮬레이트로 화재를 발생시킴
Normal Flow	(A) : Actor, (S) : System 1. (A) : 유저가 화재를 일으킨다. 2. (S) : 현재 존재하는 모든 큐를 삭제, 화재가 일어난 층을 제외한 가장 가까운 층에 모든 하중을 하차시킨다. 3. (S) : 추락을 방지하기 위해 1 층으로 엘리베이터를 이동시킨다. 4. (S) : 시스템이 정상으로 돌아가기 전까지 모든 요청을 무시한다.
Alternative Flow	화재가 1 층에서 났을 경우 1. (S) : Line 2 와 동일 2. (S) : 하차시킨 층에 대기한다. 3. (S) : Line 4 와 동일
Exceptional Flow	N/A

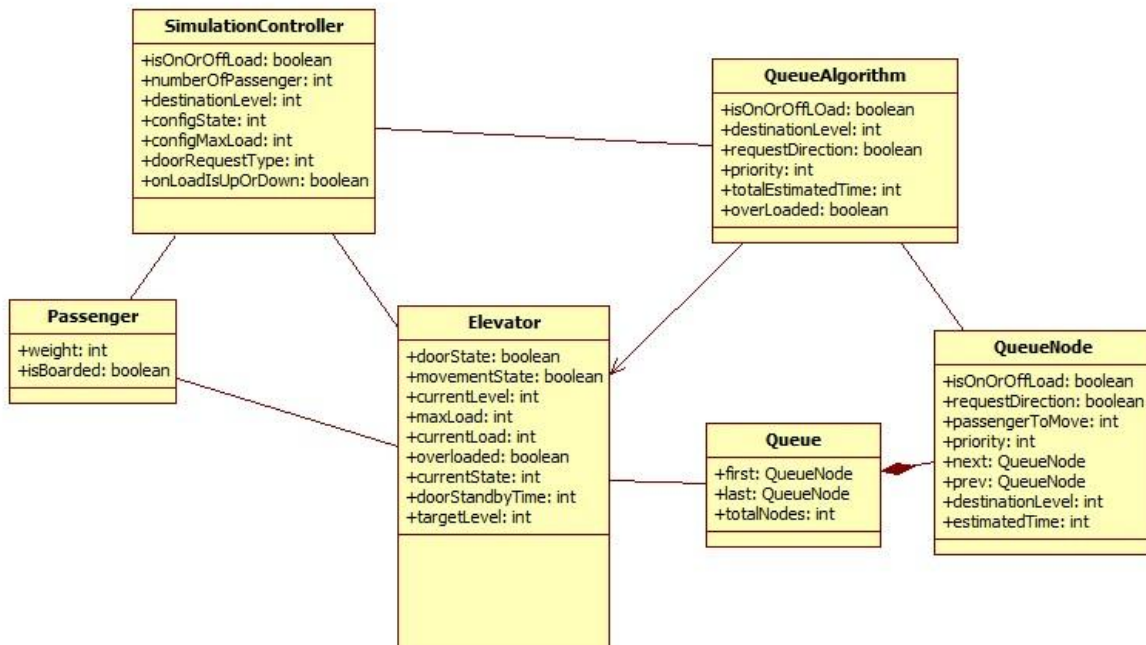
Use Case	3.3.2 정전컨트롤(Blackout Control)
Actor	System
Purpose	엘리베이터의 정전 상황을 발생할 때 실시간 처리할 수 있음
Overview	엘리베이터 운행한 동안 1.2.1로 받는 정전상황을 발생한 경우 처리한다.
Type	Primary and Essential
Prerequisites	시뮬레이터가 작동 중
Reference	3.1.1, 3.1.2, 3.3.2, 3.4
Normal Flow	<p>A: Actor S: System</p> <ul style="list-style-type: none"> A: 사용자는 정전 상황을 시뮬레이트 하고자 하여 기능 1.2.1을 이용해 희망하는 상황을 상황과 처리방식을 운영규에 전달한다. 사용자가 정의할 수 있는 상황은 정전 말고 3가지 이다 : 정상, 비상(화재), 수리 <p>S: 엘리베이터의 정전상황을 파악하고 빠르게 처리한다.</p>
Alternative Flow	N/A
Exception Flow	

Use Case	3.4. 문 제어 계산 (Door Control Calculation)
Actor	System
Purpose	문 제어 요청을 받았을 때 문을 제어할 것인지 말지를 결정
Overview	문 제어 요청과 상황을 반영하여 문을 제어할 것인지 말지를 결정하게 된다.
Type	Primary and Essential
Prerequisites	시뮬레이션이 작동 중
Reference	2.1.2, 2.1.1, 2.1.3, 3.1.2, 3.1.1
Normal Flow	<p>A: Actor S: System</p> <ul style="list-style-type: none"> A: 사용자는 문제어 요청 기능을 이용해 문을 열거나 닫고자 요청을 한다. S: 문 제어 계산 기능은 사용자의 요청이 적합한지에 대해 현재 상황을 다음을 참조하여 파악한다: 1)엘리베이터가 층에 정차 중인지, 2) 엘리베이터 운행 상태가 정상인지, 3) 하중 초과여부 현재 상황의 조건이 문 제어에 적합하다고 판단되면, 문 열기/닫기 기능을 호출하여 문을 제어하도록 한다.
Alternative Flow	<ul style="list-style-type: none"> 사용자의 제어 요청이 없으나 엘리베이터가 이동 후 층에 도달하였을 경우 S: 문 열기 기능을 호출한다. S: 문이 열려 있는데 사용자의 문 열기/ 닫기 요청이 없는 경우, 정해진 대기시간 동안 기다린 후 문 닫기 기능을 호출하도록 한다.
Exception Flow	

2032. Refine Use Case Diagrams



2033. Domain Model Definition



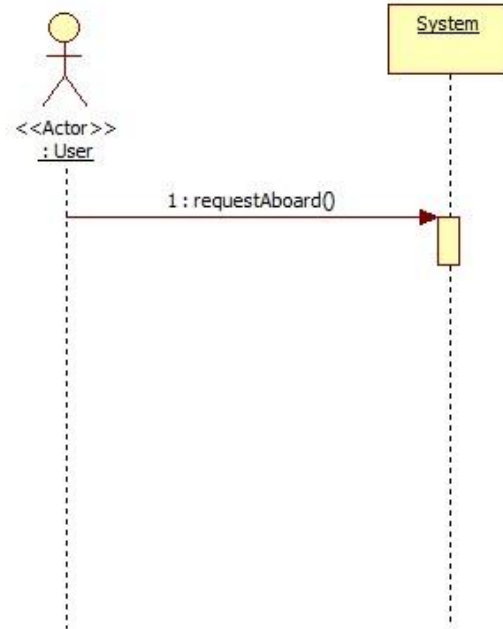
2034. Glossary Revision

Term	Category	Explanation
SimulationController	Class	사용자의 조작기능이 주로써 대부분의 요청들 (Use case 1번대)
QueueAlgorithm	Class	Flux Calc의 계산 알고리즘
Passenger	Class	시뮬레이션에서의 엘리베이터 하중계산을 무게로 하나 탑승자는 무게를 가진 사람.
Elevator	Class	엘리베이터 캐빈으로 기본적인엘레베이터의 속성 외 직접 제어를 하는 Use case 2번대의 기능들이 포함됨
Queue	Class	엘리베이터의 요청들과 명령들이 우선순위에 따라 수행되도록 이뤄지는 큐
QueueNode	Class	큐의 원소가 되는 노드
isOnOrOffLoad	Attribute	요청이 탑승 요청인지 층선택(하차) 요청인지의 여부
destinationLevel	Attribute	탑승시 탑승하는 층, 층 선택시 하차하는 층
doorRequestType	Attribute	문 열기/ 닫기 요청의 구분
onLoadIsUpOrDown	Attribute	탑승 요청시 상승 요청인지 하강 요청인지의 방향
requestDirection	Attribute	해당 요청에 대해 엘리베이터가 가야 할 방향 (상/하) 표현
totalEstimatedTime	Attribute	엘리베이터의 큐에 대해 해당 요청 추가 시 엘리베이터가 큐의 요청들을 수행하는데 걸릴 총 시간
doorState	Attribute	현재 문 닫힘/열림 상태
movementState	Attribute	현재 엘리베이터 이동/정지 상태
doorStandbyTime	Attribute	문 열기/닫기 요청 없을 시 문 대기 시간
targetLevel	Attribute	가고자 하는 목적지 층
passengerToMove	Attribute	큐 노드의 해당 요청에서 태우거나 내려야 할 승객 수
estimatedTime	Attribute	큐 노드의 해당 요청 수행에 걸리게 될 시간

2035. System Sequence Diagrams Definition

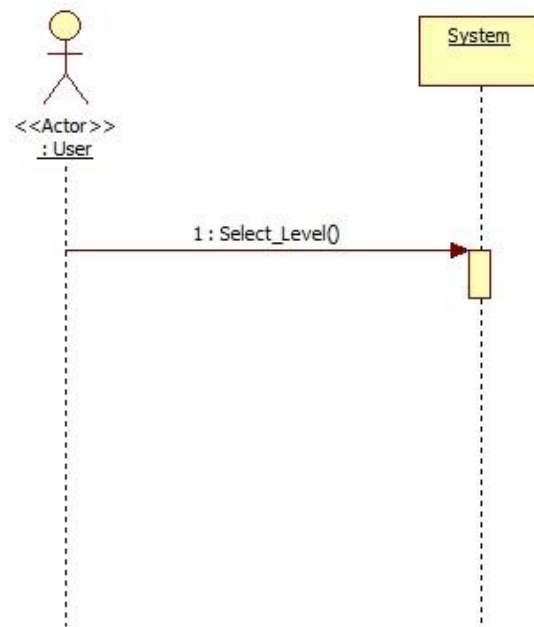
1.1.1 탑승 요청(Request Aboard)

1. User selects level to send aboard request
2. If System find same request in queue, cancel request
3. Else, use Flux Calculation to assign queue of priority and elevator. And enqueue.

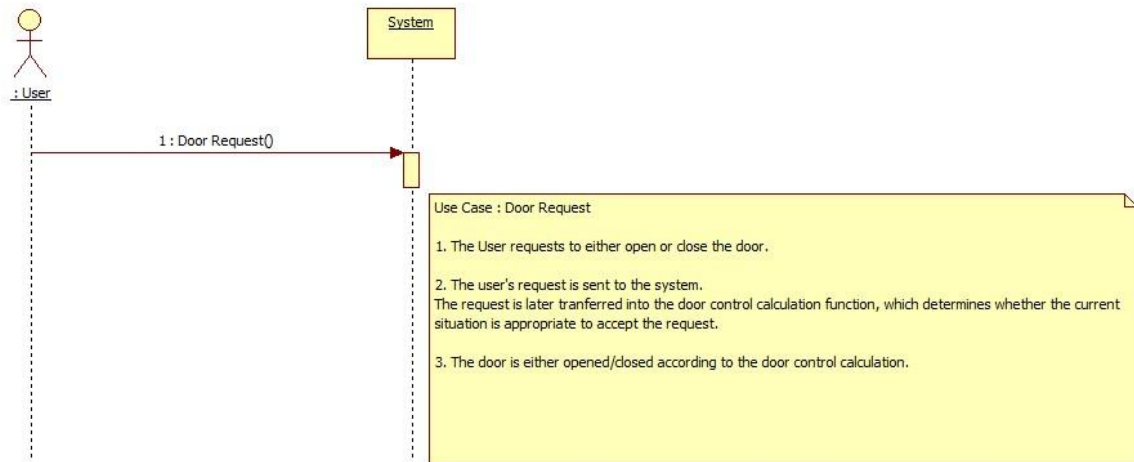


1.1.2 층 선택(Select Level)

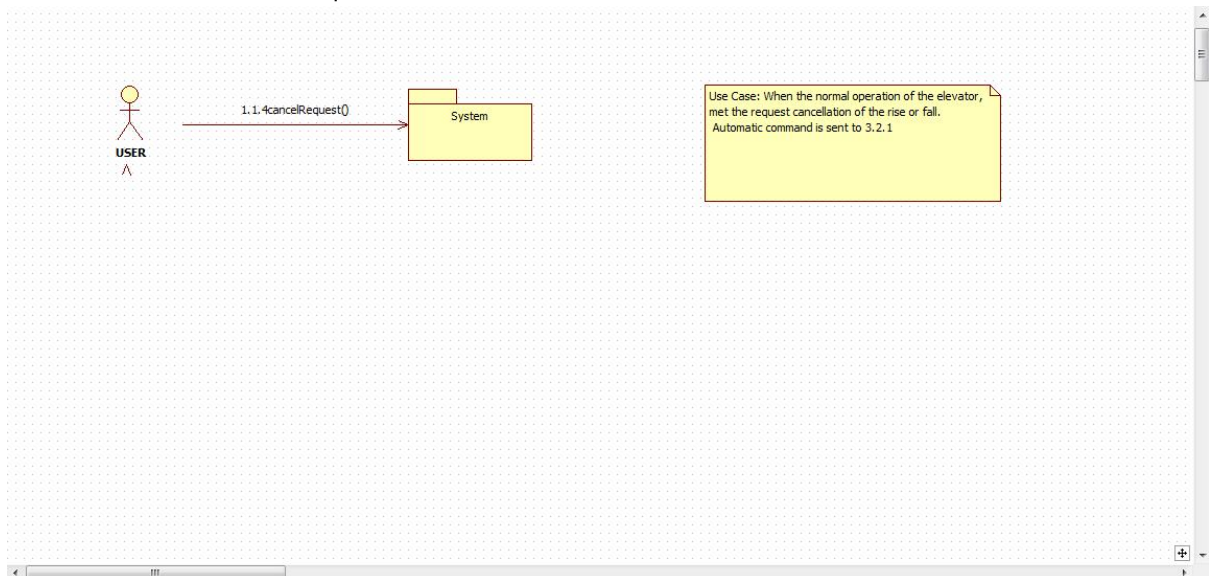
1. User selects level to send level-selection request
2. If system can same request in queue, cancel request.
3. Else, use Flux Calculation to assign queue of priority and elevator. And enqueue



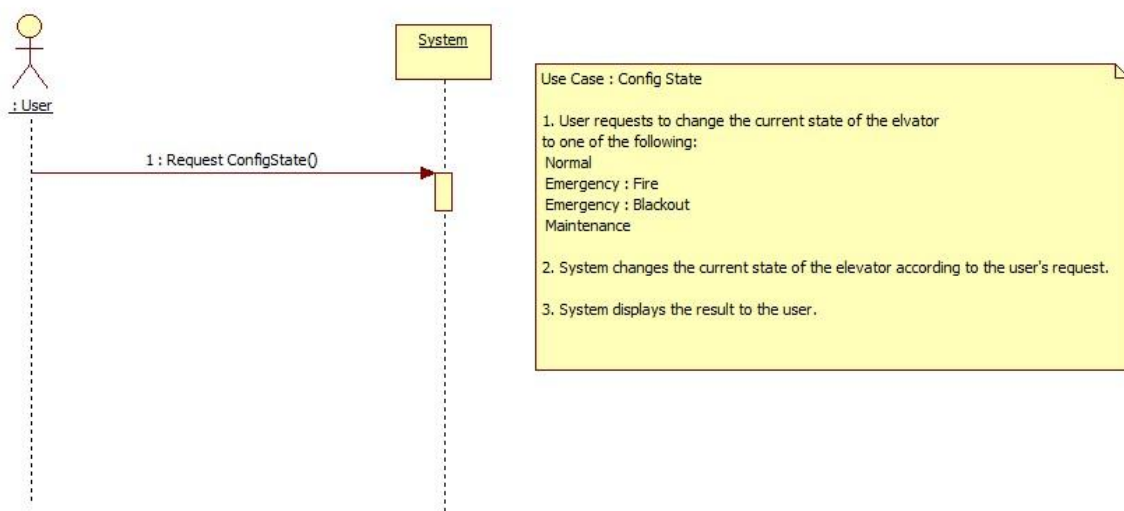
1.1.3 문제어 요청(Door Request)



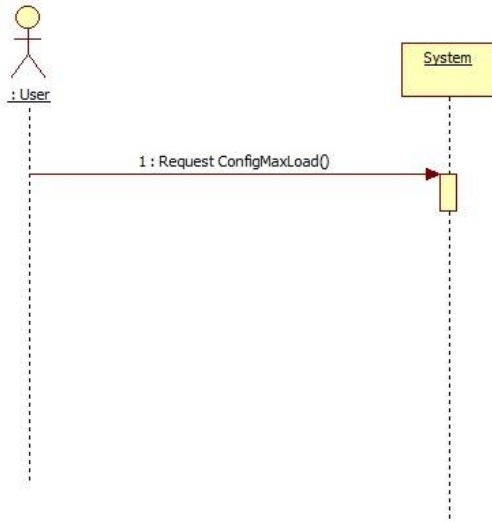
1.1.4 요청 취소(Cancel Request)



1.2.1 상태 시뮬레이트(Config State)



1.2.2 최대하중 조정(Config MaxLoad)



Use Case : Config MaxLoad

1. User requests to change the current specified max load allowance of the elevator.
2. System changes the maximum allowed load of the elevator the the user's input.
3. System gives feedback to the user.

2036. Operation Contracts Definition

Use Case	Actor-Activated Event	System Operations
1. Request Aboard	Request Aboard	requestAboard()
2. Select Level	Select Level	selectLevel()
3. Door Request	Door Request	doorRequest()
4. Cancel Request	Cancel Request	cancelRequest()
5. Config State	Request ConfigState	configState()
6. Config MaxLoad	Request ConfigMaxLoad	configMaxLoad()
7. Open Door	-	openDoor()
8. Close Door	-	closeDoor()
9. Movement Control	-	move()
		stop()
		executeQueue()
10. Determine State	-	determineState()
11. Determine Load	-	determineLoad()
12. Flux Calculation	-	handleReqeustAboard()
		handleFullLoad()
		handleSelectLevel()
		handleCancelRequest()
13. Enqueue Order	-	enqueueOrder()
14. Dequeue Order	-	dequeueOrder()
15. Read Queue	-	readQueue()
16. Fire Control	-	fireControl()
17. Blackout Control	-	balckoutControl()
18. Door Control Calculation	-	wait()
		handleDoorRequest()
		arrivalDoorCalibration()
		departDoorCalibration()

1. Request Aboard

Name	requestAboard()
Responsibilities	특정 층에서의 엘리베이터의 상승 또는 하강요청을 외부요청으로 큐에 추가하도록 한다.
Type	System
Cross Reference	3.1.1, 3.2.2
Exceptions	cancelRequest()(요청 취소)를 호출한다.
Output	상승 혹은 하강의 방향을 가지며 외부요청으로 계산된 우선순위를 가진 QueueNode.
Pre-condition	엘리베이터가 정상상태임.
Post-condition	Queue 에 QueueNode 가 추가된다.

2. Select Level

Name	Select_Level()
Responsibilities	특정 층에서의 엘리베이터의 상승 또는 하강요청을 내부요청으로 큐에 추가하도록 한다.
Type	System
Cross Reference	3.1.1, 3.2.2
Exceptions	cancelRequest()(요청 취소)를 호출한다.
Output	상승 혹은 하강의 방향을 가지며 내부요청으로 계산된 우선순위를 가진 QueueNode.
Pre-condition	엘리베이터가 정상상태임.
Post-condition	Queue 에 QueueNode 가 추가된다.

3. Door Request

Name	doorRequest()
Responsibilities	엘리베이터의 문 상태를 사용자가 Boolean 값을 요청하여 직접 제어할 수 있게 해준다.
Type	System
Cross Reference	3.1.1, 2.1.3, 2.1.1, 2.1.2
Exceptions	N/A
Output	Elevator_Cabin Object 의 Door 의 Boolean 값
Pre-condition	Elevator_Cabin 의 Movement_State 가 멈춤상태임.
Post-condition	Elevator_Cabin 의 Door 의 Boolean 값이 요청에 따라 즉시 바뀐다.

4. Cancel Request

Operation Name	cancelRequest()
Responsibilities	먼저 요청되었으나 아직 엘리베이터가 완전히 수행하지 않은 요청에 대해 취소 요청을 Flux Calculation기능 으로 보낸다.
Type	System
Cross References	1.1.4, 3.2.1, 3.1.1
Notes	
Exception	
Output	시스템 Flux Calculation 기능으로 보내는 요청 취소 메시지
Preconditions	엘리베이터가 정상 상태임
Postconditions	유효한 요청에 대한 요청 취소일 경우 해당 요청이 취소되 큐에서 삭제됨.

5. Config State

Operation Name	configState()
Responsibilities	정하고자 하는 엘리베이터의 상태(정상/화재/정전/수리)를 입력 받아 Determine State 기능으로 넘긴다.
Type	System
Cross References	1.2.1, 3.1.1
Notes	
Exception	다중비상상황 일 경우 : 회의
Output	시스템 Determine State 기능으로 보내는 메시지
Preconditions	시뮬레이션 가동 상태
Postconditions	엘리베이터 가동 상태가 변경된다.

6. Config MaxLoad

Operation Name	configMaxLoad()
Responsibilities	정하고자 하는 엘리베이터의 최대 하중을 입력받아 Determine Load 기능 등에 반영되도록 변경한다.
Type	System
Cross References	1.2.2, 3.1.2, 3.4
Notes	
Exception	
Output	시스템 Determine Weight 기능으로 보내는 메시지
Preconditions	시뮬레이션 가동 상태 & 엘리베이터 정지 상태
Postconditions	엘리베이터 한도 하중이 변경된다.

7. Open Door

Operation Name	openDoor()
Responsibilities	명령이 호출될 경우 문을 연다.
Type	System
Cross References	2.1.1, 3.4
Notes	
Exception	
Output	문 상태의 전환
Preconditions	문제어 계산의 Use Case에서 문열기 호출
Postconditions	문이 열린다.

8. Close Door

Operation Name	closeDoor()
Responsibilities	명령이 호출될 경우 문을 닫는다.
Type	System
Cross References	2.1.2, 3.4
Notes	
Exception	
Output	문 상태의 전환
Preconditions	문제어 계산의 Use Case에서 문닫기 호출
Postconditions	문이 닫힌다.

9. Movement Control

Operation Name	move()
Responsibilities	executeOrder()에 따라 엘리베이터가 움직여야 될 때 호출하여 엘리베이터를 정지 상태에서 움직임 상태로 전환하고 움직인다.
Type	System
Cross References	2.1.3
Notes	
Exception	
Output	엘리베이터가 움직임
Preconditions	executeOrder() 시행, 문 닫힘
Postconditions	엘리베이터의 움직임

Operation Name	stop()
Responsibilities	executeOrder()에 따라 엘리베이터가 정차하여야 할 때 호출하여 엘리베이터를 정지시킨다.
Type	System
Cross References	2.1.3
Notes	
Exception	
Output	엘리베이터 이동 중, executeOrder() 시행
Preconditions	엘리베이터 정지
Postconditions	엘리베이터의 운행과 이로 인한 큐 dequeueOrder()에 따른 큐의 변화

Operation Name	executeQueue()
Responsibilities	받은 큐에서의 큐 노드들 중 가장 먼저 수행해야 할 노드(i.e. 가장 우선순위가 높은 노드)를 명령으로 전환하여 엘리베이터의 직접적인 조종 수행
Type	System
Cross References	2.1.3, 3.2.4
Notes	
Exception	탑승요청에 대한 큐 수행시 탑승요청 하중으로 인해 하중 초과가 일어날 경우 초과한만큼의 하중을 탑승에서 제외하여 새로운 requestAboard()의 값으로 넣어 호출한다.
Output	수행한 명령에 해당되는 큐 노드의 dequeueOrder() 수행
Preconditions	readQueue()의 수행 결과로 큐의 최상위 큐 노드가 반환됨
Postconditions	엘리베이터의 운행과 이로 인한 큐 dequeueOrder()에 따른 큐의 변화

10. Determine State

Operation Name	determineState()
Responsibilities	Config State로부터 넘겨받은 상태 메시지를 반영하여 엘리베이터의 현재 상태를 변경시킨다.
Type	System
Cross References	3.1.1, 1.2.1
Notes	
Exception	두개의 비상 상황 중에서는 화재 상황을 우선시 한다.
Output	configState()로부터 받은 네 가지 상태(정상/화재/정전/수리) 중 하나로 현재 상태 변경
Preconditions	N/A
Postconditions	정상 상태일 경우 Flux Calculation과 Queue를 이용한 운영을 하도록 하며, 비상 상황일 경우 상황에 따라 fireControl()이나 blackoutControl()을 호출하여 Flux Calculation을 이용한 운영을 중단하도록 한다.

11. Determine Load

Operation Name	determineLoad()
Responsibilities	설정된 maxLoad와 엘리베이터의 현재 하중을 비교하여 여유가 있는지, 혹은 꽉 찼는지, 아니면 현재 하중이 초과 상태인지 판별한다.
Type	System
Cross References	3.1.2, 1.2.1
Notes	
Exception	하중이 꽉 차거나 초과한 경우 하중 상태를 꽉참으로 전환한다.
Output	현재 하중 상태를 메시지로 리턴한다
Preconditions	
Postconditions	나온 하중 상태에 따라 엘리베이터의 운영에 변화가 일어난다.

12. Flux Calculation

Operation Name	handleRequestAboard()
Responsibilities	탑승 요청을 받고 그 요청을 엘리베이터에 할당하기 위해서 먼저 readQueue()를 통해 현재 큐를 파악하고 받은 요청과 현재 큐를 비교하여 수행시간을 계산, 그리고 수행시간이 작은 엘리베이터에 큐를 할당한다.
Type	System
Cross References	3.2.1, 1.1.1, 1.1.2, 1.1.4, 2.1.3, 3.1.1, 3.1.2, 3.2.2, 3.2.3
Notes	
Exception	같은 층에 대한 같은 방향의 탑승 요청이 이미 존재하면 handleCancelRequest() 수행

Output	enqueueOrder()를 통해 새로운 큐 노드를 큐에 할당한다.
Preconditions	탑승요청을 받았을 경우
Postconditions	엘리베이터가 큐를 읽어 명령을 수행함.

Operation Name	handleFullLoad()
Responsibilities	효율적인 엘리베이터 운영을 보장하기 위해 꼭 찬 엘리베이터가 있는 경우 존재하는 큐의 우선순위들을 조정하여 준다.
Type	System
Cross References	3.2.1, 1.1.1, 1.1.2, 1.1.4, 2.1.3, 3.1.1, 3.1.2, 3.2.2, 3.2.3
Notes	
Exception	
Output	QueueNode들의 우선순위 변화로 인한 큐 순서의 변화
Preconditions	꼭 찬 엘리베이터 존재
Postconditions	변경된 큐에 따라 엘리베이터가 운행하게 됨

Operation Name	handleSelectLevel()
Responsibilities	층선택 요청에 우선순위를 매겨 큐에 할당시켜 준다.
Type	System
Cross References	1.1.2, 2.1.3, 3.2.1
Notes	
Exception	해당 엘리베이터에 대해 같은 층선택 요청이 이미 존재하면 handleCancelRequest()를 대신 수행한다.
Output	enqueueOrder()를 통해 새로운 큐 노드를 큐에 할당한다.
Preconditions	층선택 요청을 받음
Postconditions	이후 엘리베이터가 큐를 읽어 명령을 수행함으로써 순차적으로 요청에 따라 움직이게 된다.

Operation Name	handleCancelRequest
Responsibilities	해당 요청에 대한 큐 노드를 큐에서 제거한다.
Type	System
Cross References	1.1.4, 3.2.4, 3.2.1
Notes	
Exception	
Output	해당 요청에 맞는 큐 노드가 큐에서 제거된다.
Preconditions	중복되는 탑승 요청이나 층선택 요청이 이미 존재

Postconditions	큐가 변경되어 엘리베이터가 그에 따른 명령을 수행한다.
-----------------------	--------------------------------

13. Enqueue Order

Operation Name	enqueueOrder()
Responsibilities	요청에 대한 큐 노드를 생성하여 큐에 할당해 준다.
Type	System
Cross References	3.2.2, 1.1.2, 1.1.4, 2.1.3, 3.2.1, 3.3.1, 3.3.2
Notes	
Exception	
Output	새로운 큐 노드가 생성되어 큐에 할당된다.
Preconditions	enqueueOrder()를 사용하는 기능에서의 호출
Postconditions	큐 노드의 추가에 따른 큐의 변화와 그로 인한 엘리베이터의 명령 수행

14. Dequeue Order

Operation Name	dequeueOrder()
Responsibilities	queue내의 명령을 수행하였거나 요청취소로 인해 명령을 더 이상 수행할 필요가 없을 경우 큐에서 해당 큐 노드를 제거하도록 한다.
Type	System
Cross References	3.2.3, 1.1.1, 1.1.4, 2.1.3, 3.2.1, 3.3.1, 3.3.2
Notes	
Exception	
Output	해당 큐 노드가 큐에서 제거된다.
Preconditions	dequeueOrder()를 호출하는 기능(시나리오: 요청 취소, 요청 수행, 비상 상황 등)들에서의 호출
Postconditions	큐 노드의 제거에 따른 큐의 변화와 그로 인한 엘리베이터의 명령 수행

15. Read Queue

Operation Name	readQueue()
Responsibilities	Flux Calculator나 Movement Control 등 큐 내용을 읽어야 할 필요가 있을 경우에 수행되는 기능으로 큐의 내용을 읽도록 한다.
Type	System
Cross References	3.2.1, 3.2.4, 2.1.3
Notes	
Exception	
Output	큐 내의 노드들에 대한 순차적인 정보

Preconditions	readQueue()를 호출하는 기능 들에서의 호출
Postconditions	큐를 읽어 요청한 기능에 되보내줌

16. Fire Control

Name	fireControl()
Responsibilities	엘리베이터가 화재상황에 적합한 운영을 하도록 Queue 에 존재하는 모든 QueueNode 를 제거하고, 적합한 QueueNode 를 넣어준다
Type	System
Cross Reference	1.2.1, 3.2.2, 3.2.3 1.1.1, 1.1.2, 1.1.3, 1.1.4
Exceptions	Fire_Control()을 우선적으로 수행한다.
Output	1. Queue 에 존재하는 모든 QueueNode 의 삭제 2. Queue 에 상황에 따른 QueueNode 를 넣어줌
Pre-condition	Determine_State 의 값이 화재값과 동일함.
Post-condition	Queue 에 존재하는 모든 QueueNode 가 삭제되고, 새로운 QueueNode 를 넣어준다. 또한 Determine_State 값이 정상값이 될 때까지 들어오는 모든 요청을 무시한다.

17. Blackout Control

Operation Name	blackoutControl()
Responsibilities	엘리베이터가 정전상황에 적합한 운영을 하도록 Queue에 존재하는 QueueNode를 제거하고, 적합한 QueueNode를 넣어준다.
Type	System
Cross References	3.3.2, 3.2.2, 3.2.3, 3.2.4, 3.2.1, 3.1.1
Notes	
Exception	비상상황 하나만 있을 때 blackoutControl()을 수행한다.
Output	Queue 에 존재하는 모든 QueueNode 의 삭제 Queue에 상황에 따른 QueueNode를 넣어줌
Preconditions	Determine_State의 값이 정전값과 동일함.
Postconditions	Queue에 존재하는 모든 QueueNode가 삭제되고, 새로운QueueNode를 넣어준다. 또한 Determine_State값이 정상값이 될 때까지 들어오는 모든 요청을 무시한다.

18. Door Control Calculation

Name	wait()
Responsibilities	문이 열려있으면서 사용자의 문 제어 요청이 없는 상태일 때 정한 시간만큼 대기한다.
Type	System
Cross Reference	3.4
Exceptions	대기하는 중 문 제어 요청이 들어왔을 시 handleDoorRequest()를 호출하고 대기를 중단한다.
Output	지정한 시간만큼 대기한다.
Pre-condition	문이 열림
Post-condition	대기 시간 종료 후 다음으로 호출되는 동작 수행

Name	handleDoorRequest()
Responsibilities	DoorRequest()에서 사용자의 문 제어 요청이 들어왔을 때 사용자의 요청을 반영하기 적합한지 확인하고 맞을 시 문을 열거나 닫는다.
Type	System
Cross Reference	3.4, 1.1.3
Exceptions	사용자의 요청이 반영되기 적합한 상황이 아닐 때(e.g. 엘리베이터가 이동 중일 때) 사용자의 요청을 문 제어로 실행하지 않는다.
Output	사용자의 제어 요청값에 따라 문이 열리거나 닫힘(openDoor() 혹은 closeDoor() 호출)
Pre-condition	
Post-condition	openDoor() 혹은 closeDoor() 호출로 문이 열리거나 닫힘

Name	arrivalDoorCalibration()
Responsibilities	엘리베이터가 정차할 층에 도착해 문을 열어야 될 때 호출되어 엘리베이터가 정차하였는지 확인 후 문을 연다(openDoor())를 호출한다.)
Type	System
Cross Reference	2.1.3, 3.4
Exceptions	정지 상태가 아닐 경우 문을 열지 않는다.
Output	openDoor() 호출
Pre-condition	엘리베이터 도착 직후
Post-condition	문이 열림

Name	departDoorCalibration()
Responsibilities	문이 열려 있을 때 wait() 호출 후 문을 자동으로 닫도록 closeDoor()를 호출하고, 문을 닫았을 때 다시 wait()호출 후 Movement Control 에 출발을 해도 된다는 메시지를 전달한다.
Type	System
Cross Reference	2.1.3, 3.4
Exceptions	wait() 중간에 handleDoorRequest()가 요청되면 departDoorCalibration()을 중단하고 요청을 수행한다.
Output	closeDoor() 호출과 Movement Control 에 door = closed 전달
Pre-condition	엘리베이터가 정차 중
Post-condition	엘리베이터 문이 닫힌 뒤 wait() 진행 후 엘리베이터 출발

2037. State Diagrams Definition

N/A