

INTELLIGENT CONTROLLER

제안부서 T4

팀원 장재원, 이정우

1. Motivation

산업이 발전함에 따라서 도심으로 사람들이 몰리며 사람들이 몰리는 특정 지역이 생기기 시작했다. 그런 지역에는 용적당 이용 인구 비율이 높기 때문에 한정된 면적을 잘 사용하는 것에 많은 관심이 생겼고, 건축 기술의 발전과 맞물려 건축물들의 고층화가 이루어졌다. 높은 건물이 그 도시의 상징이라 불릴 수 있을 정도의 의미로 발전하게 되었고 효율적인 용적 이용을 보여준 반면에 그로 인해 건물 내의 이동시간이 늘어나면서 시간에 대한 관리가 어려워지게 되었다. 건물이 높아짐에 따라서 높은 층을 이용하는 사람은 계단으로 다니기 힘들어지면서 엘리베이터를 이용하여 조금 더 빠르고 편안하게 높은 층으로 갈 수 있다. 하지만 건물의 높이의 증가에 의해 엘리베이터를 기다리는 시간이 길어지며 고층일수록 그 시간은 더욱 길어짐에 따라서 인공지능 엘리베이터의 시스템을 도입하여 기존의 엘리베이터의 이용시간을 단축시켜 고층 건물을 이용하는 사람들에게 많은 도움이 될 것이다.

2. Project Objectives

- 보통 엘리베이터의 기능을 포함해야 한다.
- 엘리베이터의 이동 속도를 향상하여 빠르게 원하는 층으로 이동 할 수 있다.
- 값을 대입하는 알고리즘을 도입하여 엘리베이터를 기다리는 시간을 단축 시킨다.
- 엘리베이터의 위치와 엘리베이터가 멈추는 층수를 디스플레이 한다.
- 엘리베이터의 센서를 이용하여 문이 열고 닫히는 속도를 조절 한다.
- 특정 엘리베이터는 사용 빈도가 제일 많은 층으로 직행 할 수 있도록 한다.



3. Functional Requirements

- 층별 방향 선택
- 층별 디스플레이
- 층 선택
- 문 센서
- 문 제어
- 문 속도
- 문 열기/닫기 선택
- 천장 센서
- 무게 센서
- 엘리베이터 디스플레이
- 엘리베이터 제어
- 엘리베이터 속도
- 대기시간 최적화 알고리즘
- 선택 층 패스 알고리즘
- 직행 엘리베이터 전환

4. Non-Functional Requirements

- 엘리베이터 이용자가 기다리는 시간을 최소화 한다.
- 엘리베이터 탑승 시간이 1분이 넘지 않도록 한다.
- 시스템의 유지 보수가 쉬워야 한다.
- 커스터마이징이 쉬워야 한다.
- 디스플레이에 나오는 정보를 쉽게 이해 할 수 있어야 한다..



5. Resource Estimation

- Human Efforts (Man-Month) : 2
- Human Resource : 2명
- Project Duration : 6주
- Cost : 20,000,000



1. Alternative Solutions

- 기존의 엘리베이터의 기능에 새로 고층 건물에 적합한 기능을 추가한다.
- 고층 건물에 사용되고 있는 엘리베이터를 들여온다.
- 고층 건물에 사용 되는 엘리베이터의 기능 및 효율성을 조사 후 단점을 보완한다.

2. Project Justification

- Cost : 50,000,000
- Duration : 12주
- Risk : 비용과 시간이 많이 든다.
- Effect : 수익성을 높일 수 있다..

3. Risk Management

Risk	Probability	Significance	Weight
First Adoption of OSP	4	5	20
Lack of Elevator technology	4	5	20
Lack of C	4	3	12
Lack of UML	4	3	12
Team Communication	2	3	6



4. Risk Reduction Plan

- First Adoption of OSP (20)

OSP에 대한 많은 자료를 참고하고 교수님께 자문을 구한다.

- Lack of Elevator technology (20)

인터넷이나 책을 통한 자료를 조사하며 엘리베이터를 직접 이용 해 본다.

- Lack of C (12)

인터넷과 책을 통한 자료를 참고하며 수업시간에 배운 내용을 바탕으로 공부한다.

- Lack of UML (12)

인터넷과 책을 통한 자료를 참고하며 사용자에게 자문을 구해본다.

- Team Communication (6)

많은 대화를 통하여 프로젝트에 대한 의견을 나누며 자신의 일에 책임을 다한다.

5. Market Analysis

- 고층 건물에 설치되어 있는 엘리베이터를 사용 후 부족한 부분을 추가한다.

6. Other Managerial Issues

- OSP를 이용하여 성공적인 설계를 마친다.



1. Functional Requirements

- 층별 방향 선택 : 각 층에서 엘리베이터를 이용하여 다른 층으로 이동하고자 할 경우 버튼을 누른다. 현재 층보다 높은 층은 위쪽 낮은 층은 아래쪽 버튼을 이용한다.
- 층별 디스플레이 : 각 층에 설치되어 있으며 현재 엘리베이터의 층을 표시하고 엘리베이터 내에서 선택이 되어 있는 층이 어느 층인지 디스플레이 해 준다.
- 층 선택 : 엘리베이터 내에서 가고자 하는 층에 해당하는 버튼을 누른다.
- 문 센서 : 각 층의 문 앞, 엘리베이터 문 앞, 문 사이에 설치 되어 있으며 센서가 물체를 판단하여 문 근처에 물체가 있는지 없는지를 판단한다.
- 문 제어 : 문을 열고 닫고를 제어한다. 속도의 입력을 받아서 열고 닫는 속도를 증가 감소 시킨다.
- 문 속도 : 센서에 대한 입력을 받아서 문에 대한 속도를 계산하여 문 제어를 하도록 한다.
- 문 열기/닫기 선택 : 엘리베이터 내에서 문을 열고 닫는 버튼을 조작할 경우 그에 맞는 행동을 한다.
- 천장 센서 : 엘리베이터에 물체나 사람이 얼마나 많은 공간을 차지하고 있는가를 판단한다.
- 무게 센서 : 엘리베이터에 물체나 사람이 가지고 있는 총 무게를 잴다.
- 엘리베이터 디스플레이 : 엘리베이터 내에서 현재 어느 층에 위치하고 있는 지를 판단한다.
- 엘리베이터 제어 : 엘리베이터의 이동에 대해서 제어를 하며 속도를 입력으로 받아서 이동 속도의 증가 감소를 제어한다.
- 엘리베이터 속도 : 무게 센서를 입력으로 받아서 엘리베이터의 속도를 제어한다. 무게가 없을 경우 빠른 이동을 하며 빠른 이동 시에는 2층의 여유가 있어야 신호 입력에 대해서 그 층에 정지 한다.
- 대기시간 최적화 알고리즘 : 각 층에서 버튼을 누를 경우 알고리즘에 의해서 가장 빨리 올 수 있는 엘리베이터가 오도록 한다.
- 선택 층 패스 알고리즘 : 천장 센서와 무게 센서를 입력으로 받아서 사람이 더 탈수 없는 무게이거나 자리가 없을 경우 해당 층에 정지하지 않고 넘어가 버린다.
- 직행 엘리베이터 전환 : 특정 엘리베이터는 고층으로 이동 했을 경우 가장 사용이 많은 층으로 직행하는 시스템으로 전환하여 그 층으로 가는 사용자에게 한하여 정지한다. 직행 엘리베이터로 전환했을 경우 각 층에 있는 디스플레이에 표시한다.



2. System Functions

Ref. #	Function	Category
R 1.1.1	층별 방향 선택	Evident
R 1.1.2	층별 디스플레이	Evident
R 1.2.1	문 센서	Hidden
R 1.2.2	문 제어	Hidden
R 1.2.3	문 속도	Evident
R 1.2.4	문 열기/닫기 선택	Evident
R 1.3.1.1	천장 센서	Hidden
R 1.3.1.2	무게 센서	Hidden
R 1.3.2.1	엘리베이터 디스플레이	Evident
R 1.3.2.2	엘리베이터 제어	Hidden
R 1.3.2.3	엘리베이터 속도	Evident
R 1.3.2.4	층 선택	Evident
R 2.1.1	대기시간 최적화 알고리즘	Hidden
R 2.1.2	선택 층 패스 알고리즘	Hidden
R 2.2	직행 엘리베이터 전환	Frill

3. Performance Requirements

- 탑승 시간이 1분이 넘지 않도록 한다. (폐쇄된 공간에서 1분이 넘지 않는다.)



1003

Define Requirements

4. Operation Environments

Hardware : Embedded Controller
Software : Controlling Software

5. 개발환경

- OS : Windows XP
- Tool : Microsoft VC

6. Interface Requirements

- 각 층에서 이동하고자 하는 방향을 누를 경우 시작된다.
- 엘리베이터 내에서 원하는 층을 누른다.
- 문을 열고 닫고를 임의로 수행 할 수 있다.

7. Other Requirements

- 엘리베이터 사용을 할 경우 항상 사용자의 안전을 보장해야 한다.



Term	Description	Remarks
층별 디스플레이	각 층에서 엘리베이터 내에서 선택된 층과 현재 엘리베이터가 있는 층을 사용자에게 보여줌	
문 센서	각 층의 문 앞과 엘리베이터 문 앞에서 물체가 있는지를 판단	
천장 센서	현재 엘리베이터의 면적 사용을 확인	
직행 엘리베이터	가장 많이 쓰이는 층으로 직행하는 엘리베이터	

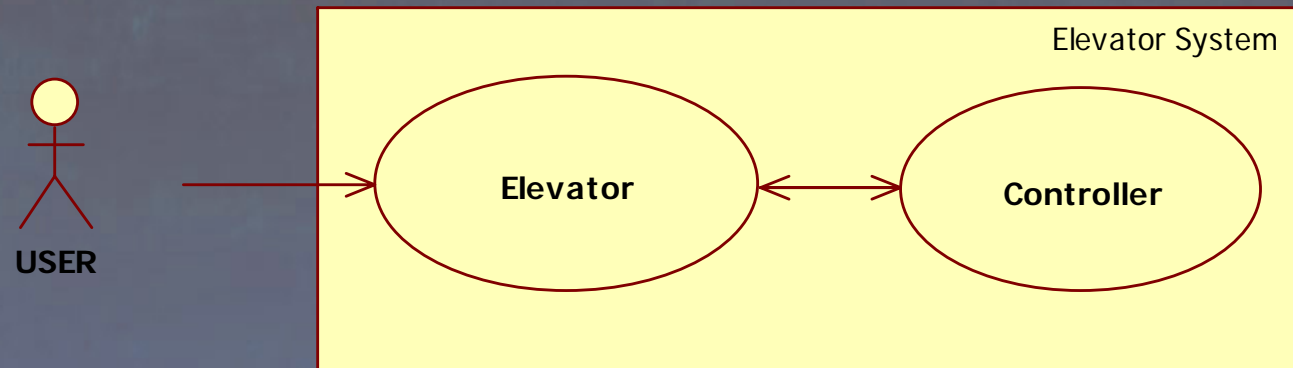


1005

Implement Prototype



1. Define System Boundary

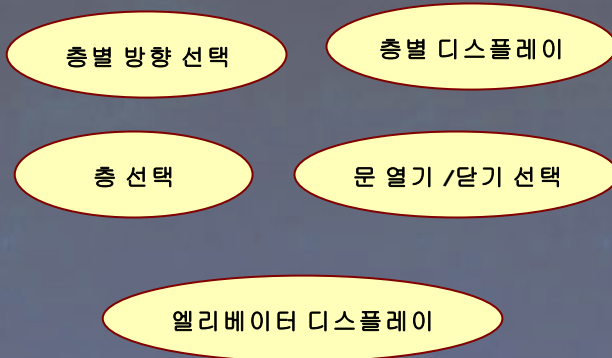


2. Identify and Describe Actors

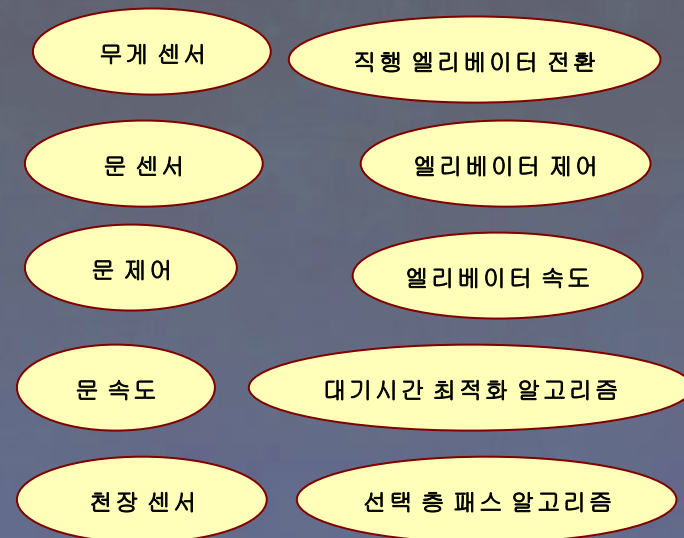
- 사용자 : 엘리베이터 시스템에 입력을 주어 엘리베이터를 사용하는 객체

3. Identify Use-Case

- Use-case by actor-based



- Use-case by event-based



3. Allocate system functions into Related Use-Case

Ref. #	Function	Remarks
R 1.1.1	층별 방향 선택	
R 1.1.2	층별 디스플레이	
R 1.2.1	문 센서	
R 1.2.2	문 제어	
R 1.2.3	문 속도	
R 1.2.4	문 열기/닫기 선택	
R 1.3.1.1	천장 센서	
R 1.3.1.2	무게 센서	
R 1.3.2.1	엘리베이터 디스플레이	
R 1.3.2.2	엘리베이터 제어	
R 1.3.2.3	엘리베이터 속도	
R 1.3.2.4	층 선택	
R 2.1.1	대기시간 최적화 알고리즘	
R 2.1.2	선택 층 패스 알고리즘	
R 2.2	직행 엘리베이터 전환	



1006

Define Business Use-Case

5. Categorize Use-Cases

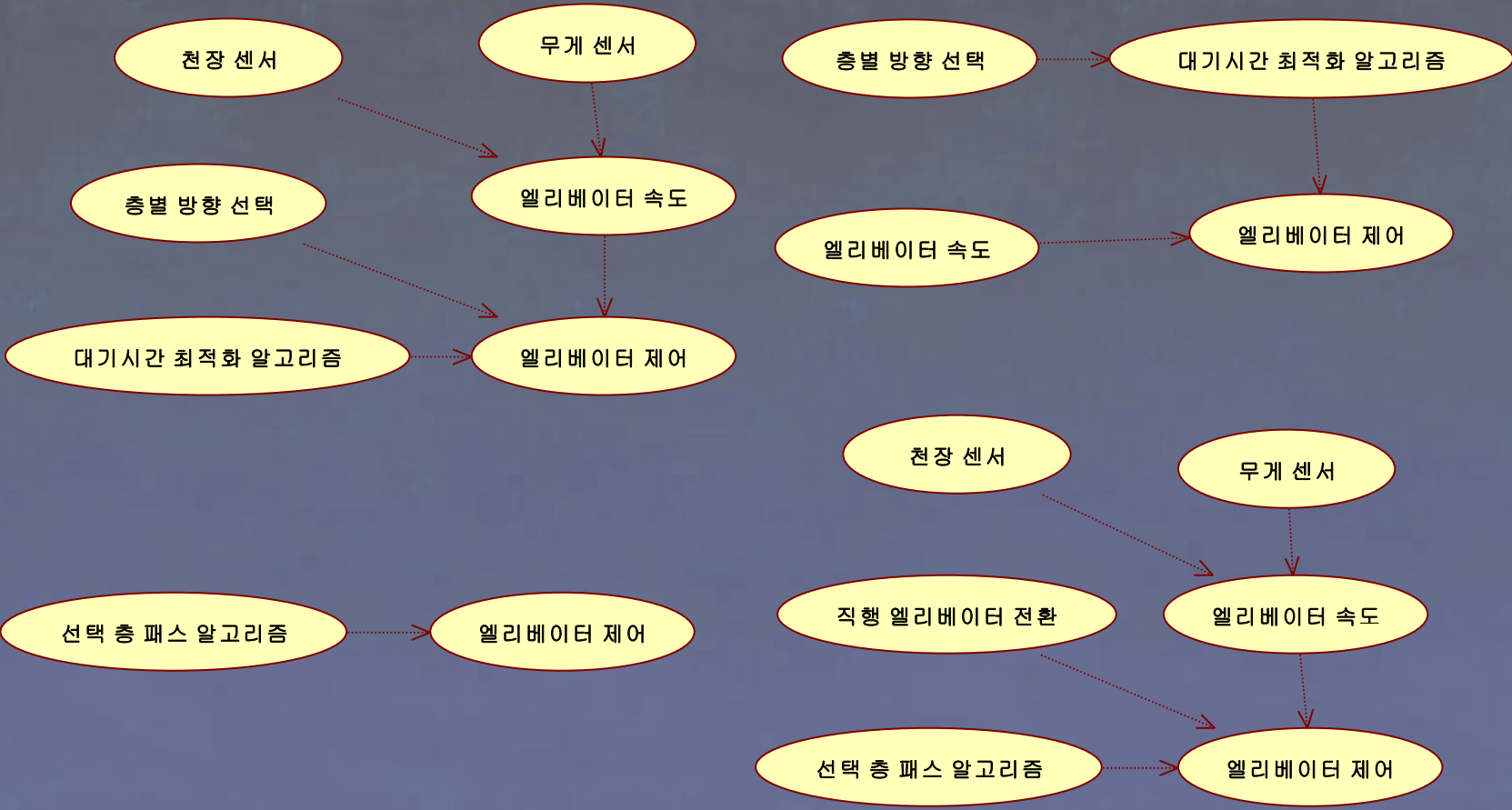
Use-case Number & Name	Category
1. 층별 방향 선택	Primary
2. 층별 디스플레이	Primary
3. 문 센서	Primary
4. 문 제어	Primary
5. 문 속도	Secondary
6. 문 열기/닫기 선택	Primary
7. 천장 센서	Secondary
8. 무게 센서	Primary
9. 엘리베이터 디스플레이	Primary
10. 엘리베이터 제어	Primary
11. 엘리베이터 속도	Secondary
12. 층 선택	Primary
13. 대기시간 최적화 알고리즘	Secondary
14. 선택 층 패스 알고리즘	Secondary
15. 직행 엘리베이터 전환	Secondary



6. Identify the relationships between Use-Case



6. Identify the relationships between Use-Case



7. Draw a Use-Case diagram



8. Describe Use-Case

- Name 1 : 층별 방향 선택

Actors : User

Description :

- User로 부터 입력을 받아서 엘리베이터를 입력이 들어온 층으로 오도록 한다.
- 대기 시간 최적화 알고리즘의 입력과 출력에 따라서 작동한다.
- User로 부터의 입력은 2가지이다.
- 엘리베이터 속도 use-case 의 입력이 된다.

- Name 2 : 층별 디스플레이

Actors : None

Description :

- 엘리베이터의 위치를 숫자로 보여준다.
- 엘리베이터가 정지 할 층들을 숫자로 보여준다.
- 엘리베이터의 위치와 정지 할 층을 분리하여 보여준다.
- 각 엘리베이터 별로 다르다.
- 엘리베이터 내의 층 선택을 입력 값으로 받는다.

- Name 3 : 문 센서

Actors : None

Description :

- 문이 열릴 때와 닫힐 때의 이벤트에 의해서 발생한다.
- 각 층의 문 앞과 각 엘리베이터의 문 앞에 있는 센서는 움직임을 감지한다.
- 위의 두 센서는 움직임을 감지 했을 경우 문 속도에 감속 이벤트를 발생 시킨다.
- 문 안쪽의 센서는 움직임을 감지한다.
- 문 안쪽 센서에 움직임을 감지 된 경우 문 제어에 열기 이벤트를 발생 시킨다.

- Name 4 : 문 제어

Actors : None

Description :

- 문 속도에서 입력을 받아서 문을 열고 닫는 역할을 한다.
- 문 열기/닫기 선택의 이벤트를 받아서 수행한다.
- 문 센서에서 안쪽 센서로부터 이벤트를 받아서 수행한다.
- 문 제어 수행이 끝나기 전까지 문 속도에서 이벤트를 받을 수 있다.



8. Describe Use-Case

- Name 5 : 문 속도

Actors : None

Description :

- 각 층의 센서와 엘리베이터 문 앞의 센서를 입력으로 받는다.
- 센서에 움직임이 감지되지 않은 경우 빠른 속도로 설정 된다.
- 센서에 움직임이 있을 경우 보통 속도로 설정이 된다.
- 속도 설정에 변화가 있을 경우 문 제어에 이벤트를 발생 시킨다.

- Name 6 : 문 열기/닫기 선택

Actors : User

Description :

- User로 부터 입력을 받는다.
- 열기 버튼과 닫기 버튼 2가지가 있다.
- 열기 버튼을 누른 경우 문 센서의 영향을 받지 않고 열린다.
- 닫기 버튼의 경우 문 센서 중 문 안쪽 센서에 움직임이 없어야 한다.
- User가 버튼을 누른 경우 문 제어에 이벤트를 보낸다.

- Name 7 : 천장 센서

Actors : None

Description :

- 엘리베이터 속도에 입력이 된다.
- 센서의 반사로 현재 엘리베이터의 빈 공간을 확인한다.

- Name 8 : 무게 센서

Actors : None

Description :

- 엘리베이터 속도에 입력이 된다.
- 무게를 측정하여 빈 경우와 한도 초과의 경우를 확인한다.



8. Describe Use-Case

- Name 9 : 엘리베이터 디스플레이
Actors : None
Description :
 - 층 선택으로부터 입력을 받는다.
 - 현재 엘리베이터의 위치를 숫자로 나타내어 준다.
 - 현재 엘리베이터가 정지해야 할 층을 표시한다.
 - 위치와 정지해야 할 층을 따로 표시하여 구분하여 준다.
 - 엘리베이터 내에서 선택이 된 모든 층과 층별 방향 선택에서 알고리즘에 의해서 선택이 된 엘리베이터에 한해서 추가적으로 표시하여 준다.

- Name 10 : 엘리베이터 제어
Actors : None
Description :
 - 엘리베이터의 속도를 입력 받는다.
 - 선택 층 패스 알고리즘과 대기시간 최적화 알고리즘에 의해서 이동한다.
 - 이동 중에는 알고리즘이 적용 되지 않는다.
 - 정지 후 이동 할 경우 알고리즘의 결과에 따라 이동한다.

- Name 11 : 엘리베이터 속도
Actors : None
Description :
 - 무게 센서와 천장 센서를 입력으로 받는다.
 - 엘리베이터에 아무것도 없을 경우 빠른 속도로 설정이 된다.
 - 엘리베이터에 User가 있을 경우 보통 속도이다.

- Name 12 : 층 선택
Actors : User
Description :
 - User가 누른 층 버튼을 입력으로 받는다.
 - 입력이 되어 있는 층을 경우 한번 더 입력을 받을 경우 그 층은 취소가 된다.
 - 버튼이 눌린 경우 엘리베이터 디스플레이와 층별 디스플레이에 이벤트를 발생한다.



8. Describe Use-Case

- Name 13 : 대기시간 최적화 알고리즘
Actors : None
Description :
 - User가 층별 방향 선택을 한 경우 입력을 받는다.
 - 엘리베이터의 선택 된 층과 위치와 방향을 입력 받아서 알고리즘을 계산한다.
 - 입력 값에 따른 알고리즘의 출력은 엘리베이터 제어에 이벤트를 발생 시킨다.

- Name 14 : 선택 층 패스 알고리즘
Actors : None
Description :
 - 천장 센서와 무게 센서를 입력을 받는다.
 - 층별 방향 선택을 입력으로 받는다.
 - 천장 센서의 결과에 의해서 더 이상 자리가 없는 경우 FULL 이 된다.
 - 무게 센서의 결과에 의해서 초과 무게가 된 경우 FULL 이 된다.
 - FULL 일 경우 엘리베이터 내에서 선택 된 층이 아닌 경우 정지하지 않는다.

- Name 15 : 직행 엘리베이터 전환
Actors : None
Description :
 - 직행 엘리베이터 전환 이벤트가 일어나지 않은 경우 일반 엘리베이터와 같다.
 - 직행 엘리베이터로 전환이 가능한 엘리베이터만 영향을 받는다.
 - 지정된 층 이상일 경우 직행 엘리베이터 전환 이벤트가 발생한다.
 - 직행 엘리베이터의 경우 지정된 층이나 사용 빈도가 높은 층으로 직행한다.
 - 전환한 경우 층별 디스플레이에 표시가 된다.
 - 직행 층에 도착한 경우 일반 엘리베이터로 전환 된다.
 - 직행 일 경우 층별 방향 선택은 입력으로 층 선택은 입력 받지 않는다.
 - 문 열기/닫기 선택은 입력으로 받는다.



9. Rank Use-Case

Rank	Function
High	층별 방향 선택
Medium	층별 디스플레이
Medium	문 센서
High	문 제어
Medium	문 속도
High	문 열기/닫기 선택
Medium	천장 센서
Medium	무게 센서
Medium	엘리베이터 디스플레이
High	엘리베이터 제어
Medium	엘리베이터 속도
High	층 선택
Medium	대기시간 최적화 알고리즘
Medium	선택 층 패스 알고리즘
Low	직행 엘리베이터 전환



층별 방향 선택

직행 엘리베이터 전환

천장 센서

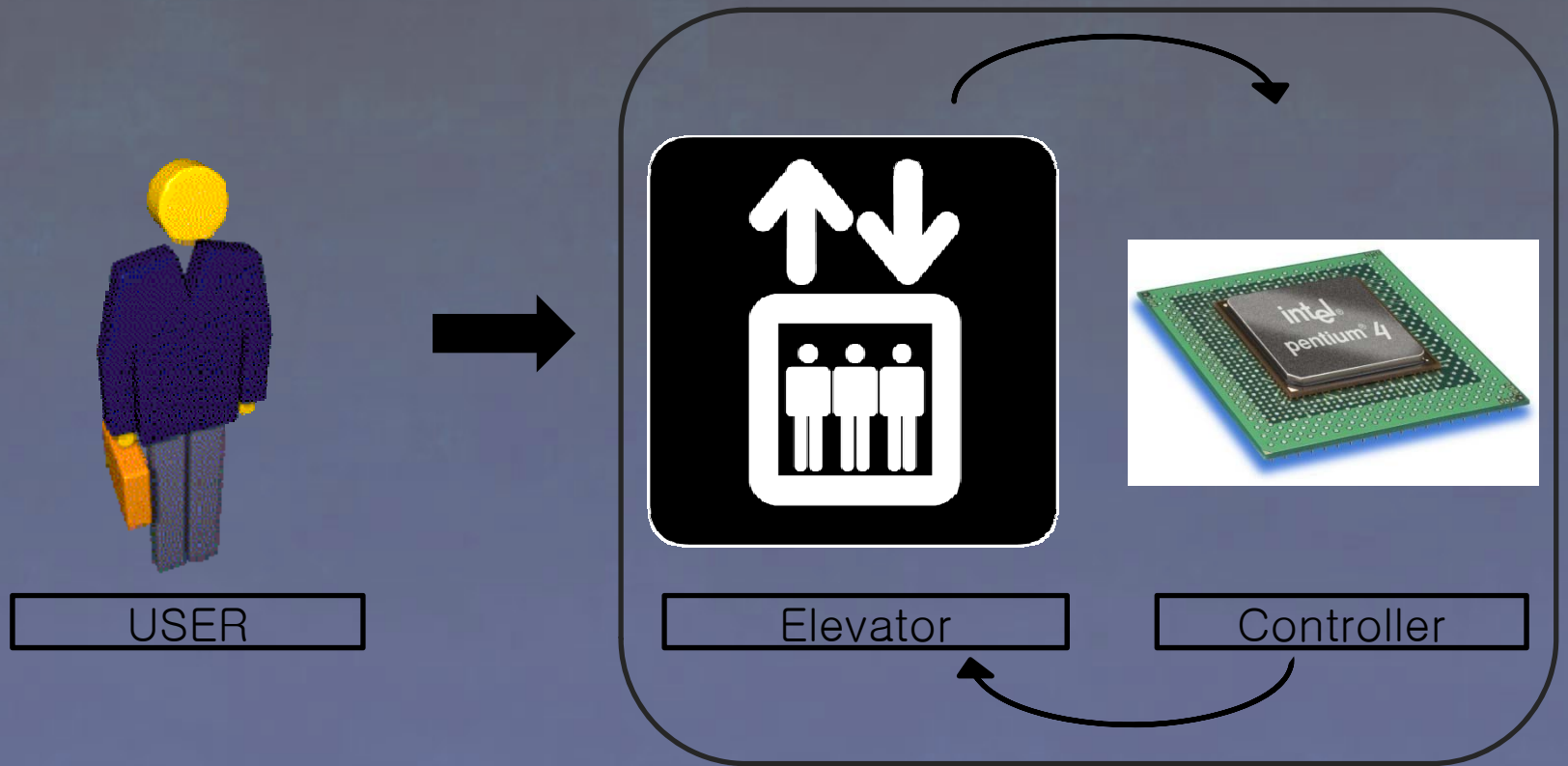
엘리베이터 디스플레이

무게 센서

층별 디스플레이

문 센서





1. Project Scope

- 현재 사용 되고 고층 건물의 일반 엘리베이터 보다 기다리는 시간과 이동하는 시간을 단축 시키기 위한 기능을 추가한 인공지능 엘리베이터 시스템이다.

2. Project Objectives

- 이동 속도 향상, 새로운 알고리즘 도입, 센서 이용, 직행 시스템 등을 이용하여 건물의 층 수가 높아 질수록 엘리베이터 이용 시간이 줄어 들 수 있도록 한다.

2. Functional Requirements

- 층별 방향 선택
- 층별 디스플레이
- 층 선택
- 문 센서
- 문 제어
- 문 속도
- 문 열기/닫기 선택
- 천장 센서
- 무게 센서
- 엘리베이터 디스플레이
- 엘리베이터 제어
- 엘리베이터 속도
- 대기시간 최적화 알고리즘
- 선택 층 패스 알고리즘
- 직행 엘리베이터 전환



4. Performance Requirements

- 엘리베이터 이용자가 기다리는 시간을 최소화 한다.
- 엘리베이터 탑승 시간이 1분이 넘지 않도록 한다.
- 시스템의 유지 보수가 쉬워야 한다.
- 커스터마이징이 쉬워야 한다.
- 디스플레이에 나오는 정보를 쉽게 이해 할 수 있어야 한다.

5. Operating Environment

Hardware : Embedded Controller
Software : Controlling Software

6. User Interface Requirements

- 각 층에서 이동하고자 하는 방향을 누를 경우 시작된다.
- 엘리베이터 내에서 원하는 층을 누른다.
- 문을 열고 닫고를 임의로 수행 할 수 있다.

7. Other Requirements

- 안정성이 최우선 이어야 한다.
- User의 입력 값에 따라서 컨트롤러가 엘리베이터를 제어 해야 한다

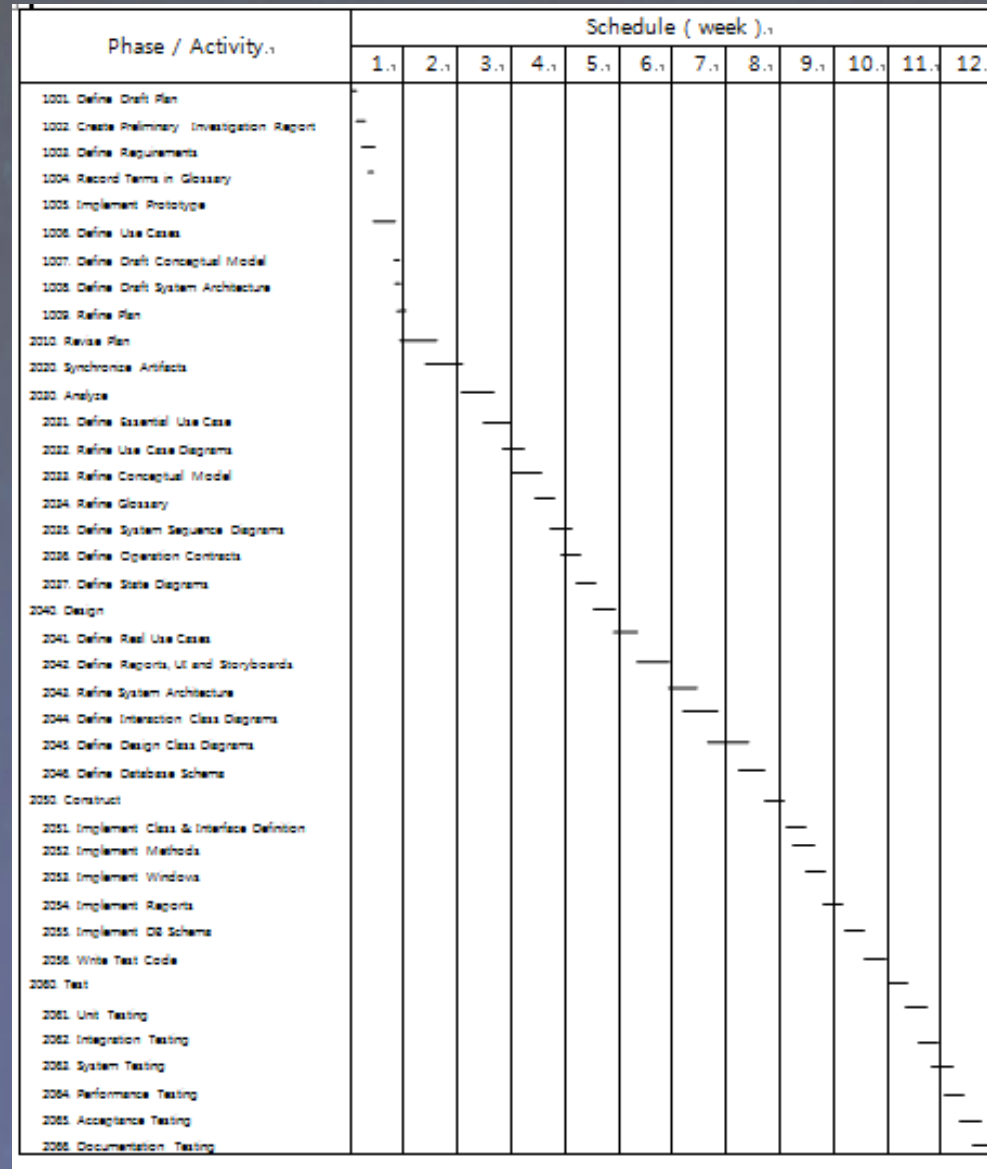


8. Resources

- Human Efforts (Man-Month) : 2
- Human Resource : 2명
- Duration : 12주
- Elevator System
 - Hardware : Embedded Controller
 - Software : Controlling Software
- 개발 환경
 - OS : Windows XP
 - Tool : Microsoft VC



9. Scheduling



10. Configuration Management

- 설치 될 건물의 크기와 용도에 의해서 커스터마이징 된다.

11. Quality Assurance Plan

- 기본적인 엘리베이터의 기능을 완벽하게 수행 한다.
- 추가 기능은 용도에 맞게 CUSTOMIZE 한다.
- 엘리베이터의 시간 절약에 대한 프로젝트이지만 이용 안전에 최우선을 둔다..

