

IoT기반 승강기 제어장치 설계

하지영, 박준호, 안은지, 조경록, 이은서*
안동대학교 컴퓨터공학과

e-mail : ja05065@naver.com, wnsghw@naver.com, enji25@naver.com,
chokr0216@naver.com, eslee@andong.ac.kr*

IoT-based Elevator Control Device Design

Ji-Yeong Ha, Jun-Ho Park, Eun-Ji Ahn, Gyeong-rok Cho, Eun-Ser Lee*
Dept of Computer Engineering, Andong National University

요 약

사물인터넷을 기반으로 실시간으로 승강기의 상태를 확인하고 제어가 가능한 웹 관리 시스템과 애플리케이션을 설계하였다. 웹 관리 시스템에서는 회원 정보의 수정과 열람, 승강기의 상태확인 및 변경을 할 수 있으며 애플리케이션에서는 회원 등록과 승강기의 호출 및 상태확인이 가능하다. 본 논문에서는 승강기 제어장치의 설계 부분인 UML(Unified Modeling Language)을 활용한 여러 다이어그램과 애플리케이션의 UI를 볼 수 있다.

1. 서론

고층빌딩이 많이 들어서게 되면서 승강기는 없어서는 안 되는 것이 되었다. 승강기는 편리하게 사용되고 있지만, 종종 승강기를 이용하다 사고를 당하게 된다. 올해 여름 강한 장맛비가 쏟아지면서 관동동 일대에 정전이 발생하게 되었고, 그로 인해 한 아파트에서 1분 간격으로 승강기에 갇혔다는 신고가 2건 접수됐다. [1] 소방당국이 출동 중 전원이 들어오면서 별다른 인명 피해는 발생하지 않았지만, 만약 계속 전원이 들어오지 않은 상태로 마땅히 사고 상황을 알릴 수 없는 상황이 된다면 굉장히 난처한 상황이 되었을 것이다. 또한, 승강기를 사용하다 보면 출근 시간에 여러 사람이 사용하게 되어 승강기에서 많은 시간을 소비하게 된다. 그러다 보면 지각을 면하지 못하는 사람들도 생길 것이다. 이러한 피해들을 미리 방지하고자 생각한 것이 승강기 제어장치이다.

본 연구를 통하여 실시간으로 승강기의 상태를 확인하고, 제어함으로써 승강기의 고장을 빠르게 감지하고, 사고에 즉각 대응함으로써 피해를 줄이고, 더 편리하게 승강기를 사용할 수 있을 것으로 예상된다.

2. 관련 연구

본 논문에서 관련 연구 부분은 프로젝트에서 사용한 기반 연구들을 설명하는 것으로 승강기의 실시간 정보와 제

어를 위해 이용한 사물인터넷과 소프트웨어 설계를 위해 이용한 UML에 관해 설명한다.

2.1 IoT(Internet of Things)

사물인터넷(Internet of Things, 약어로 IoT)은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술. 즉, 무선 통신을 통해 각종 사물을 연결하는 기술을 의미한다. 인터넷으로 연결된 사물들이 데이터를 주고받아 스스로 분석하고 학습한 정보를 사용자에게 제공하거나 사용자가 이를 원격 조정할 수 있는 인공지능 기술이다. [1] 여기서 사물이란 가전제품, 모바일 장비, 웨어러블 디바이스 등 다양한 임베디드 시스템이 된다. 사물인터넷에 연결되는 사물들은 자신을 구별할 수 있는 유일한 아이피를 가지고 인터넷으로 연결되어야 하며, 외부 환경으로부터의 데이터 취득을 위해 센서를 내장할 수 있다. 모든 사물이 바이러스와 해킹의 대상이 될 수 있어 사물인터넷의 발달과 보안의 발달은 함께 갈 수밖에 없는 구조이다. [2]

2.2 UML (Unified Modeling Language)

UML(Unified Modeling Language)은 1994년 소프트웨어 방법론의 선구자인 그래디 부치(Grady Booch), 제임스 럼바(James Rumbaugh), 이바 야콥슨(IVar Jacobson)에 의해 연구되었다. 1997년 객체관리그룹에서 여러 표기법을 통합하여 UML을 발표하였고, 객체지향 시스템 개발 분야에서 가장 우수한 모델링 언어로 인식되고 있다. UML은 시스템을 만들기 전에 어휘와 규칙을 마련하여 시스템을 개념적, 물리적으로 표현하는 모델이 필요하여 사용되고 있다. UML이 모델링 언어로 사용될 수 있는 대표적인 시스템의 유형으로는 정보 시스템, 기술적 시스템, 내장 시

* 본 논문의 교신저자임.

** 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학지원사업의 연구결과로 수행되었음. (2019-0-01113)

스택이 있다. [3][4][5]

3. 요구사항 분석

요구 분석은 소프트웨어 개발 생명주기의 첫 단계로, 현재의 상태를 파악하고 사용자가 잠재적으로 또는 명시적으로 원하는 요구를 파악한 후, 소프트웨어에 반영할 사용자의 요구를 결정하는 것이다. <표-1>은 승강기 제어장치의 요구사항을 분석한 후 표로 만든 것으로 크게 기능 요구사항과 비기능 요구사항으로 정리하였다.

아래의 <표-2>는 <표-1>을 기반으로 작성하였다. 요구사항 명세서는 고객과 개발자가 협의를 통하여 공통의 목표와 기대에 맞추어 소프트웨어 시스템에 대한 개발 기준선을 만든 문서이다. [6]

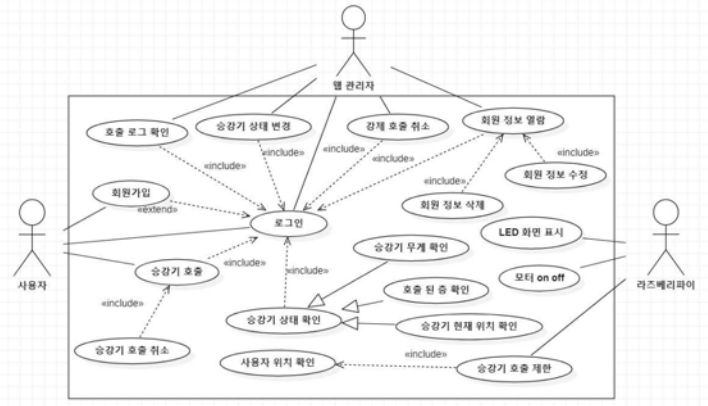
분류	고유번호	요구사항명칭	비고	
기능 요구사항	R-0001	회원가입		
	R-0002	로그인		
	R-0003	승강기 호출		
	R-0004	승강기 호출 취소		
	R-0005	승강기 현재 위치 확인		
	R-0006	승강기 상태확인		
	R-0007	승강기 호출제한		
	R-0008	승강기 무게 확인		
	R-0009	사용자 위치 확인		
	R-0010	승강기 상태 변경		
	R-0011	호출 로그 확인		
	R-0012	호출된 층 확인		
	R-0013	강제 호출 취소		
	R-0014	회원 정보 삭제		
	R-0015	회원 정보 수정		
	R-0016	회원 정보 열람		
	R-0017	모터 on, off		
	R-0018	LED 화면 표시		
	비 기능 요구사항	R-0017	모터 on, off	
		R-0018	LED 화면 표시	
R-0019		무게 제한		
	R-0020	층수 제한		

<표-1> 요구사항 정의서 분량상 문제로 요구사항 명세서 중 기능 요구사항인 승강기 호출을 예로 들었다. 중요도는 상·중·하로, 난이도는 기준표에 의거했다.

요구사항 ID	R-0003	요구사항 명	승강기 호출
개요	버튼을 눌러 승강기를 호출한다.		
요구사항 내역	상세 설명	- 어플리케이션의 버튼을 눌러 승강기를 호출한다.	
	유형	기능	
	중요도	상	난이도

<표-2> 요구사항 명세서

다음의 <그림-1>은 위의 요구사항 명세서를 기반으로 작성한 유스케이스 다이어그램이다. 유스케이스란 UML의 행위자와 액터가 요구하여 시스템이 수행하는 일의 목표이다. 유스케이스의 행위자와의 관계를 구조적으로 나타내며 시스템의 정적 유스케이스 뷰를 다룬다.[7] 액터는 사용자, 웹관리자, 라즈베리파이로 설정하였다.



<그림-2> 유스케이스 다이어그램

<표-3>은 미야방지 목걸이의 유스케이스 시나리오로써, 유스케이스 명, 액터, 선행조건, 선행입력, 이벤트 흐름, 후행조건, 후행 입력, 제약사항으로 나누어 작성하였다. 이것 또한 분량상의 이유로 앞에서 제시한 승강기 호출에 대한 유스케이스 시나리오를 내용을 예를 들었다.

유스케이스 시나리오 : 애플리케이션의 버튼을 눌러 승강기를 호출한다.

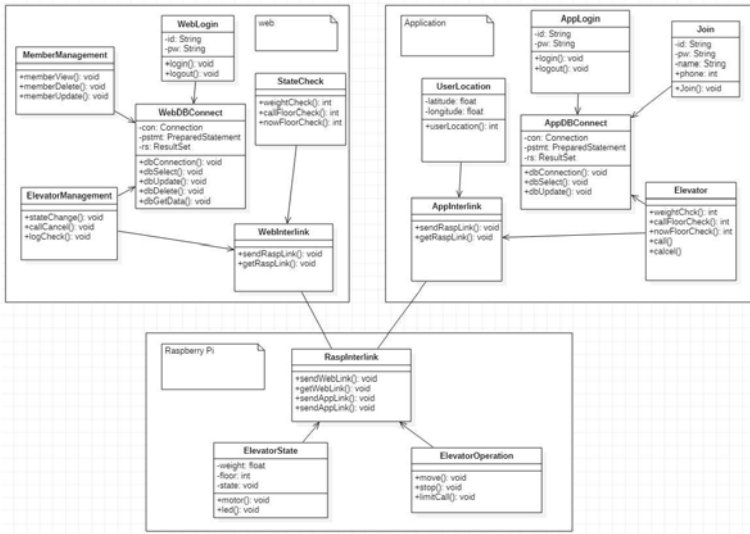
유스케이스명	승강기 호출
액터	사용자
선행조건	애플리케이션에 로그인되어 있는 상태이다.
선행입력	애플리케이션의 버튼을 누른다.
이벤트흐름	1. 사용자가 애플리케이션의 원하는 층의 버튼을 눌러 승강기를 호출한다. 2. 라즈베리파이에서 신호를 받고 호출된 층으로 승강기를 이동시킨다.
후행 조건	라즈베리 파이에서 신호를 받은 상태이다.
후행 출력	호출된 층에 승강기가 도착한다.
제약사항	사용자와 승강기의 거리

<표-3> 유스케이스 시나리오

4. 설계

본 프로젝트의 설계 부분은 UML 기법 중 클래스 다이어그램과 시퀀스 다이어그램, 그리고 상태 다이어그램을 사용한다. 클래스 다이어그램은 클래스, 인터페이스, 통신 그리고 이들의 관계들을 나타내며, 객체지향 시스템 모델링에서 가장 공통적으로 쓰이는 다이어그램이다. 시스템의

정적 설계 뷰를 다루며, 활성 클래스를 갖는 클래스 다이어그램은 시스템의 정적 프로세스 뷰를 다룬다. [3] 아래의 <그림-2>는 프로그램에 대한 클래스 다이어그램을 작성한 것이다.



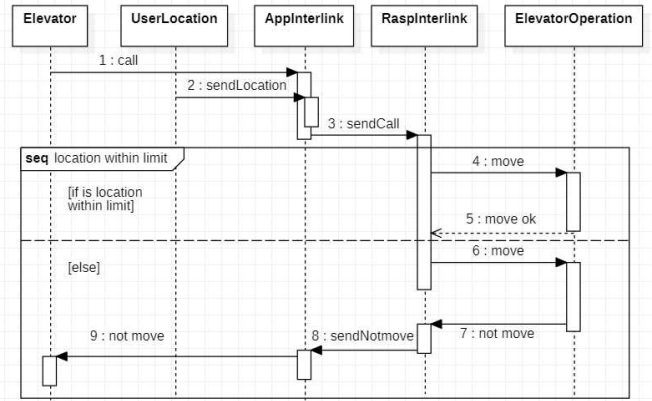
<그림-3> 클래스 다이어그램

웹 부분에서는 WebLogin, MemberManagement, WebDBConnect, ElevatorManagement, StateCheck, WebInterlink의 클래스가 있다. WebDBConnect 클래스는 WebLogin, MemberManagement, ElevatorManagement와 연관관계에 있으며, WebInterlink 클래스는 StateCheck 클래스와 ElevatorManagement 클래스와 연관관계에 있다.

애플리케이션 부분에서는 AppLogin, UserLocation, AppInterlink, AppDBConnect, Join, Elevator의 클래스가 있다. AppDBConnect 클래스는 AppLogin, Join, Elevator 클래스와 연관관계에 있으며, AppInterlink 클래스는 UserLocation, Elevator 클래스와 연관관계에 있다. 로그인할 수 있는 클래스는 웹과 애플리케이션 양쪽에 모두 있지만, 회원가입을 할 수 있는 Join클래스는 애플리케이션에만 있는 것을 확인 할 수 있다.

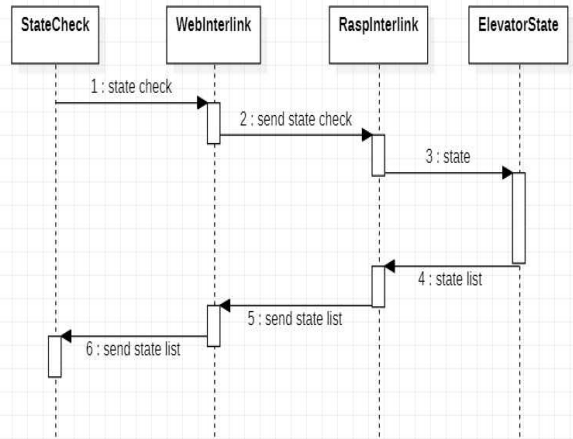
라즈베리파이에서는 RaspInterlink, ElevatorState, ElevatorOperation 클래스가 있으며, RaspInterlink 클래스는 ElevatorState 클래스와 ElevatorOperation 클래스와 연관관계에 있고, WebInterlink 클래스와 AppInterlink 클래스와는 연관관계에 있다는 것을 볼 수 있다. WebInterlink, AppInterlink, RaspInterlink 클래스는 각각 서로 다른 모듈과 연결할 수 있는 기능을 가지고 있다.

시퀀스 다이어그램은 시간 순서로 정렬된 객체 상호작용을 보여준다. 메시지의 시간적 순서를 강조하는 교류도이며, 시스템 외부 이벤트를 처리하기 위하여 시스템 내부 객체간에 주고받는 동적 메시지를 시간의 흐름에 따라 표현한 것이다. 아래의 <그림-3>과 <그림-4>는 승강기 제어장치의 시퀀스 다이어그램 중 '승강기 호출제한', '웹 승강기 상태확인' 시퀀스 다이어그램을 작성한 것이다.



<그림-4> 승강기 호출제한 시퀀스

'승강기 호출제한'시퀀스 다이어그램은 애플리케이션 사용자가 승강기 호출 버튼을 눌렀을 경우 사용자의 위치에 따라 승강기가 움직이는 기능의 시퀀스 다이어그램이다. Elevator에서 호출요청을 하면 UserLocation에서 사용자의 위치정보와 함께 AppInterlink에서 RaspInterlink로 호출요청을 하게 된다. 호출요청과 함께온 사용자 위치에 따라 ElevatorOperation에서 승강기를 움직이거나 움직일 수 없다는 메시지를 RaspInterlink를 통해 애플리케이션으로 보낸다.



<그림-5> 웹 승강기 상태 확인 시퀀스

'웹 승강기 상태 확인'시퀀스 다이어그램은 웹 관리자가 승강기 상태 확인 버튼을 통해 승강기의 상태를 확인 할 수 있는 시퀀스 다이어그램이다. StateCheck에서 상태를 확인 요청을 WebInterlink를 통해 RaspInterlink로 보내고 RaspInterlink에서 ElevatorState로 상태가 어찌지에 대한 요청을 보낸다. ElevatorState에서 승강기의 상태에 대한 정보를 RaspInterlink로 보내면 RaspInterlink에서 WebInterlink로 승강기 상태의 정보를 보내 웹 관리자가 승강기 상태에 대해 알 수 있게 한다.

상태 다이어그램은 객체지향모델에서 클래스의 인스턴스 사건에 의거한 작동을 보여주는 다이어그램이다. 상태

