

Company Logo	Document Name : 소프트웨어 아키텍처 설계서	Page : 1/22
--------------	-----------------------------------	----------------

S/W 기본설계서

(S/W High Level Design)

Project Name	빌딩보안 관리시스템		
Date	2015-01-01	Version	1.0
변경 이력			
작성자	홍길동	승인자	
조직명			

<p>Company Logo</p>	<p>Document Name : 소프트웨어 아키텍처 설계서</p>	<p>Page : 2/22</p>
----------------------------	--	------------------------

제정 및 개정 이력

버전	개정 내용	작성자	승인자	적용 날짜
1.0	최초 생성	홍길동		

<p>Company Logo</p>	<p>Document Name : 소프트웨어 아키텍처 설계서</p>	<p>Page : 3/22</p>
----------------------------	--	------------------------

목 차

1. 개요.....	7
1.1. 범위	7
1.2. 목적	7
1.3. 시스템 개요.....	7
1.4. 항목 식별자.....	8
1.5. 관련 계획 및 표준.....	9
1.6. 가정	9
1.7. 제약사항	9
2. 분할 뷰.....	11
2.1. 요소, 관계, 속성	13
2.2. 사용한 모델링 언어	16
2.3. 상세 내용	16
3. 프로세스간 통신 뷰.....	17
3.1. 요소, 관계, 속성	18
3.2. 사용한 모델링 언어	18
3.3. 상세 내용	18
4. 구현 뷰.....	19
4.1. 요소, 관계, 속성	20
4.2. 사용한 모델링 언어	20
4.3. 상세 내용	20
5. 요구사항 추적 매트릭스.....	21

Company Logo	Document Name : 소프트웨어 아키텍처 설계서	Page : 4/22
---------------------	--	----------------

표 1 관련 계획 및 표준.....	9
---------------------	---

Company Logo	Document Name : 소프트웨어 아키텍처 설계서	Page : 5/22
---------------------	--	----------------

▪ 용어 및 약어

[용어] 용어 설명

[약어] 약어 설명

약어	원어	설명
EB	Event-Bus	Event-Bus 패턴에서 사용하는 Event-Bus를 말한다.

▪ References

[1] 저자. 인쇄 날짜. 제목. Volume Number. Edition. 인쇄사. 회사 이름. 문서 이름. Version Number. 날짜.

[2] <Committee Name>. <표준안 이름>. <Version Numb>. <날짜>

템플릿 작성 방향

소프트웨어의 아키텍처를 표현하는 방법으로 본 문서에서는 IEEE 1016-2009 Software Design Description의 용어와 개념을 사용한다.

Company Logo	Document Name : 소프트웨어 아키텍처 설계서	Page : 7/22
--------------	-----------------------------------	----------------

1. 개요

1.1. 범위

템플릿 작성 가이드

소프트웨어 아키텍처 설계서에는 소프트웨어 개발에 필요한 소프트웨어 아키텍처 설계 내용을 기록한다..

본 소프트웨어 아키텍처 설계서는 빌딩보안시스템 제어용 소프트웨어의 개발에 필요한 소프트웨어 아키텍처 설계를 기록한다. 소프트웨어 아키텍처 설계서는 빌딩보안시스템의 설계요소와 이벤트버스 패턴을 이용한 설계 내용을 상세화한다.

1.2. 목적

템플릿 작성 가이드

소프트웨어 아키텍처 설계서를 작성하고 이 문서를 참조하는 관련자들의 목적을 작성한다. 소프트웨어 아키텍처 설계서를 기준으로 소프트웨어 단위 설계, 개발, 검증을 수행하며 단위 개발된 소프트웨어의 통합 및 통합 검증을 수행한다. 관련자들은 소프트웨어 아키텍처 설계서를 사용해서 개발되는 소프트웨어의 구조, 성능, 통합 방법을 확인할 수 있다. 따라서 소프트웨어 아키텍처 설계서의 목적을 정리하면 다음과 같다.

- 소프트웨어 아키텍처의 평가 기준 제공
- 소프트웨어 통합 방안 파악
- 소프트웨어 통합 테스트 기준 제공
- 개별 모듈 별 개별 역할 분장
- 소프트웨어 변경 시 영향도 분석
- 소프트웨어 아키텍처에 대한 가이드라인 제공

이상의 목적을 해당 프로젝트에 맞도록 수정하여 작성하거나 신규 작성한다.

본 명세서의 목적은 빌딩의 보안을 위하여 향후 다양한 센서가 추가 될 수 있어 센서 추가 및 관리가 용이해야 하며 빌딩관리자가 쉽게 알람을 확인하고 처리 할수 있어야 한다.

- 빌딩보안시스템 소프트웨어 상세 설계를 위한 명세를 제공함
- 빌딩보안시스템 소프트웨어 검증을 위한 평가 기준을 제공함
- 빌딩보안시스템 교육 매뉴얼 작성 시 기초자료로 활용함

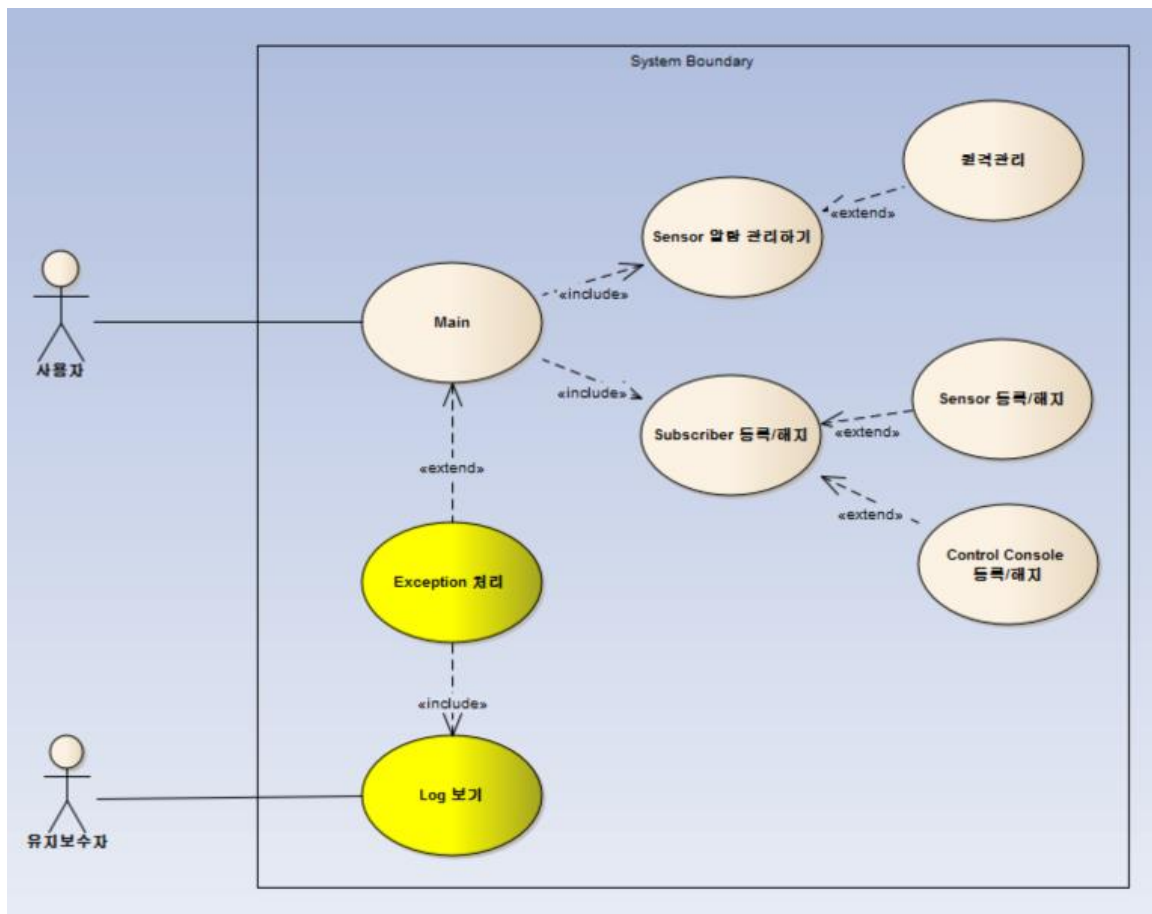
1.3. 시스템 개요

<div>Company Logo</div>	<div>Document Name :</div> <div>소프트웨어 아키텍처 설계서</div>	<div>Page :</div> <div>8/22</div>
-------------------------	--	-----------------------------------

템플릿 작성 가이드

제어용 소프트웨어와 이 소프트웨어가 사용되는 시스템의 목적에 대해서 간략하게 기술한다.

빌딩보안시스템은 빌딩의 보안을 위하여 다양한 센서를 관리하고 알람을 확인하고 처리하기 위한 시스템이다. 본 시스템은 Event-Bus 패턴 이용한 빌딩보안시스템으로 사용자 화면으로 알람을 쉽게 확인 할수 있어야 하고 사용자 화면에서 알람을 끄고 결수 있어야 한다. 유지보수자 입장에서는 장애의 원인을 쉽게 파악할 수 있어야 한다.



1.4. 항목 식별자

템플릿 작성 가이드

형상 관리 계획서에서 소프트웨어 아키텍처 항목의 식별을 위한 식별자 부여 방법을 설명한 경우, 여기서는 관리 계획서를 참조하라고만 언급한다. 형상 관리 계획서에서 별도 언급이 없는 경우 여기에서 정의한다.

형상 관리 계획서에서 소프트웨어 아키텍처 설계서의 항목 부여 방법을 참조한다.

1.5. 관련 계획 및 표준

템플릿 작성 가이드

소프트웨어 아키텍처 설계서와 관련된 계획과 표준을 언급한다.

본 계획서는 아래 계획서와 관련 표준을 참고 하고 있다.

구분	식별자	이름	참고 목적
계획서	BS-SW-PLAN-1	소프트웨어 개발 계획서	소프트웨어 개발에 대한 전반적인 계획을 수립한 것을 참조
	BS-SW-PLAN-3	소프트웨어 테스트 계획서	소프트웨어 테스트에 대해서 계획을 수립한 것을 참조
	BS-SW-PLAN-4	소프트웨어 통합 테스트 계획서	소프트웨어 통합 테스트에 대해서 계획을 수립한 것을 참조
	BS-EE-PLAN-1	요구사항관리 계획서	소프트웨어 요구사항 관리 방법 및 절차 참조
	BS-EE-PLAN-2	형상관리 계획서	소프트웨어 형상 관리 방법 및 절차 참조

표 1 관련 계획 및 표준

1.6. 가정

템플릿 작성 가이드

<개발하고자 하는 S/W 시스템과 관련된 가정 및 의존성을 기술한다.>

본 빌딩보안시스템은 Stand-alone 방식으로 되어 있지만 웹으로의 확장성을 고려한다.

1.7. 제약사항

<설계에 중요한 영향을 끼치는 제약 사항을 적는다. 다음 사항을 포함할 수 있다.>

- 하드웨어 및 소프트웨어 환경
- 사용자 환경

<p>Company Logo</p>	<p>Document Name : 소프트웨어 아키텍처 설계서</p>	<p>Page : 10/22</p>
---------------------	---	-------------------------

- 자원의 사용 가능성
- 준수할 표준
- 인터페이스 및 프로토콜
- 메모리 및 기타 성능 제한

본 빌딩보안시스템은 ARM 7 을 사용하는 16 비트 마이크로프로세스 기반위에서 작동한다.

모듈 뷰 타입 설명

모듈이란 잘 정의된 인터페이스와 구현을 갖춘 소프트웨어의 단위다. 모듈은 대체로 프로시저나 함수와 같은 서비스를 제공하고 모듈 내부의 자료 구조나 알고리즘의 일부나 전체를 숨긴다.

모듈 뷰를 살펴보면 소스코드를 독립적인 모듈로 어떻게 나누었는지, 모듈이 제공하는 서비스에 대해 각 모듈이 어떤 가정을 하는지, 이런 모듈을 어떻게 조합해서 더 큰 단위로 조합하는지 확인할 수 있다.

모듈화의 방법에 따라서 소프트웨어 아키텍처가 변경용이성, 이식성, 재사용성을 갖추었는지 확인할 수도 있다. 이런 모듈화 가운데 임베디드 소프트웨어와 관련이 있는 뷰는 분할 뷰, 사용 뷰, 계층 뷰로 나눌 수 있다.

- 분할 뷰: 모듈 간의 포함관계에 초점을 맞출 때 사용함
- 사용 뷰: 모듈 간의 기능적 의존관계를 밝힐 때 사용함
- 계층 뷰: 모듈 사이의 관계를 사용허가(allowed-to-use) 관계로 제약해 놓은 형태로 나타낼 때 사용함

모듈 뷰 타입에서 요소, 관계 속성은 다음과 같이 정의한다.

- 요소: 모듈 뷰타입에서 요소는 모듈임. 모듈은 소프트웨어 구현의 단위로서 응집된 단위 기능을 제공함
- 관계: 모듈에서는 is-part-of, depends-on, is-a 형태의 관계가 있음
 - A is-part-of B: A 모듈은 B 모듈의 부분이나 자식에 해당하는 관계를 나타냄
 - A depends-on B: A 가 B 에 의존하는 관계를 나타냄
 - A is-a B: A 가 B 에서 파생된 것을 나타냄
- 요소의 속성: 모듈의 속성에는 이름, 책임, 구현 정보가 있음
- 관계의 속성: is-part-of 의 경우 가시성, depends-on 의 경우 의존성, is-a 의 경우 구현 속성이 있음

<p>Company Logo</p>	<p>Document Name : 소프트웨어 아키텍처 설계서</p>	<p>Page : 11/22</p>
---------------------	---	-------------------------

2. 분할 뷰

템플릿 작성 가이드

시스템의 기능이 어떤 식으로 모듈 단위로 나뉘고 모듈이 어떤 식으로 하위 모듈로 나뉘는지를 나타낸다. 대부분의 아키텍처는 모듈 분할 스타일로 시작하며 아키텍트가 분할 정보(divide and conquer) 방식으로 문제를 공략한다. 분할 정복으로 나온 결과를 분할 뷰로써 기록한다.

모듈을 하위의 모듈로 분할하는 기준은 다음과 같다.

- 품질 목표 달성: 아키텍처가 변경 용이성이라는 품질을 달성해야 한다면 변경이 잦을 것이라고 예상하는 것을 하나의 모듈로 만든다. 성능 측면에서 수행시간이 빨라야 하는 것과 수행이 느려도 되는 기능을 나누어서 모듈로 만들면 스케줄 정책으로 달리하는 것으로써 품질 목표를 달성할 수 있다.
- 구매 혹은 구축 결정(Make or buy decision): 여러 가지 이유로 특정 기능은 상용 제품을 사용해야 하는 경우 구매할 기능과 프로젝트를 수행해서 구축할 기능을 분할한다.
- 프로덕트 라인 구축: 제품군을 효과적으로 구현하려면 공용부와 가변부를 구분하는 게 필요하다.

분할 뷰에서 요소, 관계 속성은 다음과 같이 정의한다.

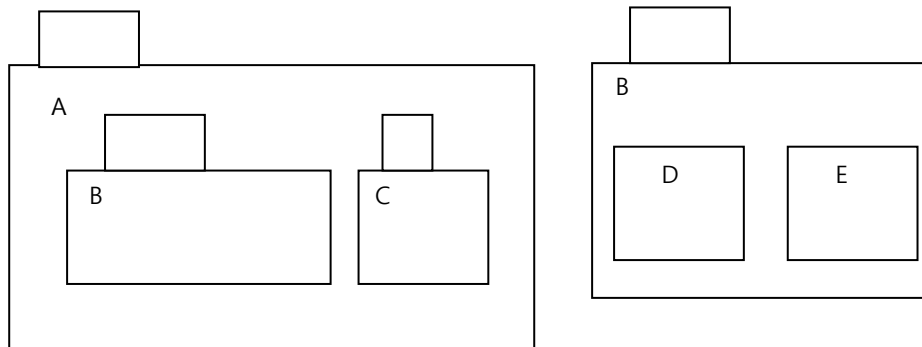
- 요소: 모듈- 모듈 뷰타입에서 정의한 요소다. 다른 모듈들을 묶어 놓은 모듈인 하위 시스템도 있다.
- 관계: 분할 관계- 'is-part-of' 관계를 정제해 놓은 형태이다. 분할을 결정할 때 사용한 기준은 반드시 기록해야 한다.
- 요소의 속성: 모듈 뷰타입에서 정의한 속성을 그대로 사용한다.
- 관계의 속성: 가시성- 모듈이 존재하는 범위와 가용한 모듈의 기능을 부모 모듈 밖에서 확인할 수 있는지를 나타낸다.
- 구성 형태: 분할을 나타낸 다이어그램에서 재귀 표현을 허용하지 않는다. 하나의 뷰 내에서 한 모듈의 부모는 둘 이상일 수 없다.

모듈 뷰의 사용처는 다음과 같다.

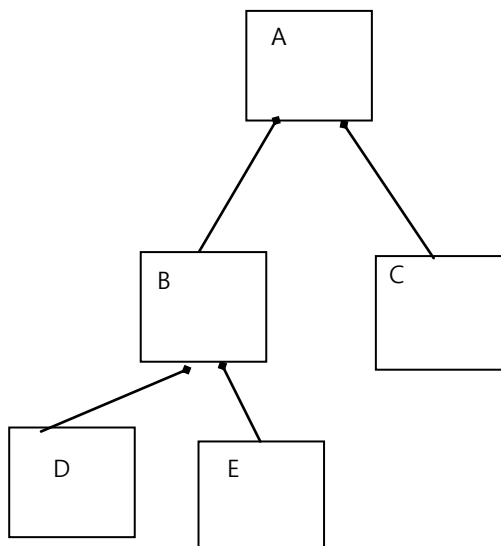
- 임베디드 제어용 소프트웨어 시스템을 학습 시
- 형상 관리의 형상 항목을 도출할 때
- 임베디드 소프트웨어 개발을 위한 작업할당의 기초자료
- 변경의 영향도를 분석할 때

모듈 뷰를 표현하는 방법은 다양하다. 우선 일반적인 표기 방법으로 박스를 사용해서 모듈 및 모듈의 포함관계를 표현하는 방법이 있다. UML 을 사용하는 방법으로는 패키지 다이어그램과

복합 관계를 사용해서 나타내는 방법도 있다.

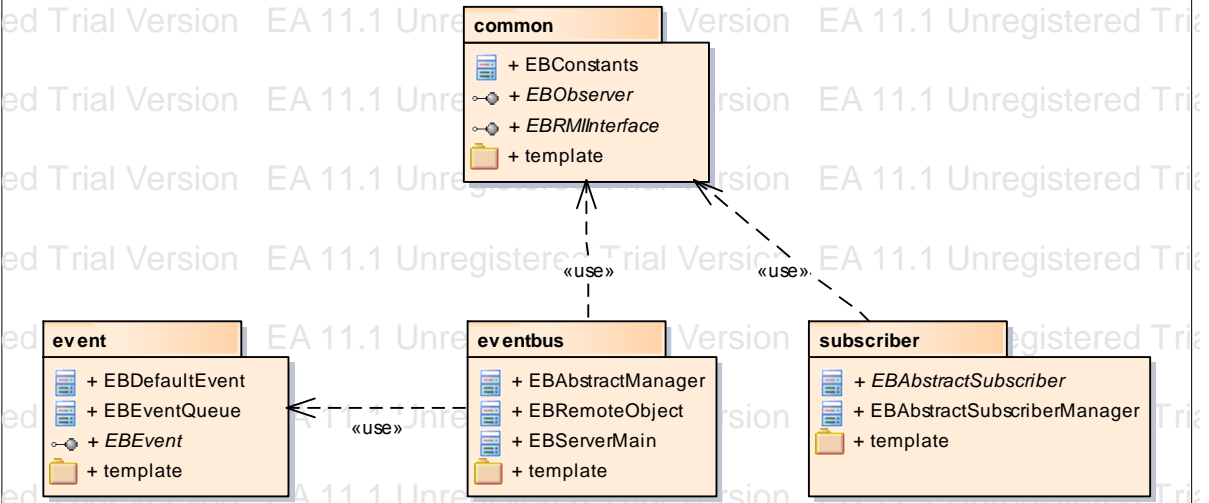


패키지 다이어그램을 사용해서 모듈 뷰 나타내기



복합 관계를 사용해서 모듈 뷰 나타내기

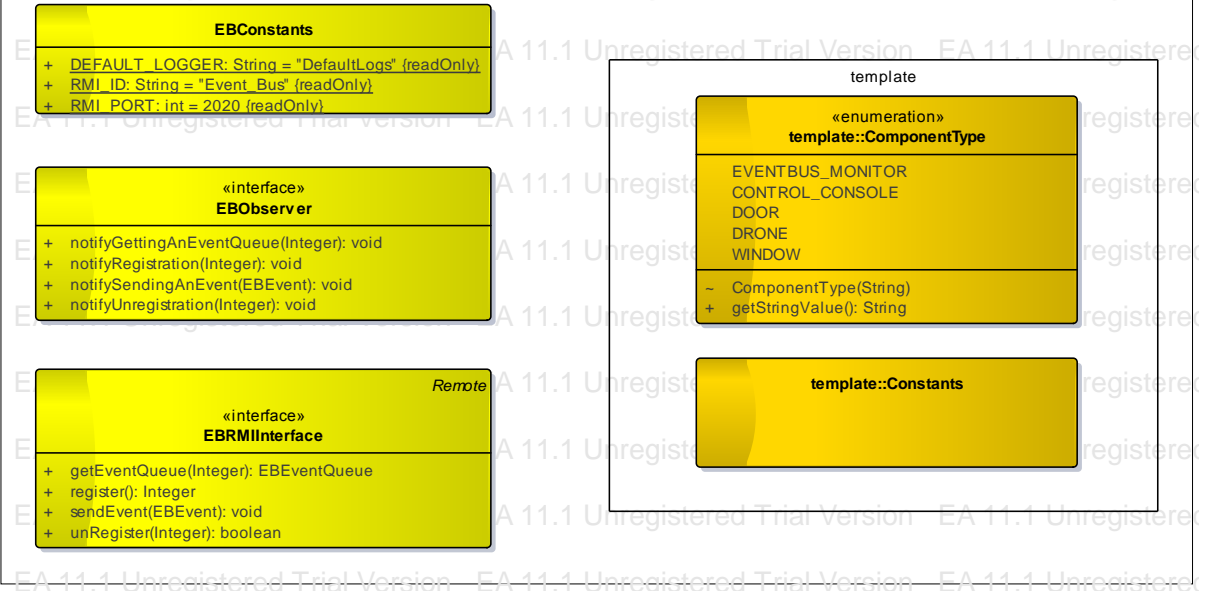
class ModuleView



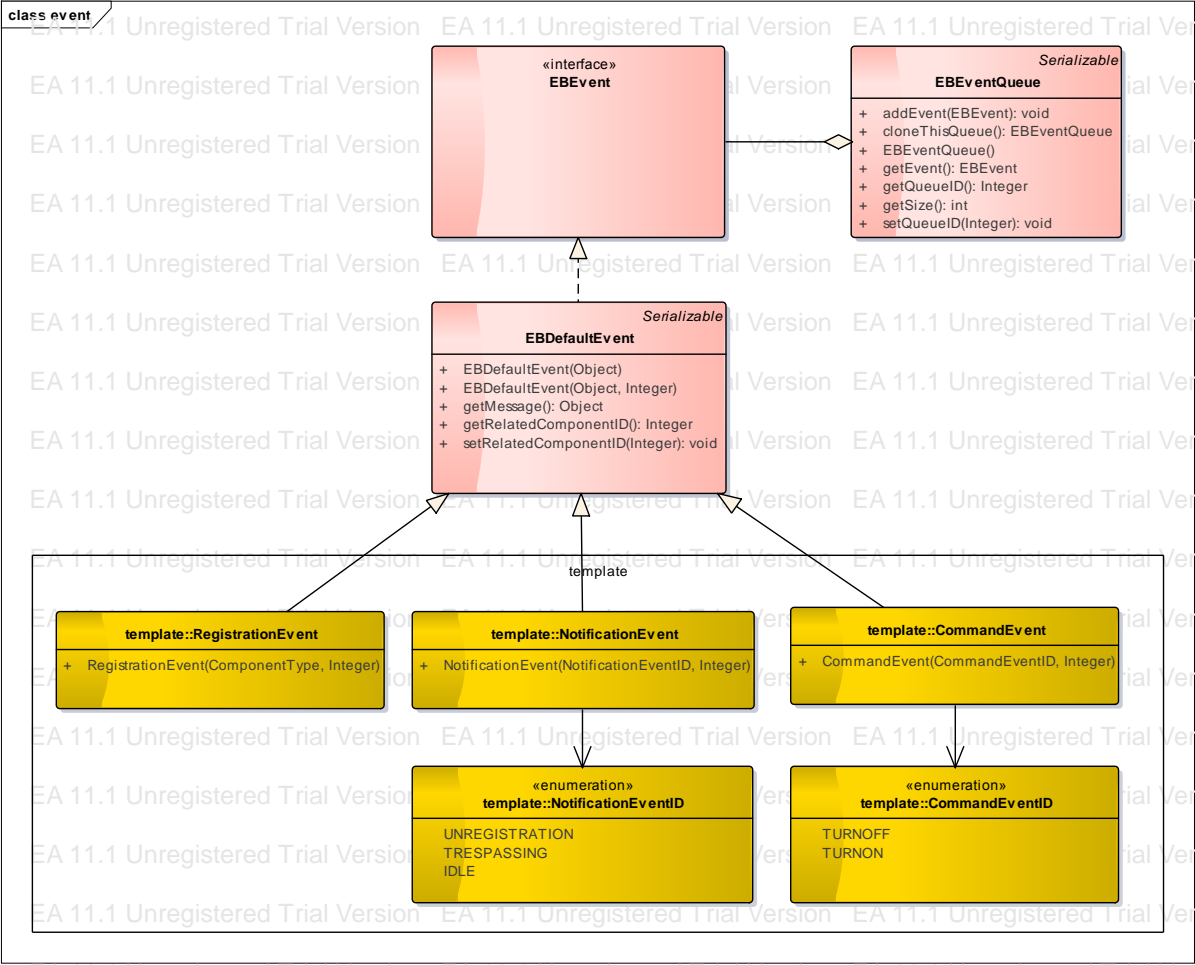
2.1. 요소, 관계, 속성

Common

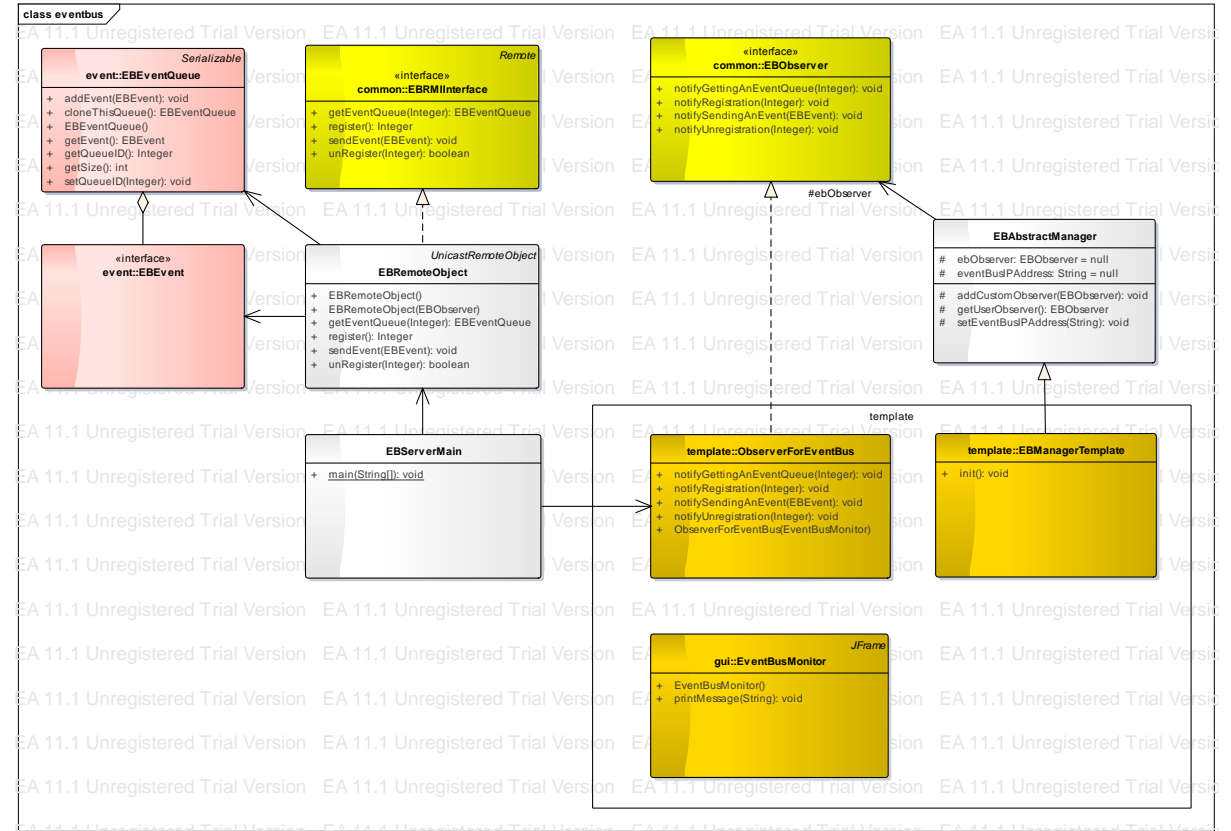
class common



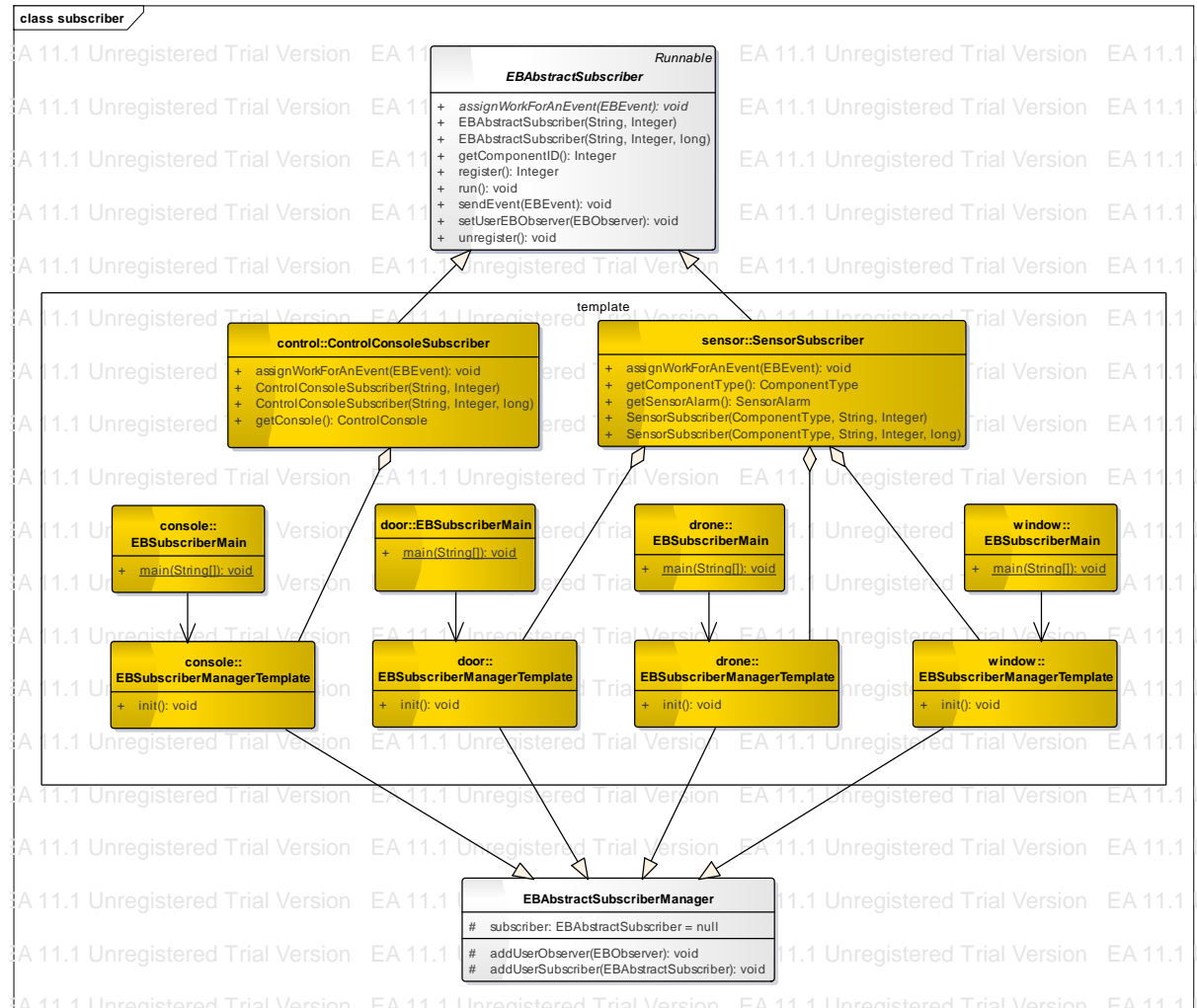
Event



Eventbus



Subscriber



2.2. 사용한 모델링 언어

본 시스템의 아키텍처 설계는 UML 2.1 notation 을 따른다.

2.3. 상세 내용

모듈뷰는 크게 Common, event, eventbus, subscriber 의 4 개의 패키지로 구성이 되어 있다.eventbus 와 subscriber 패키지는 common 패키지의 메소드를 상속한다. Event 패키지는 eventbus 패키지의 메소드를 상속 받아서 사용한다.

<p>Company Logo</p>	<p>Document Name : 소프트웨어 아키텍처 설계서</p>	<p>Page : 17/22</p>
---------------------	---	-------------------------

3. 프로세스간 통신 뷰

템플릿 작성 가이드

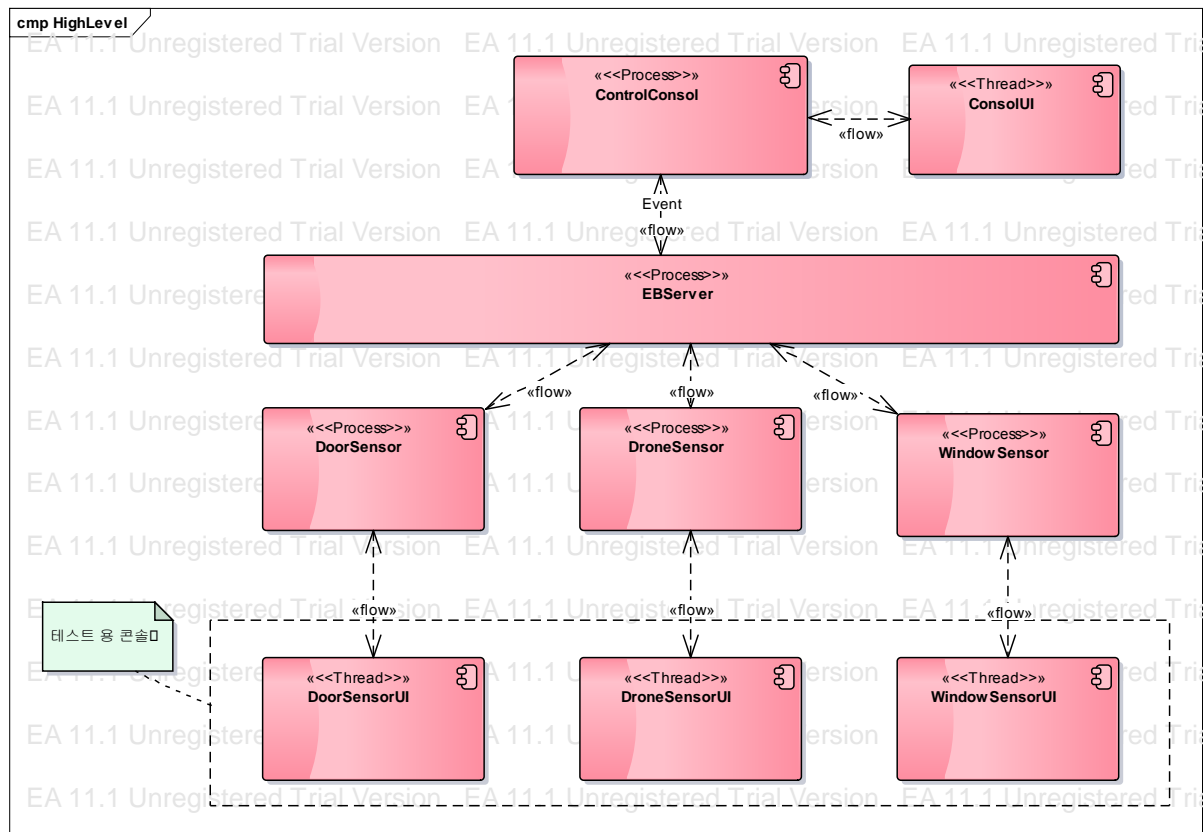
프로세스간 통신 뷰의 특징은 동시에 수행되는 컴포넌트들끼리 다양한 커넥터 방식으로 상호작용하는 데 있다. 여기에 사용되는 커넥터 방식으로는 동기화, 메시지 전송, 데이터 교환, 시작, 중단 등이 있다. 동시성과 관련된 행위를 이해하는 데 프로세스간 통신 뷰를 사용하면 좋다.

프로세스간 통신 뷰에서 요소, 관계 속성은 다음과 같이 정의한다.

- 요소: 컴포넌트 타입(태스크, 프로세스, 스레드), 커넥터 타입(데이터 교환, 메시지 전송, 동기화, 제어, 기타 통신 유형)
- 관계: C&C 뷰 타입에서 정의한 것과 같은 종류의 붙임
- 연산 모델: 동시에 수행되는 컴포넌트들이 특정 커넥터 방식을 통해 상호작용함
- 속성: 동시수행 단위(선점가능성, 우선순위, 시간제한 인자), 데이터 교환(즉시 처리될 수 없는 메시지는 저장해 둬를 의미함)
- 구성 형태: 임의의 그래프

프로세스간 통신 뷰의 사용처는 다음과 같다.

- 시스템에서 병렬적으로 작동하는 부분 확인
- 컴포넌트를 포함할 프로세스 확인
- 제어 흐름 파악



3.1. 요소, 관계, 속성

가장 상위에 콘트롤 콘솔 프로세스가 있고 콘솔 UI와 연결되어있다.. 콘트롤 콘솔 프로세스는 EBServer 프로세스를 제어한다. EBServer는 door, drone, window 프로세스를 관리한다. Door Sensor, drone sensor, window sensor는 각각의 UI 프로세스를 관리한다.

3.2. 사용한 모델링 언어

본 시스템의 아키텍처 설계는 UML 2.1 notation 을 따른다.

3.3. 상세 내용

가장 상위에 콘트롤 콘솔 프로세스가 있고 콘솔 UI에 정보를 정보 받는다. 콘솔 UI는 스크레드로 동작한다. 콘트롤콘솔 프로세스는 EBServer와 주기적으로 통신을 한다. EBServer는 door, drone, window의 상태를 모니터링한다. Door Sensor, drone sensor, window sensor는 각각 UI를 가지고 해당 UI는 스크레드로 동작한다.

<p>Company Logo</p>	<p>Document Name : 소프트웨어 아키텍처 설계서</p>	<p>Page : 19/22</p>
---------------------	---	-------------------------

할당 뷰 타입

모듈 뷰나 커넥터 뷰에 나오는 요소를 개발 환경이나 형상 관리 환경에 대응할 때 할당 뷰 타입을 사용한다. 할당 뷰는 다음과 같다.

- 구현 뷰: 모듈과 파일 시스템이 어떻게 대응하는지 설명함
- 작업할당 뷰: 모듈을 개발할 조직에 어떻게 대응하는지 설명함

4. 구현 뷰

템플릿 작성 가이드

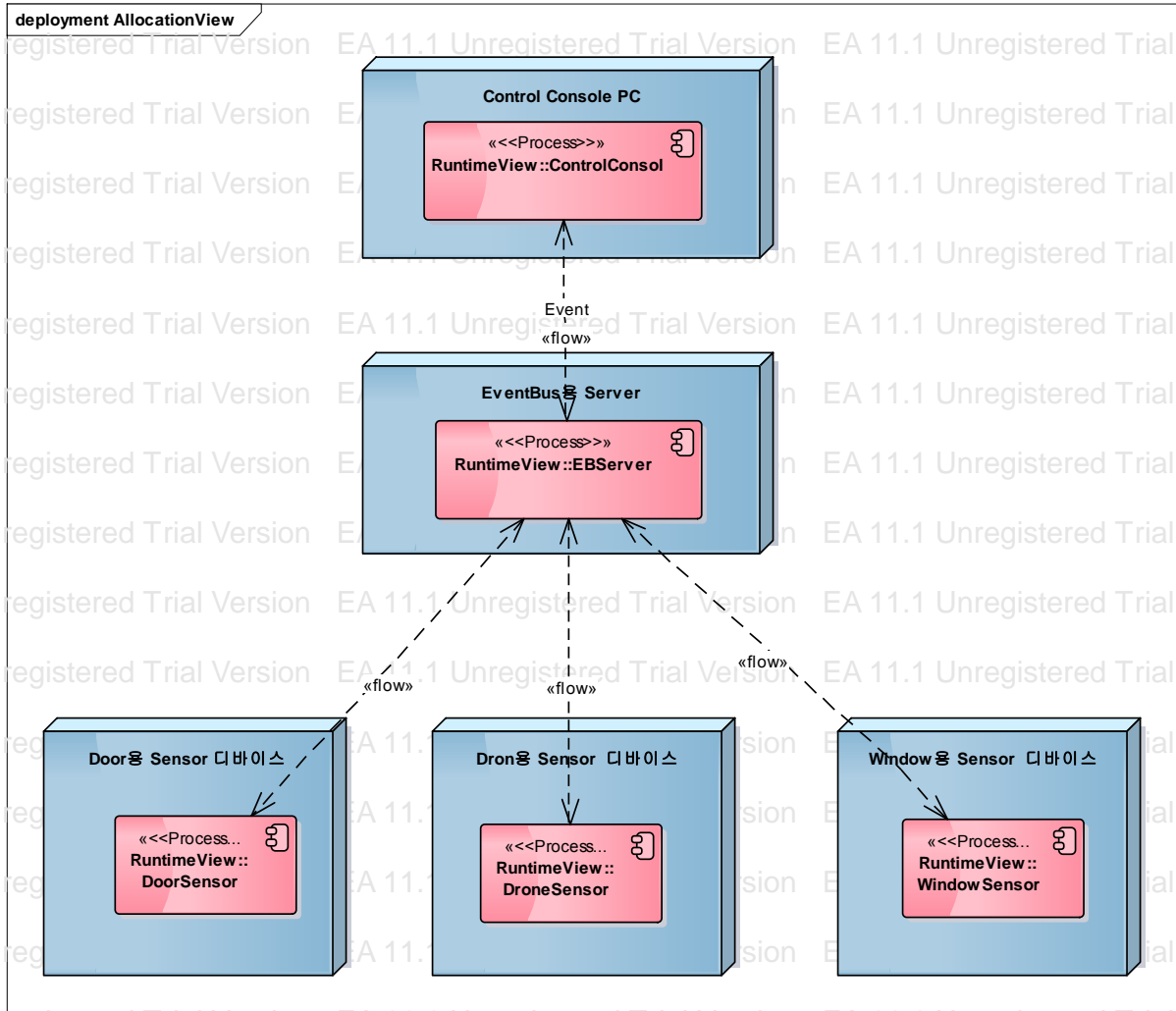
모듈 뷰 타입에 나오는 모듈들을 개발 인프라에 대응한다. 모듈을 구현할 때 소스코드 파일, 헤더 파일, 빌드 파일을 만든다. 즉 이런 파일을 형상 관리 시스템으로 관리하기 위한 뷰이다.

구현 뷰에서 요소, 관계 속성은 다음과 같이 정의한다.

- 요소: 소프트웨어 요소(모듈), 환경 요소(파일, 디렉터리)
- 관계: 포함(다른 형상 항목을 포함하는 형상 항목을 보여 줌), 할당(모듈이 형상 항목에 할당됨)
- 구성 형태: 형상 항목이 포함관계를 형성한다.

구현 뷰의 사용처는 다음과 같다.

- 소프트웨어 요소와 연동하는 파일을 유지하고 관리하는 데 사용함



4.1. 요소, 관계, 속성

물리적인 형상항목을 보여준다. Contorl PC, 서버, Door Sensor, Dron Sensor, Window용 센서 디바이스로 구성되어 있다. Control Consol PC는 서버의 이벤트를 제어한다. Event bus서버는 3개의 센서를 모니터링하면서 제어한다.

4.2. 사용한 모델링 언어

본 시스템의 아키텍처 설계는 UML 2.1 notation 을 따른다.

4.3. 상세 내용

Control Console PC 는 가장 상위에 위치한 제어기로서 Control Console 프로세스를 가지고 있다. Control Console PC 는 서버에 이벤트를 보내서 빌딩보안관련 제어 및 모니터링을 한다. 서버는 EventBus 로 구현이 되어 있고 각 센서 디바이스를 관리한다. 센서 디바이스는 Door, Drone, Window 용의 3 개로 구성되어 있고 각각 운영 프로세스를 가지고 있다. 또한 UI 도 제공을 하며 쓰레드로 운영한다.

5. 요구사항 추적 매트릭스

템플릿 작성 가이드

소프트웨어 요구사항과 소프트웨어 아키텍처 요소 사이의 추적성을 기록한다.

요구사항 식별자	요구사항	소프트웨어 아키텍처 식별자	요소
REQ-FR-1	사용자가 제어콘솔 화면에서 등록 버튼을 누른다.	Module View	
REQ-FR-2	등록버튼을 누르고 해지의 경우는 사용자가 컨트롤 콘솔의 해지 버튼을 누른다	Module View	
REQ-FR-3	콘솔 화면을 통해 컨트롤 콘솔이 등록된 것을 확인한다.	Module View	
REQ-FR-4	콘솔 화면이 등록이 되지 않으면 시스템 로그 파일에 저장한다	Module View	
REQ-FR-5	사용자는 subscriber화면(Door, drone, window)에서 등록 버튼을 누른다	Module View	
REQ-FR-6	해지의 경우 Subscriber 해지 버튼을 누른다. 콘솔화면을 통해 Subscriber가 해지 된 것을 확인한다.	Module View	
REQ-FR-7	사용자가 테스트를 위	런타임뷰	

<p>Company Logo</p>	<p>Document Name : 소프트웨어 아키텍처 설계서</p>	<p>Page : 22/22</p>
----------------------------	--	-------------------------

	<p>한 Sensor화면에서 알 람 발생 버튼을 누른 다</p>		
<p>REQ-FR-8</p>	<p>콘솔 화면을 통해 센 서(Door, drone, window)을 확인한다.</p>	<p>런타임부</p>	
<p>REQ-FR-9</p>	<p>알람을 해제 할때는 사용자가 control console화면의 각 sensor 버튼을 누른다.</p>	<p>구현부</p>	
<p>REQ-FR-10</p>	<p>콘솔 화면을 통해 subscriber 가 해지된 것을 확인한다.</p>	<p>구현부</p>	
<p>REQ-FR-11</p>	<p>등록이 실패하면 알람 이 발생하지 않는다.</p>	<p>구현부</p>	
<p>REQ-FR-12</p>	<p>시스템 로그파일을 통 해 실패원인을 확인한 다.</p>	<p>구현부</p>	