

# M2M/IoT 기반의 사무실 관리 서비스 플랫폼 연구

남강현\*

A Study on the Office Management Service Platform based on M2M/IoT

Kang-Hyun Nam\*

요 약

사무실 관리 서비스 플랫폼은 사무실 센서 장치, G/DSCL(Gateway/Device Service Capability Layer), NSCL(Network Service Capability Layer), NA(Network Application)로 구성된다. 본 논문에서는 사무실 관리 서비스에 적합한 게이트웨이 리소스트리와 서비스 시나리오를 설계하고, 지능형 기능 모델링을 통해서 사무실 관리 서비스가 운영 될 수 있음을 보인다.

ABSTRACT

The office management service platform configured with office's sensor devices, G/DSCL(Gateway/Device Service Capability Layer), NSCL(Network Service Capability Layer), and NA(Network Application). In this paper, we designed gateway resource tree and service scenario to fit the office management service and demonstrated appropriate operation of the office management service through intelligent functional modeling.

키워드

M2M(Machine to Machine) or IoT(Internet of Thing), Device, Gateway, Resource Tree  
사물지능 통신, 디바이스, 게이트웨이, 리소스트리

## 1. 서 론

사물 지능 통신은 사람이 통신 과정에 개입하지 않고 통신이 이루어지는 다양한 통신을 지칭하며 ETSI(European Telecommunications Standards Institute)에서 M2M(Machine to Machine)을 기반으로 스마트 미터링, 헬스케어, 스마트 홈 등에 Use Case, 아키텍처, 요구사항 기술 표준을 제정하였고, oneM2M에서 지능형 검침(Smart Meter), 전자 보건(e-Health), 통신 가전(Connected Consumer), 도시 자동화(City Automation), 차량 응용(Automotive Application) 등을 포함하는 다양한 분

야에 TS(Technical Specifications) 및 TR(Technical Reports)를 준비하고 있다[1-4].

본 논문에서는 사무실에 설치된 센서들의 데이터를 수집처리 하는 게이트웨이 기능과 사무실 센서 서비스에 특화된 리소스트리 구조를 설계 하고, 지능화된 서비스를 처리하는 사무실 관리 어플리케이션 서비스를 구현한다.

본 논문은 2장에서는 사무실 관리 서비스 플랫폼 구조와 기능을, 3장은 사무실 센서 데이터 수집 프로세스 및 관리서버에 정보전달을, 4장은 지능화 모델링을 통한 사무실 관리 서비스를, 마지막 5장에서 결론 및 향후 연구로 끝을 맺는다.

\* 교신저자(corresponding author) : 광주대학교 정보통신학과(khnam@gwangju.ac.kr)  
접수일자 : 2014. 10. 20

심사(수정)일자 : 2014. 11. 21

게재확정일자 : 2014. 12. 15

## II. 사무실 관리 서비스 플랫폼 구조와 기능

ETSI에서 정의하고 있는 SCL(Service Capability Layer)은 NSCL(Network SCL), DSCL(Device SCL), GSCL(Gateway SCL)로 존재하며, 단말 인증, 원격제어, 데이터 수집, 서비스 관리 등 M2M 네트워크 및 서비스 관리를 위한 공통 기능을 제공하는 역할을 한다. 그림 1과 같이 사무실 관리서비스 제공자는 서버에서 REST 방식의 인터페이스를 통해서 NSCL과 G/DSCL의 리소스트리 저장 및 기능 제어처리 한다.

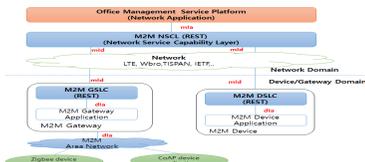


그림 1. 사무실 관리 서비스 플랫폼 구조  
Fig. 1 The office management service platform structure

그림 2와 같이 사무실 관리 서비스의 서비스 구축을 위하여 SCL 과 Application을 연결하는 N/GAE(Network/Gateway Application Enablement), 서로 다른 SCL을 연결하는 N/GGC(Network/Gateway Generic Communication), 자원의 등록 및 정보를 관리하는 N/GRAR(Network/Gateway Reachability, Addressing and Repository) 그리고 네트워크 등의 설정을 관리하는 N/GREM(Network/Gateway Remote Entity Management)들로 구성되어, 센서(Sensor)들의 정보들과 사무실 관리 서비스 플랫폼 사이에 기능동작이 처리될 수 있도록 한다.

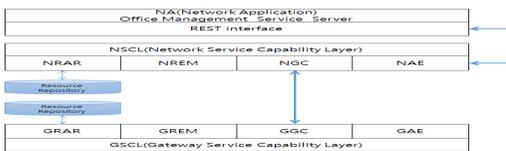


그림 2. 사무실 관리 서비스 플랫폼 기능  
Fig. 2 The Office management service platform features

Network 기능들에 있어서, NRAR기능은 센서(Sensor)나 게이트웨이의 IP 주소를 가질 수 있게 하고,

센서 등록과 관련된 내용을 저장하며, 이벤트와 관련된 가입과 통지를 수행하고, 센서나 게이트웨이를 생성 및 삭제할 수 있게 한다. NREM기능은 게이트웨이 구성관리를 하며, 장애 또는 성능 관련 수집 활성화 시키는 것과 그에 따른 데이터 수집과 저장을 할 수 있고, 게이트웨이의 소프트웨어와 펌웨어(Firmware) 업그레이드를 수행한다. NGC기능은 게이트웨이와 통신을 하며, 데이터를 교환 기능을 제공한다. NAE기능은 네트워크 및 애플리케이션 도메인에 애플리케이션의 단일 접속점이며, NSCL에서 구현된 기능들을 공개하고 사무실관리 서비스 서버가 NSCL에 등록할 수 있도록 한다.

게이트웨이 기능들에 있어서, GRAR기능은 센서들의 이름과 라우팅 가능한 네트워크 주소, 상태, 스케줄링 정보를 관리하며, 이벤트와 관련된 가입(subscription)과 통지(notification), 그리고 센서의 그룹을 생성, 삭제하고 조회할 수 있으며, 등록정보를 NSCL을 통하여 사무실관리 서버가 정보를 저장할 수 있다. GREM기능은 원격 관리 클라이언트로 동작하며, 사무실관리 서버로부터 NREM을 통하여 하나 또는 여러 게이트웨이를 목적지로 하는 관리 요청을 받아들이고 처리한다. 만일 사무실관리 서버에서 장애 감지 트리거에 의해 장애 감지가 되는 경우, 센서로부터 처리될 수 있는 이벤트 정보를 NREM을 통하여 사무실 관리 서버에 보고한다. GGC기능은 NSCL과 통신하기 위한 단일 접속점이며, REST 프로토콜을 통하여 NSCL과 통신 한다. GAE기능은 게이트웨이 애플리케이션과 센서 애플리케이션에 대한 단일 접속점이고, 센서 디바이스에서 코디네이터(Coordinator)로 전달된 정보를 게이트웨이의 SPF(Sensor Proxy Function)핸들러를 통하여 게이트웨이 애플리케이션 기능들에 접근할 수 있도록 한다. 그림 3은 디바이스와 게이트웨이 그리고 사무실 관리 서버에서 제공하는 기능을 모델링 한 것이다.

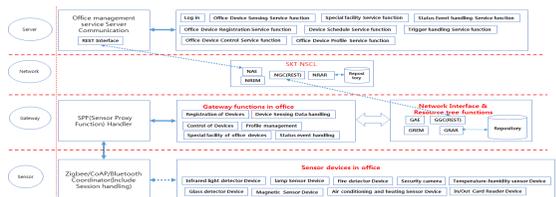


그림 3. 게이트웨이/디바이스를 이용한 사무실관리 서비스 플랫폼의 Use-case 다이어그램  
Fig. 3 Use-case diagram for office management service platform using gateway/device

### III. 사무실 센서 데이터 수집 프로세스 및 관리서버에 정보전달

사무실에 구성된 장비들은 냉난방기, 전등, 출입개폐 자석감지기, 유리감지기, 화재감지기, 출입 카드 리더기, 적외선감지기, 보안카메라, 온습도 측정 장비 등 이 있고, 게이트웨이의 기능구조는 그림 4에서 처럼 센서의 정보를 코디네이터가 처리하여, SCL 기능 블록(SCL Function Blocks)에서 센서 등록 또는 삭제와 같은 기능들을 처리한다. 각각의 기능설명은 표 1과 같다.

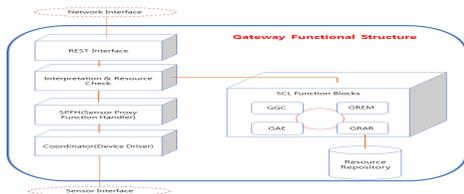


그림 4. 게이트웨이 기능 구조  
Fig. 4 Gateway functional architecture

표 1. 게이트웨이 기능에 대한 설명  
Table 1. Description of gateway function

REST Interface	Interface between Network and Gateway using REST Protocol.
Interpretation & Resource Check	Retrieve information receiving from Network. Send information to SCL Function Blocks. Retrieve information receiving from SPFH. Checking SCL discovery, registration, Applications registration, access-rights management, Group management, resource discovery, subscription management, announce/de-announce management
Sensor Proxy Function Handler	Retrieve information receiving from Coordinator. Handling Session messages from Set-up to Close.
Resource Repository	Repository of Resource tree
SCL Function Blocks	Handle of Sensing data Profile management Handle of status event Special Service function Device Control
Coordinator	Support interface between Sensor and Gateway

#### 3.1 사무실 센서 데이터 수집 프로세스

센서 초기화 상태에서 사무실 관리 서비스 서버로부터 사무실의 구성 데이터를 GSCL 기능 블록을 통해 자원 저장소(Resource Repository)에 저장되며, 코디네이터는 사무실 센서(Sensor) 들의 상태 점검을 하고, 등록 처리 한다. 등록 처리 되면, 센서 데이터 송수신 처리 그리고 세션 처리, 운영 및 관리 처리를 위한 초기화가 완성된다. 그림 5는 코디네이터가 처리 하는 클래스 다이어그램이다.

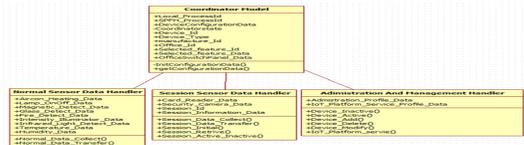


그림 5. 코디네이터 클래스 다이어그램  
Fig. 5 Class diagram of coordinator

#### 3.2 사무실 관리서버로 정보전달

ETSI TS 102.690에서는 M2M 서비스관리 목적으로 SCL 리소스트리(Resource Tree)를 정의하고, ETSI TS 102 921은 리소스에 대해 SCL 프리미티브(Create, Retrieve, Update, Delete로 구성)를 정의하고 있으며, 상호간 연동을 위해서 mId, mIa, dIa 참조점을 정의 했다. 사무실 관리서버와 센서들 사이에 정보교환을 위해서 게이트웨이 리소스트리가 설계 되었고(그림 6), 그에 따른 사무실 관리서버와의 연동 서비스 시나리오 절차가 제시 되었다(그림 7).

##### 3.2.1 GSCL 리소스트리 설계

그림 6에 설계된 리소스 트리는 세부분으로 구분된다. 센서로부터 수신되어 처리되는 코디네이터 부분, 네트워킹을 위해 센서를 구분 지어 처리 하는 부분, 센서 디바이스 어플리케이션을 처리해야하는 디바이스 등록/센싱/제어, 상태 이벤트, 사무실 특수 기능 서비스 부분으로 구성되어 설계되었다[5-6].

##### 3.2.1.1 코디네이터 설계

코디네이터에서 센서 데이터 처리와 세션 데이터 처리 그리고 운영 및 관리 처리 부분들은 ETSI M2M <application> 리소스로 모델링되고, 이 리소스의 "searchString" 애트리뷰트가 ETSI.ObjectType/ ETSI.COORDINATOR tag형태로 코디네이터에서 사용하는 기능들을 인식하게

하며, URI가 아래의 포맷으로 사용한다.

```
<scBase>/applications/<Interacting_application>
<application> 리소스는 서브리소스로 <container>
를 포함하고, 이 서브리소스의 "searchString" 애트리뷰트
가 ETSI.ObjectSemantic 부류의 태그를 이용 객체들
을 대체하는 규약으로 표시할 수 있으며, <container>
리소스에 접근 하려는 URI는 아래의 포맷을 가진다.
```

```
<scBase>/applications/<interacting_application>/c
ontainers/descriptor
```

<container> 리소스는 여러개의 <contentInstance> 서브리소스를 포함할 수 있고, 이 서브리소스의 "content" 애트리뷰트는 코디네이터를 이용한 처리 내용을 포함할 수 있다.

코디네이터는 <networking\_sensor\_devices>에 접근 할 수 있고, <contentInstance> 리소스의 "content" 애트리뷰트가 <networking\_sensor\_devices>의 URI를 가지고 되고, 현재의 코디네이터의 정보들이 <contentInstance> 리소스를 통해서 <networking\_sensor\_devices>로 정보 처리할 수 있다. URI는 <contentInstance> 리소스 접근을 위해서 사용되고 아래의 포맷을 가진다.

```
<scBase>/applications/<networking_sensor_devices>/
containers/descriptor/contentInstances/latest
```

### 3.2.1.2 센서 디바이스 네트워킹 설계

각기 코디네이터의 기능들은 <networking\_sensor\_devices>에 속하게 되고, ETSI M2M <application> 리소스로 모델링된다. 이 리소스의 "searchString" 애트리뷰트가 ETSI.ObjectType/ETSI.SENSOR\_DEV tag 형태로 센서(Sensor) 디바이스에 코디네이터기능이 인식하게 하며, URI가 아래의 포맷으로 사용한다.

```
<scBase>/applications/<networking_sensor_devices>
<application> 리소스는 서브리소스로 <container>
를 포함하고, 이 서브리소스의 "searchString" 애트리뷰트
가 ETSI.ObjectSemantic 부류의 태그를 이용 객체들
을 대체하는 규약으로 표시할 수 있으며, <container>
리소스에 접근 하려는 URI는 아래의 포맷을 가진다.
```

```
<scBase>/applications/<networking_sensor_devic
es>/containers/descriptor
```

<container> 리소스는 여러개의 <contentInstance> 서브리소스를 포함할 수 있고, 이 서브리소스의 "content" 애트리뷰트는 센서 디바이스를 이용한 처리 내

용물을 포함할 수 있다. 센서 디바이스는 <networking\_sensor\_device\_application>에 접근 할 수 있고, <networking\_sensor\_devices> 리소스의 "content" 애트리뷰트가 <networking\_sensor\_device\_application>의 URI를 가지고 되고, 현재의 코디네이터의 정보들이 <contentInstance> 리소스를 통해서 <networking\_sensor\_devices>로 정보 처리할 수 있다. URI는 <contentInstance> 리소스 접근을 위해서 사용되고 아래의 포맷을 가진다.

```
<scBase>/applications/<networking_sensor_
device_application>/
containers/descriptor/contentInstances/latest
```

### 3.2.1.3 센서 디바이스 어플리케이션 설계

각기 센서 디바이스의해 적용된 인터페이스에 포함되어서 처리되는 어플리케이션들은 미러링과 리타캐팅이 필요하다. 미러링은 어플리케이션 각기 기능의 동기를 유지하는 자원 구조의 표현이고, 리타캐팅은 자원구조에서 데이터의 저장 없이 디바이스에서 직접 데이터를 패치하는 방법을 정의한 것이다. 미러링을 지원하기위해서 어플리케이션이 모델링된 <application> 리소스는 미러 되는 각기 인터페이스 요소를 위해서 <container> 서브 리소스를 포함한다. 이러한 <container> 리소스를 접속하기 위해서 사용되는 URI는 아래와 같다.

```
<scBase>/applications/<networking_sensor_devic
e_application>/containers/<deviceRegistration_Admin
istration>
```

```
<scBase>/applications/<networking_sensor_devic
e_application>/containers/<ServiceStateHandle>
```

<container> 리소스는 다수의 <contentInstance> 서브리소스를 포함할 수 있고, 이 서브리소스의 "content" 애트리뷰트는 응용 서비스 사용을 위해서 활용되고, <contentInstance> 리소스를 접근하기 위해 사용되는 URI는 아래와 같다.

```
<scBase>/applications/<networking_sensor_devic
e_application>/containers/<deviceRegistration_Admin
istration>/contentInstances/latest
```

```
<scBase>/applications/<networking_sensor_devic
e_application>/containers/<ServiceStateHandle>/cont
entInstances/latest
```



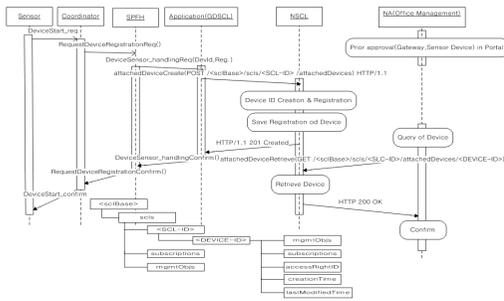


그림 7. 센서 디바이스 등록 시나리오 및 리소스 트리  
Fig. 7 Registration scenario of sensor device and resource tree

3.2.2.2 센서 수집 Data 전달 시나리오

센서 데이터들은 주기적으로 SensorData\_req로 해당 Sensor ID와 Data를 보내고, 코디네이터는 해당 센서 디바이스를 인식 할 수 있도록 SPFH로 프로시저콜 (SensorDeviceAttach)처리를 통하여 해당 센서 를 찾고, M2M 규격의 데이터 형태로 저장 처리될 수 있도록 Syntax check를 하고, 맵핑된 어플리케이션으로 DeviceSensor\_handlingReq를 처리한다. 해당 어플리케이션은 센서 데이터를 자신의 리소스에 저장하고, NSCL의 리소스트리에 저장하는 절차를 수행하기 위해서 contentInstanceCreate 프리미티브를 NSCL로 전달한다.

HTTP Request 헤더에는 Auth\_id와 Auth Key와 content가 전달되며, 이후 NSCL은 그림 8에 제시된 리소스 형태로 해당 센서(Sensor) 디바이스에 데이터를 저장한 후, 성공적으로 저장되었음을 알린다.

POST /OMS/scls/SC01/containers/contCollection/contentInstances HTTP/1.1

```
Host: m2mofficemanagementservice.com
Content-Type: application/xml; charset=UTF-8
AuthId: aabcc
AuthKey: ddeeff
<?xml version="1.0" encoding="UTF-8"?>
<contentInstance>
<content>DATA</content>
</contentInstance>
HTTP/1.1 201 Created

Content-Type: application/xml; charset=UTF-8
Location: m2m.officemanagementservice.com/OMS/scl
```

```
s/SC01/containers/contCollection/contentInstances/CI01
<?xml version="1.0" encoding="UTF-8"?>
<contentInstance id="CI01">
<creationTime>2014-07-21T17:50:35</creationTime>
<lastModifiedTime>2014-07-21T17:50:35</lastModifiedTime>
<content>DATA</content>
</contentInstance>
```

데이터 저장 이후 NA에서는 두가지 경우의 센서 수집 데이터 요청을 할 수 있고, 하나는 주기 수집 데이터 요청이고, 또 다른 하나는 주기 수집 전송 데이터 요청이다. 주기 수집 데이터 요청의 경우는 contentInstanceRetrieve 프리미티브로 NSCL에 보내며, 수신한 NSCL은 요청받은 주기보고 데이터를 조회하여서 주기보고 데이터를 사무실관리 서버로 리턴 한다.

GET /OMS/scls/SC01/containers/CO01/contentInstances/CI01

```
Host: officemanagementservice.com
Content-Type: application/xml; charset=UTF-8
Authorization: zzzzbbbb
HTTP/1.1 200 OK
```

```
Content-Type: application/xml; charset=UTF-8
<?xml version="1.0" encoding="UTF-8"?>
<contentInstance id="CI01">
<contentType>text/plain</contentType>
<contentSize>5</contentSize>
<creationTime>2014-07-21T20:00:00</creationTime>
<lastModifiedTime>2014-07-21T20:00:00</lastModifiedTime>
<content>Fault</content>
</contentInstance>
```

주기전송 데이터 요청의 경우는 subscriptionCreate 리미티브로 주기보고 데이터 전송을 받을 URL 등록 요청을 NSCL에 보내며, 수신한 NSCL은 요청받은 URL 등록을 하고, NA로 HTTP/1.1 201 Created 처리 한다.

POST /OMS/scls/SC01/containers/CO01/contentInstances/subscriptions

```
Host: officemanagementservice.com
Content-Type: application/xml; charset=UTF-8
Authorization: kkljjjjj
```

```
<?xml version="1.0" encoding="UTF-8"?>
<subscription>
<contact>http://subs.mydomain.com/receive</contact>
</subscription>
```

**HTTP/1.1 201 Created**

Content-Type: application/xml; charset=UTF-8  
 Location: m2m.officemanagementservice.com/OMS/scls/SC01/containers/CO01/subscription/SS01

```
<?xml version="1.0" encoding="UTF-8"?>
<subscription id="SS01">
<creationTime>2014-07-21T21:00:00</creationTime>
<lastModifiedTime>2014-07-21T21:00:00</lastModifiedTime>
<subscriptionType>SYNCHRONOUS</subscriptionType>
<contact>http://subs.mydomain.com/receive</contact>
</subscription>
```

이후 subscriptionNotify 프리미티브를 이용 사무실 관리 서버로 주기전송 Data를 Push 처리 한다.

**POST <http://subs.mydomain.com/receive>**

Host: m2m.officemanagementservice.com  
 Content-Type: application/xml; charset=UTF-8  
 Authorization: kkhnnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<contentInstance>
<content>Fault</content>
</contentInstance>
```

**HTTP/1.1 202 Accepted**

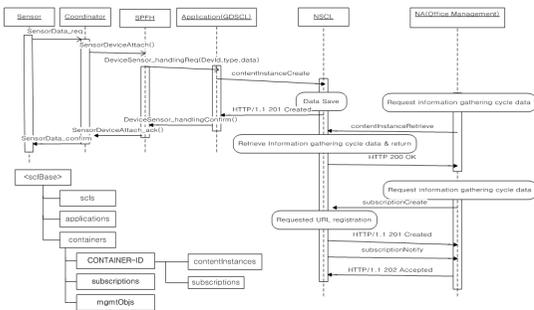


그림 8. 데이터 전달 시나리오 및 리소스 트리  
 Fig. 8 Data transfer scenario and resource tree

**IV. 지능화 모델링을 통한 사무실 관리 서비스**

각종 센서 에서 측정되는 데이터를 근거로 사무실 관리 서비스 서버가 처리해야하는 서비스 요소와 그에 따른 기능들 그리고 기능 들이 맵핑되어야할 서비스를 표 2와 같이 제시한다.

표 2. 서비스 기능 관계  
 Table 2. Relation of service function

Office Management Service Data		Service element	Function	Service	Related Sensor
Name of Sensor Device	Mea. Unit				
Air conditioning	Temperature	Air conditioning On/Off Service Profile management Service	Pre-Arranged Temperature decision On/Off Handling of Office profile information Request of manager On/Off Super manager On/Off Monitoring of Temperature Sensor	Status Event Service Status Event Service Triggering Service Triggering Service Collation Service	Temperature Sensor Profile Data
Heating	Temperature	Heater On/Off Service Profile management Service	OffPre-Arranged Temperature decision On/Off Handling of Office profile information Request of manager On/Off Super manager On/Off Monitoring of Temperature Sensor	Status Event Service Status Event Service Triggering Service Triggering Service Collation Service	Temperature Sensor Profile Data
Lamp	lux	Illuminance service by Office environment	Adaptation of Illuminance by environment	Status Event Service	Illuminance Sensor
Magnetic	Gauss	Office Open/Close Service Office Open/Close Service by Risk	Open/Close decision by GPS data Open/Close decision by manager request Open/Close decision by fire	Status Event Service Triggering Service Status Event Service	Open/Close Sensor Fire Sensor
Fire detector	smoke mass concentration	Fire detection	Open/Close Service by fire Fire Service	Status Event Service Triggering Service	Fire Sensor Risk Management
Glass detector	Frequency	External press detection	Open/Close Service by external press	Status Event Service	Press Sensor
In/Out Card Reader	Image	Image Service about IN/OUT Person	In/Out Image Information Report In/Out Image Information Trace	Collation Service Collation Service	Card Reader & Image Image Trace
Temperature	Mea. Temperature	Mea. Temperature	Mea. Temperature	Collation Service	Temperature Sensor
humidity	%	Mea. humidity	Mea. humidity	Collation Service	Humidity Sensor
Infrared light detector	εV	Infrared light detection	Infrared light detection event Handling of Security camera	Status Event Service Triggering Service	Infrared light detection Sensor Security camera
Security camera	Image	Image detection	Handling of Risky action Protection of security area	Status Event Service Status Event Service	Security camera

상기의 표에서와 같이 사무실 관리 서비스는 센서 데이터 처리 서비스, 상태 이벤트 서비스, 트리거링 서비스들로 구성한다. 서비스 지능화 모델링을 도입하여, 통합적으로 기능의 지능화를 책임질 트리거 처리블록과 이벤트 처리 블록 그리고 NSCL 메시지 처리 블록으로 구성한다. 그리고 세 가지 서비스(센서 데이터 처리 서비스, 상태 이벤트 서비스, 트리거링 서비스)들은 독립적인 기능 블록들로 구성되어, 독창적인 서비스 기능을 수행할 수 있다. 센서 데이터 처리 서비스 부분은 통계처리 블록, 관리처리 블록, 온도센서 측정 모니터링 블록, 출입 인식 영상정보 보고 블록, 출입인식 영상 정보 트래이스 블록, 온도 측정 블록, 습도 측정 블록으로 구성된다. 상태 이벤트 서비스 부분은 적정온도 사전 판단 자동 ON/OFF 블록, 서비스 등록 시 사무실 profile 정보처리 블록, 조도

사무환경별 상태 처리 블록, 화재에 따른 개폐 블록, 외부 충격 발생 처리 블록, 적외선 감시 이벤트 경고 블록, 돌출 행위 판단처리 블록, 사무실 보안구역 침입 처리블록으로 구성된다. 트리거링 서비스 부분은 관리자 요청 On/Off 블록, 슈퍼관리자 On/Off 블록, 출입 센서 On/Off에 따른 보안카메라 작동 및 중지 블록, 관리자 요청에 따른 개폐 블록, 화재 발생 처리 블록, 외부충격발생 처리 블록, 출입 시도에 대한 보안카메라 처리 블록으로 구성된다. 많은 블록 중에 그림 9에 제시된 상태 이벤트 서비스 부분의 화재에 따른 개폐 블록에 대해서만 기능 설명을 하도록 한다. 게이트웨이에 화재발생에 따르는 상태 이벤트 서비스가 등록되어 있었고, 서버에서는 디스커버리(Discovery) 리소스에 의해서 사무실 화재 발생에 대한 이벤트 처리가 점검되고 있는 상황이다. 이때 사무실 화재 발생이 되어서 이벤트가 발생되었음을 디스커버리 기능 작동에 의해 사무실 관리 서버가 감지하면, 화재에 따른 개폐 블록으로 화재 이벤트를 전달한다. 이를 수신한 해당 블록은 사무실 상태 파악하기 위해 해당 office\_id를 가지고 프로파일 정보를 확인하여, 보안 카메라의 숫자와 디바이스 인식자를 확인하고, 사무실 상태 인식 트리거 요청을 한다. 트리거 요청이후 슈퍼 관리자의 스크린을 통해서 멀티 세션 처리된 보안 카메라 영상 정보가 전달된다. 트리거 요청시 중재 서비스를 요청 하며, 트리거에 의해 사무실의 근무 중인 사람의 존재 유무가 파악된다. 만일 사람이 존재 하면 모든 개폐 센서에서 Open 처리하며, 인체가 존재하지 않으면 Close 처리한다. 마지막으로 보안 감시 카메라의 영상 정보 수집 트리거의 종료는 슈퍼 관리자가 상황종료 되었을 때 트리거 종료 처리 명령을 받아서 블록의 기능들이 종료 된다[7].

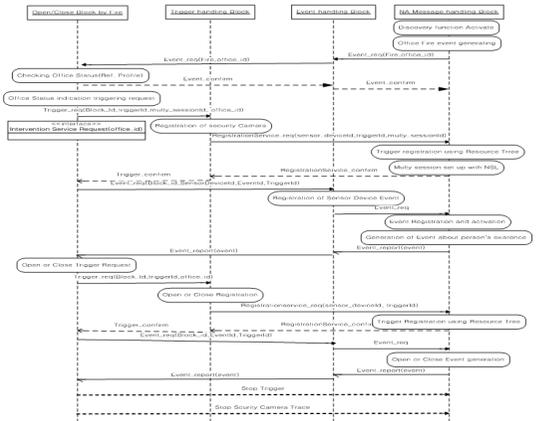


그림 9. 화재에 따른 개폐 블록 처리도 및 시나리오  
Fig. 9 Handling diagram of open/close block and scenario by fire

V. 결론

본 논문에서는 사무실 관리 서비스 플랫폼 구조와 기능을 통해 게이트웨이를 이용한 기능화된 사무실관리 서비스들과 게이트웨이 기능 구조, 코디네이터 클래스 다이어그램, 리소스 트리 설계, 사무실 관리 서버 연동 정보전달 시나리오가 제시됨으로 하여 사무실 환경에서 일어날 수 있는 다양한 상황을 점검 할 수 있도록 하였다.

본 연구를 하면서 센서 데이터의 활용 측면에서 빅데이터 플랫폼의 필요성을 느끼게 되었으며, 향후 빅데이터 플랫폼과 연동되는 센서 데이터를 이용한 다양한 데이터 패턴을 분석 반영하는 진일보된 시스템으로 발전시킬 계획이다.

감사의 글

본 논문은 2014년도 광주대학교의 연구비의 지원을 받아 수행되었음

References

[1] S. Kim, K. Kim, and Y. Shon, "Information Analysis as Keyword of integrated IoT and Advanced Leisure Sport," *J. of the Korea*

*Institute of Electronic Communication Sciences*, vol. 9, no. 5, May. 2014, pp. 609-616.

- [2] K. Jeong and W. Kim, "The Implementation of Smart Raising Environment Management System based on Sensor Network and 3G Telecommunication," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 6, no. 4, Aug. 2011, pp. 595-601.
- [3] J. Kim, "A cluster head replacement based on threshold in the Internet of Things," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 9, no. 11, Nov. 2014, pp. 1241-1248.
- [4] B. Choi, S. Eun, and B. Kim, "Design and Implementation of M2M Platform based on PWW," *J. of The Korea Institute of Information and Communication Engineering*, vol. 17, no. 3, Mar. 2013, pp. 740-746.
- [5] Y. Kim, Y. Jun, and I. Chong, "Device Objectification and Orchestration Mechanism for IoT Intelligent Service," *J. of The Korea Institute of Communications and Information Sciences*, vol. 38C, no. 01, Jan. 2013, pp. 19-32.
- [6] J. Park, and S. Lee, "Development of Safety Monitoring System for Multi Homes and Spaces," *J. of Korean Society for Internet Information*, vol. 15, no. 1, May. 2014, pp 207-208.
- [7] H. Choi, J. Lee, N. Yang, and W. Rhee, "Ontology Based User-centric Service Environment for Context Aware IoT Services," *J. of The Korea Contents Association*, vol. 14, no. 7, July. 2014, pp. 29-44.

## 저자 소개



### 남강현(Kang-Hyun Nam)

2003년 용인대학교 경영정보학과 졸업(이학사)

2006년 경희대학교 대학원 정보통신학과 졸업(공학석사)

현재 광주대학교 정보통신학과 교수

1986년~2006년 삼성전자 Core망 개발팀 근무

2013년~현재 산업통산자원부 이동통신분야 산업기술평가단 위원

2014년~현재 사물인터넷포럼 기술분과위원회 위원

2014년~현재 사물인터넷포럼 표준분과위원회 위원

※ 관심분야 : 사물지능통신, 빅데이터 플랫폼, SDN

