

IoT (Internet of Things) 활성화 방안

• 김종영 (한양대학교 컴퓨터공학과)

I. 사물인터넷

1. 사물인터넷 정의

사물인터넷(이하 IoT)는 사물, 동물, 사람 등의 주체가 고유 식별자를 부여받은 다음 네트워크를 통해 데이터를 주고 받는 환경을 말한다. 데이터 송수신 과정에 사람-사람, 사람-컴퓨터 간 인터랙션은 필요하지 않다. IoT는 무선 기술, MEMS (Micro-ElectroMechanical System), 인터넷의 융합의 결과라고 볼 수 있다.

용어	발표기관	주요 내용
M2M	IEEE	- 가입자 장치와 기지국을 거쳐 코어-네트워크에 위치하는 서버간의 정보교환 혹은 가입자 장치간 인간의 개입 없이 발생하는 정보 교환
	ETSI	- 인간의 직접적인 개입이 꼭 필요하지 않은 둘 혹은 그 이상의 객체간에 일어나는 통신
IoT	ITU	- 모든 사물에게 네트워크 연결을 제공하는 네트워크의 네트워크
	EU	- 객체간의 통신이 가능한 네트워크와 서비스
	CASAGRAS	- 데이터 수집과 통신기능을 통하여 물리적 객체와 가상의 객체를 연결해 주는 글로벌 네트워크 기반 구조
	IETF	- 표준 통신 프로토콜을 기반으로 독자적인 주소를 가지며 상호 연결된 객체들의 전 세계 네트워크
MTC (Machine Type Communication)	3GPP	- 인간의 개입이 꼭 필요하지 않은 하나 혹은 그 이상의 객체가 관여하는 데이터 통신의 형태
MOC (Machine Oriented Things)	ITU-T	- 인간의 직접적인 개입이 최소한으로, 혹은 요구되지 않은 둘 혹은 그 이상의 객체간의 통신
IoE (Internet of Everything)	CISCO GE	- 사람과 사물에 이어 프로세스와 데이터가 상호 밀접하게 연결되어 있는 새로운 형태의 네트워크 환경

그림 1. 각 기관별 M2M/IoT 용어 정의[1]

2 사물인터넷 서비스 동향

구글은 32억 달러를 들여 네스트랩을 인수했다. 네스트랩의 온도 조절기는 사용자의 사용 패턴 데이터를 수집하여 사

용자의 온도 설정을 학습한다. 일주일 정도의 사용기간을 통해 네스트랩 온도 조절기는 자동으로 알아서 집안 온도를 적절하게 제어한다. 동작 인식 센서도 포함되어 있어 외출 시 온도를 알아서 내려주고 귀가 시간이 되면 집안 온도를 자동으로 높게 된다.

아마존이 개발한 에코(Echo)는 가정용 음성인식 기기이다. 사용자가 정보, 음악, 뉴스, 날씨 등을 물어보면 대답을 하고 클라우드에 연결되어 데이터가 축적될수록 학습 능력이 높아진다. 원통형의 기기에 7개의 마이크가 내장되어 있고 소음 제거 기능이 탑재되어 있어 인식 성능을 높였다.

일본 소프트뱅크의 페퍼(Pepper)나 미국 MIT의 지보(Jibo)는 홈로봇으로서 가족들의 생김새나 목소리 정보를 이용하여 개인을 구분하고 대화를 할 수 있는 기능이 있다.

삼성이 인수한 SmartThings는 스마트 홈의 주요 기능으로 안전/모니터링, 엔터테인먼트, 편리성/절약을 강조하고 삼성의 다양한 가전기기를 하나로 묶는다는 계획을 발표했다.

커넥티드 카라고 불리는 개념은 텔레메트릭스(Telematics)을 자동차 분야에 발전 적용한 것이다. 텔레메트릭스는 원거리를 뜻하는 tele와 계측을 뜻하는 metry, 전자 공학을 뜻하는 electronics가 합쳐진 단어이다. 양방향 통신과 원격자동계측 기술의 결합으로 인해 전기/가스/수도 원격 자동검침, 국가 기반시설물 안전관리, 환경관리 등에 활용되고 있다. 커넥티드 카의 주요 요소에는 스마트 단말, 음성 인식 기술, 인포테인먼트, 차량 간 통신, 자체 진단 기능, 차량 인프라 통신 등이 있다.

애플은 BMW, GM, Mercedes, Jaguar, Audi, Chrysler, Toyota, Honda에 자사의 음성인식시스템인 시리를 탑재한 것을 시작으로 인포테인먼트 시스템인 카플레이(CarPlay)를

소개했다. 카플레이는 운전자가 음성 또는 터치로 전화를 걸거나 음악을 감상하거나 메시지 확인, 길 찾기 등을 할 수 있게 한다. 센서가 설치된 도로 인프라와 데이터를 주고 받으며 교통체증 해결이나 안전도 향상에 기여할 것으로 기대된다.

부산시는 SKT, Cisco와 2014년 부산에서 열린 ITU 전권 회의에서 스마트 시티 조성을 위한 업무협약 및 기본협약 (Framework Agreement)을 체결하고 다양한 사업을 추진하기로 했다. 부산시는 이어 2015년 11월 2019년까지 1035억을 투자해 안전/관광/교통/에너지/환경/생활 등의 25개 분야에서 스마트 시티를 단계적으로 구축하기로 했다. 특히 스마트 가로등, 안심 서비스, 미아 방지 서비스, 해상 안전 서비스 분야를 실증사업으로 정하고 국제표준기반의 개방형 플랫폼을 적용하여 올 12월 서비스 개시를 한다고 밝혔다. 센텀시티를 비롯한 해운대 지역에 10개 항목의 IoT 서비스를 시작하기로 했는데 가로등, 횡단보도, 주차장 등의 시설이 첨단화된다. 가로등은 사고 동영상 녹화, 시간에 따른 자동 밝기 조절 등의 기능을 갖출 것으로 알려져 있다. 횡단보도는 사람의 움직임을 감지하며 주차장, 매장 등 주변 정보를 제공하는 역할도 하게 된다[1][2].

II. 사물인터넷 플랫폼 동향

1. AllJoyn

AllJoyn은 Qualcomm Innovation Center에서 시작된 오픈 소스 API 프레임워크로서 디바이스와 어플리케이션이 네트워크 상에 API를 “내어놓는” 표준 방법이다. API의 실체는 네트워크 상의 “thing”이 다른 “thing”에 노출하는 기능이다. 예를 들면, 온도, 시간 등이 해당된다. 디바이스와 서비스 사이의 상호운용성에 핵심적인 내용이 바로 API가 된다. AllJoyn은 어떤 어플리케이션에 어떤 API가 사용가능한지를 알아내는(discovery) 방법을 제공한다. 이 작업은 플랫폼 또는 연결 방식에 무관하게 진행된다. 여기서 말하는 어플리케이션이란 마이크로컨트롤러 펌웨어가 될 수도 있고, 스마트폰 어플리케이션, PC 또는 서버 상의 어플리케이션이 될 수 있다.

외부에 노출한 API와 AllJoyn 어플리케이션은 버스를 통해 통신한다. 통신에서 주고받는 내용은 메시지이며 해당

API로 맵핑된다. 메시지에는 다음과 같은 네가지 종류가 있다.

- Signal(비동기식): 이벤트 관련 브로드캐스팅, 멀티캐스팅 또는 P2P를 사용하여 메시지를 수신하는 단위를 정할 수 있음
- Method Call(동기식): RPC (Remote Procedure Call)의 호출 부분
- Method Replies(동기식): RPC (Remote Procedure Call)의 응답 부분
- Error(동기식): RPC (Remote Procedure Call)에 대한 에러 응답

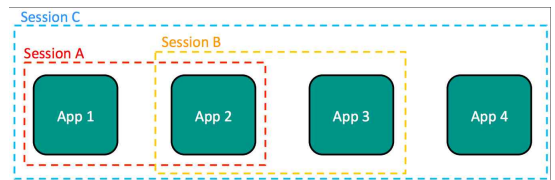


그림 2. AllJoyn 세션

AllJoyn의 메시지에 포함되는 정보에는 일련번호, 송신자, 목적지, 객체 경로(object path), 인터페이스, 멤버, 시그니처, 세션 ID 및 페이로드(payload)가 있다. AllJoyn은 분산 소프트웨어 버스를 구현했으며 버스 포메이션은 Ad Hoc 방식이며 프로토콜은 네트워크 독립적이다. 연결 프로토콜은 D-Bus를 확장했으며, Wi-Fi, Wi-Fi Direct, 이더넷, PLC, 블루투스 상에서도 실행이 가능하다. 세션이란 연결되어 있는 어플리케이션 그룹을 말하는데 세션내의 어플리케이션들은 데이터를 주고 받을 수 있다.

AllJoyn 버스는 두 가지 종류의 노드가 있는데 라우팅 노드(RN; Routing Node)와 리프 노드(LN; Leaf Node)로 나뉘어진다. RN은 RN끼리만 연결이 가능하며 LN은 RN에만 연결이 된다. AllJoyn 데몬은 transport 메커니즘, 메시지 라우팅, 네임스페이스 관리를 관리하는 추상 레이어를 제공하는데 어플리케이션은 데몬과의 통신만을 수행하면 된다. AllJoyn 어플리케이션이 다른 디바이스와 통신하기 위해선 디바이스 상에 AllJoyn 데몬이 실행되고 있어야 한다. 어플리케이션은 오직 데몬과 통신하기 때문에 개발자는 내부적인 통신 프로토콜에 대해 신경을 쓸 필요가 없다. 개발자는 데몬

에 서비스를 등록하여 서버 구성을 하거나 등록된 서비스를 검색하여 클라이언트를 구현한다.

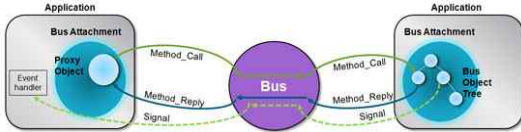


그림 3. AllJoyn bus attachment

AllJoyn 어플리케이션은 소위 “bus attachment”라는 것이 필요한데 이는 객체 계층구조의 루트(/)를 제공하기 위한 목적이다. 이를 통해 어플리케이션은 버스와 통신할 수 있게 된다(그림 3 참조). 각 attachment에는 AllJoyn 인터페이스를 구현한 버스 객체가 포함되는데 버스 객체에는 파일 경로와 유사한, 예를 들어 “/org/AllJoyn/Games/chess”와 같은 경로 이름(path name)이 들어있다. 이 경로 이름은 RPC를 통해 원격 호출이 된다. 버스 객체가 AllJoyn 인터페이스를 구현했기 때문에 시그널, 메시지 전송이 가능한 것이다. RPC를 호출하는 측에서는 로컬에 위치한 프락시 버스 객체가 어플리케이션의 원격 버스 객체 호출을 대신해 주는 기능을 담당한다. 그림 3에서 왼쪽 편의 어플리케이션이 클라이언트에 해당되고 오른쪽 편의 어플리케이션이 서비스 기능을 제공한다. 클라이언트 측에서 Method_Call은 프락시 버스 객체 및 버스를 통해 서비스 측에 전달된다. 서비스는 Method_Reply를 클라이언트 측에 전달하게 되는데 만약 이 과정에서 오류가 발생하면 Method_Error 응답이 전송된다. 버스에 연결된 어플리케이션은 이벤트 핸들러를 통하여 들어오는 시그널을 수신할 수 있다. 앞서 시그널은 비동기 방식이라고 했는데, 따라서 이벤트 핸들러는 들어오는 시그널 수신에 대해 응답하지 않는다.

그림 3은 두 대의 AllJoyn 디바이스가 연결하는 과정을 나타낸 것이다. 한 쪽이 Find-Name 요청을 보내면 다른 한 쪽은 Advertise-Name 요청을 보내게 된다. 이로써 두 디바이스는 상대방을 발견하게 되고 페어링 상태가 된다. 즉 ad hoc 버스가 만들어진 것이다. Discovery 과정은 전송방식에 따라 다른데 Wi-Fi의 경우 lightweight IP multicast protocol을, Bluetooth 방식의 경우, EIR (Extended Inquiry Response)와 SDP (Suspended Discovery Protocol)을 합

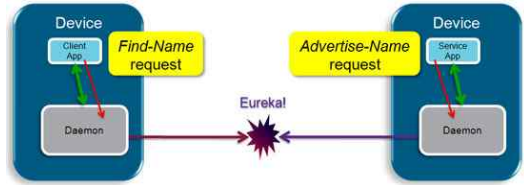


그림 4. Discovery 과정

게 사용한다. 연결이 되고 나면 데몬은 네임스페이스를 공유하는 단일 버스를 만들게 되고 이를 통해 RPC 또는 메시지 전송을 하게 된다.

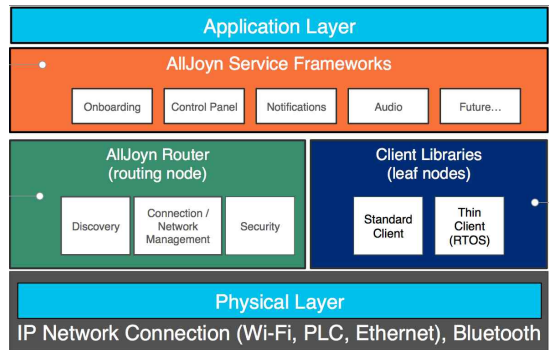


그림 5. AllJoyn 소프트웨어 프레임워크

AllJoyn 소프트웨어 프레임워크는 Application Layer와 Physical Layer를 제외하고 다음의 두 가지 주요 소프트웨어 구성 요소로 이루어져 있다.

- Router: Router를 가지고 있는 노드가 바로 RN이 된다. Router가 표준 클라이언트 라이브러리를 사용하여 만들어졌기 때문에 반드시 HLOS상에서 실행되어야 한다. Router가 수행하는 두 가지 중요한 일은 다음과 같다.
 - 버스 관리: 버스를 통해 연결된 어플리케이션의 주소관리 목적의 네임스페이스 관리 및 디바이스 간 통신
 - 메시지 라우팅
- Client 라이브러리: AllJoyn에서 모든 어플리케이션은 RN의 클라이언트이기 때문에 Client 라이브러리라는 이름을 사용하며 개발

자가 사용하는 API 집합임. Client 라이브러리는 다음 두가지 형태로 구현됨.

- Standard Client (SC): HLOS에서 구동되는 어플리케이션용, 고수준의 추상화 제공, 복잡한 멀티쓰레딩 기능 제공
- Thin Client (TC): 저용량 메모리 디바이스용, C언어로만 제공

2. IOTivity

IOTivity는 오픈소스 프로젝트로서 Linux Foundation이 운영하며 삼성전자, 인텔이 참여하고 있는 OIC (Open Internet Consortium)에서 후원하고 있다. IOTivity 프로젝트의 목표는 수십억개의 유/무선 디바이스들이 상호 또는 인터넷에 연결되게 만드는 새로운 표준을 만드는 것이며, 확장 이용이하며 강건한(robust) 아키텍처를 지향한다.

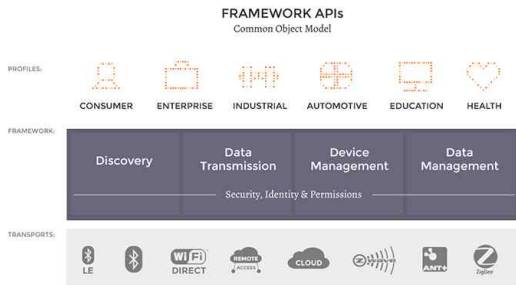


그림 6. IOTivity 프레임워크

IOTivity 프레임워크는 Linux, Android, Tizen, Arduino 환경에서 미들웨어로 동작하며 개발자는 IOTivity 프레임워크 API를 사용하게 된다. API는 “resource-based, RESTful” 아키텍처 모델을 채택하고 있다. IOTivity 프레임워크의 주요 구성 요소는 다음과 같다(그림 6 참조).

- Discovery
- Data transmission
- Data management
- Device management

Discovery는 근접 및 원격 장치 또는 자원 발견을 수행, Data transmission은 메시징/스트리밍 모델 기반 정보 교환 및 제어를 담당, Data management는 다양한 자원에서 수집된 정보의 저장, 분석을 담당, Device management는 장치의 구성, 배치, 진단을 담당하는 부분이다.

OIC 아키텍처는 리소스 기반 IoT 디바이스, 어플리케이션 간 상호작용을 용이하게 한다. OIC 아키텍처는 기존 산업 표준 및 기술을 활용함과 동시에 폼 팩터, 운영체제, 서비스 제공자와 상관없이 기기 간 정보 흐름을 관리하며 유/무선 형태의 연결을 관리하는 솔루션을 제공한다. OIC 아키텍처는 다음과 같은 기능을 제공한다.

- 다양한 시장 영역, 운영체제, 플랫폼, 통신 방법 등에 대해 통신 및 상호 운용성 프레임워크
- 환경을 일관적이며 공통적으로 묘사할 수 있는 모델과 정보 및 의미론적 상호 운용성
- 발견 및 연결 목적의 공통 통신 프로토콜
- 공통 보안 및 식별 메커니즘
- 혁신 및 제품 차별화 가능성
- 스마트 디바이스에서부터 가장 작은 사이즈의 커넥티드 디바이스 및 웨어러블 디바이스에 확장이 용이한 솔루션

OIC 아키텍처에서 실제 세상의 개체(entities)들은 리소스로 표현된다. 개체와의 상호작용은 대상 개체의 표현(representation)에 의해 RESTful 스타일로 이루어진다. 그림 7과 같이 OIC는 전체 OIC 프레임워크 구조를 일종의 정보 시스템으로 정의했으며 OIC 프레임워크를 구성하는 각 요소 간의 관계를 정의하고 있다. 개체는 OIC 리소스로 표현되며 식별자 역할을 하는 URI를 부여받는다.

각 OIC 리소스는 RESTful 작업이 가능하게끔 인터페이스를 제공해야 한다. 모든 RESTful 작업은 그것을 시작한 주체(클라이언트)와 해당 작업에 대해 응답을 하는 주체(서버)가 있다. 모든 OIC 디바이스는 OIC 클라이언트가 되어

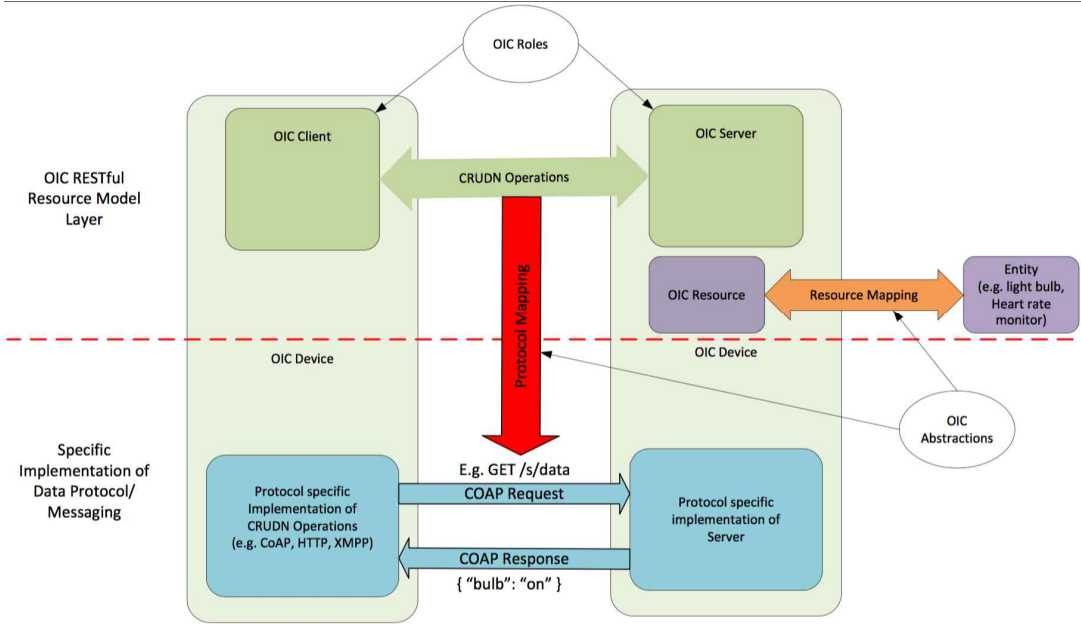


그림 7. OIC 아키텍처 개념도

OIC서버에 RESTful 작업을 요청할 수 있다. 마찬가지로 모든 OIC 디바이스는 OIC 서버 역할을 수행하여 개체를 OIC 리소스 형태로 노출할 수 있다. OIC의 RESTful 작업은 상호작용의 명확성을 위해 필요한 모든 정보를 담고 있으며 최소한의 기본 작업으로 이루어진다. 기본 작업에는 Create, Read, Update, Delete, Notify (CRUDN)이 해당된다.

OIC 아키텍처는 개념적으로 리소스 모델, RESTful 작업, 추상화의 세부분으로 나뉜다. 리소스 모델은 애플리케이션과 그 환경을 논리적으로 모델링하고 논리적으로 연산을 표현할 수 있는 추상화와 개념을 제공한다. 프로토콜과 기술에 독립적인 방법으로 OIC 자원들간의 상호작용을 RESTful 스타일을 적용하여 정의한 것이 CRUDN 작업이다. 리소스 모델의 추상화와 RESTful 작업을 맵핑하여 구체적인 대상과 연결할 때 사용하는 것이 추상화에 해당된다. 개체를 OIC 리소스와 연결할 때 사용하는 entity handler, 논리적인 RESTful 작업을 데이터 통신 프로토콜 및 해당 기술로 연결할 때 사용하는 connectivity abstraction primitive 등이 추상화 범주에 속하게 된다.

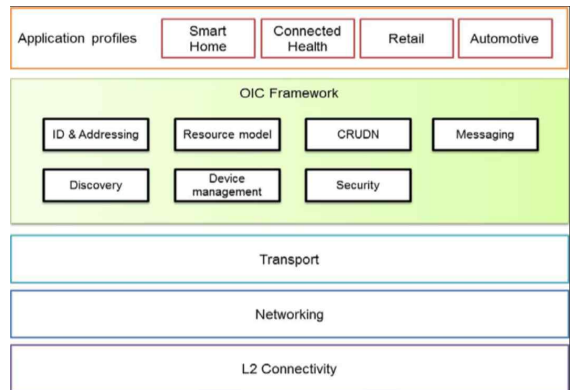


그림 8. OIC 아키텍처의 기능적 구분

그림 8은 OIC 작업에 요구되는 모든 기능을 망라하여 보여준다. 이런 기능들은 다음과 같이 분류할 수 있다[3].

- L2 connectivity: physical 및 data link 레이어 수준의 연결을 할 때 필요한 기능을 제공 (예를 들어, WiFi 또는 Bluetooth)
- Networking: OIC 장치 간 인터넷 상 데이터

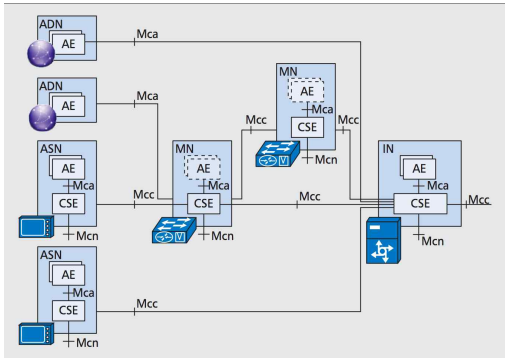


그림 9. oneM2M 기능 아키텍처

교환을 위해 필요한 기능(예를 들어,

IPv4/IPv6)

- Transport: 특정 QoS 조건을 만족하면서 end-to-end 전송에 필요한 기능으로서 기존 TCP, UDP 뿐만 아니라 현재 개발되고 있는 트랜스포트 프로토콜도 포함(예를 들어, Delay Tolerant Networking)
- OIC framework: OIC 코어 기능을 말할
- Application profile: 특정 시장 분야에 특정된 데이터 모델 및 기능을 제공(예를 들어, 스마트 홈 데이터 모델 및 관련 기능)

3. OneM2M

oneM2M은 한국의 TTA, 북미 ATIS, 일본 TTC, 중국 CCSA, 유럽의 ETSI, TIA로 이루어진 국제 협력 프로젝트로서 2015년 9월 14일부터 16일까지 프랑스 파리의 소피아 앙티폴리스에서 첫 상호 운용성 개최 행사를 개최하였다. 이 행사의 목적은 표준에 정의된 기본 상호운용성을 검증하고 oneM2M 디바이스 종단 간의 기능 확인을 목표로 했다. oneM2M에서 개발한 공통 플랫폼 기술은 등록, 검색, 보안, 그룹관리, 가입/알림, 디바이스 관리, 어플리케이션 관리, 데이터 전달 관리, 네트워크 서비스 이용, 위치 정보 관리, 서비스 과금으로 이루어져 있다.

그림 10은 oneM2M의 기능 아키텍처이다. oneM2M에서 기능을 담당하는 개체를 노드로 표현하는데 어플리케이션 전용 노드(ADN), 어플리케이션 서비스 노드(ASN), 미들노드

(MN), 인프라스트럭처 노드(IN)로 분류된다. 각 노드에는 적어도 하나의 oneM2M CSE (Common Service Entity) 또는 oneM2M AE (Application Entity)가 있어야 한다. CSE는 논리적인 단위로서 M2M 노드에서 구체화되며 CSF (Common Service Functions)라고 하는 서비스 기능 집합을 구성한다. CSE는 어플리케이션 또는 다른 CSE가 사용하게 된다. AE는 원격 현황 관리와 같은 end-to-end M2M 솔루션에 필요한 어플리케이션 로직을 제공한다.

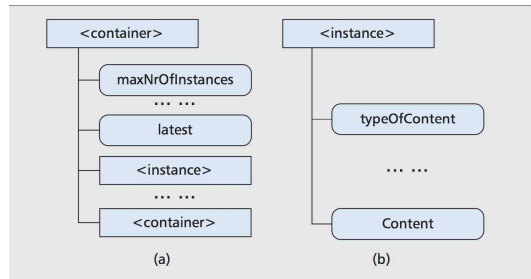


그림 10. oneM2M 자원 표현

oneM2M은 자원 기반 데이터 모델을 채택하였는데 모든 서비스는 자원으로 표현된다. 자원의 실체는 URI를 사용하여 식별되는 자료구조이다. 각 자원은 CSF와 연결되어 있어 CSF 기능을 지원한다. 사용자 데이터 저장 및 처리를 담당하는 CSF를 DMR (Data Management and Repository)라고 하는데 그림 10은 그와 관련된 자원 container와 instance를 나타내고 있다. container 자원은 다수의 속성이 있으며 instance, container와 같은 또 다른 자원을 가질 수 있다. instance 자원은 container 자원에 들어가는 데이터 인스턴스에 해당된다. 자원 및 속성에 대한 작업은 CRUD (Create, Retrieve, Update, Delete)를 통해 이루어진다.

oneM2M에서 디바이스는 관리와 추상화라는 두 가지 관점에서 접근한다. 디바이스 관리(Device Management; DM)는 디바이스 설정, 장애 관리, 진단 등의 여러 가지 사항에 관련된 솔루션을 제공한다. 디바이스 추상화(Device Abstraction; ABS)는 oneM2M 시스템이 외부 시스템과 통신할 수 있게끔 하는 상호운용 메커니즘을 가능하게 한다 [4][5].

III. 사물인터넷 활성화 방안

1. 생태계 구축

사물인터넷 기술을 적용한 도어락 제품, 운동 관리 웨어러블 제품, 디지털 장난감, 식물 관리 등의 제품 등 사물 인터넷 관련 제품을 만들어내는 기업들이 늘어나고 있다. 하지만 이는 기존의 제조업과는 성격이 다르다. 제조 기술뿐만 아니라 소프트웨어, 제품과 소프트웨어가 결합된 형태의 서비스 제공 등 복합적인 내용 구성이 필요하기 때문이다. 이는 모바일 생태계와도 확연히 구분되는데 모바일 생태계는 통신사, 앱 개발 회사, 디바이스 제조 사간의 구분이 명확했기 때문이다. 하드웨어와 소프트웨어를 모두 개발/출시하는 회사는 애플 정도였지만 앞으로의 사물인터넷은 하드웨어와 소프트웨어의 연계가 더 밀착된 생태계 도출이 중요해지는 시장이라고 할 수 있다.

2. 킬러서비스

사물인터넷 시장이 활성화되려면 B2C 시장에서 계기가 만들어져야 할 것으로 보인다. 사물 인터넷은 빅데이터 분석과 이어진다. 소비자들이 자신이 생산한 데이터로부터 혜택을 받는 서비스가 개발된다면 킬러서비스는 가능할 것으로 예상된다.

3. 디바이스 활성화

스마트폰 이후의 디바이스로 웨어러블 디바이스가 그 가능성을 인정받고 있지만 소비자 입장에서 일상생활의 불편한 점이나 라이프 스타일의 주목할 만한 변화를 일으키기에는 아직은 부족한 상황이다. 소비자의 세분화된 욕구를 만족시키기 위해선 다품종+소량생산 전략하에 건강, 스트레스 해소, 스포츠, 교육 등 실용적인 욕구를 만족시킬 수 있는 디바이스 개발이 이루어져야 할 것으로 예상된다.

품 및 서비스가 개발될 것으로 예상된다. 이를 위해 사물인터넷 센서 산업을 육성하고 관련 인력 확충, 사물인터넷 분야에 특화된 통신 기술 및 통신망 구축, 글로벌 민관협의체 구성을 통한 사물인터넷 전문 기업의 육성이 필요하다고 하겠다. 또한 사물인터넷에서 가장 중요한 요구조건 중의 하나인 정보 보안 또한 강화되어야 한다. 이를 통해 개인 수요에 바탕한 개별 맞춤형 서비스 개발을 제공해 사회, 문화, 산업의 새로운 토대로 활용해야 할 것이다.

참 고 문 헌

- [1] ITFIND 주간기술동향, 2013년 10월 30일
- [2] KISTI Market Report(스마트카), Vol. 4, Issue 2
- [3] <https://wiki.allseenalliance.org>
- [4] <http://www.onem2m.org/>
- [5] Jorg Swetina, Guang Lu, Philip Jacobs and et al., Toward a Standardized Common M2M Service Layer Platform: Introduction to oneM2M, IEEE Wireless Communications, June 2014.

저자소개



김 종 영

1996: 한양대학교 전자계산학과 공학사.

1998: 한양대학교 컴퓨터공학과 공학석사.

2012: 한양대학교 컴퓨터공학과 박사과정 (박사 수료)

관심분야: Bayesian Optimization

IV. 결론

현재 정부 주도로 분야별 사물인터넷 실증단지가 조성되는 등 사업화 토대를 마련하기 위해 노력하고 있다. 이를 토대로 사물 인터넷 테스트베드가 마련이 되면 기업을 중심으로 제