

논문 2015-10-16

## 사용자 상황 정보 관리를 지원하는 IoT 통합 제어 모듈 설계 및 구현

(Design and Implementation of IoT Collaboration Module  
Supporting User Context Management)

금 승 우\*, 임 태 범, 박 종 일

(Seung Woo Kum, Tae Beom Lim, Jong Il Park)

Abstract : Various personalized services are provided based on user context these days, and IoT(Internet of Things) devices provides effective ways to collect user context. For example, user's activity such as walking steps, calories, and sleeping hours can be collected using smart activity tracker. Smart scale can sense change of user's weight or body fat percentage. However, these services are independent to each other and not easy to make them collaborate. Many standard bodies are working on the documents for this issue, but due to diversity of IoT use case scenarios, it seems that multiple IoT technologies co-exist for the time being. This paper propose a framework to collaborate heterogeneous IoT services. The proposed framework provides methods to build application for heterogeneous IoT devices and user context management in more intuitive way using HTTP. To improve compatibility and usability, gathered user contexts are based on MPEG-UD. Implementation of framework and service with real-world devices are also presented.

Keywords : Internet of things(IoT), User context, Service collaboration, IoT framework, IoT cloud

### I . 서 론

최근 네트워크 기술의 발전을 통하여 다양한 분야에 IoT 기술이 적용된 제품 및 서비스들이 출시되고 있다. 이러한 IoT 기술은 저전력 센서들을 활용한 센서 네트워크로부터 홈 정보 가전 제어, 멀티미디어 콘텐츠의 공유가 가능한 홈 네트워크 등 다양한 분야에 적용되고 있다. 기존 홈 네트워크나

\*Corresponding Author(swkum@keti.re.kr)

Received: 28 Apr. 2015, Revised: 20 May 2015,

Accepted: 28 May 2015.

S.W. Kum: Hanyang University, KETI

T.B. Lim: KETI

J.I. Park: Hanyang University

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (R0101-15-0159, 기기 정보뿐 아니라 사용자의 환경/감성/인지 정보에 적응적으로 반응하는 정보기기용 원격 UI 기술 개발)

Machine-to-Machine에 대한 IoT 기술의 가장 큰 차별성 중의 하나는 연결성(Connectivity)을 꼽을 수 있다. IoT 기술은 인터넷 기반의 Infrastructure를 사용하여 정보의 수집 및 전송이 이루어지므로 언제 어디서든 사용자의 기기 정보를 확인하고 제어할 수 있는 환경을 제공하고 있다. 기존의 센서 네트워크 혹은 홈 네트워크 등 특정 목적을 위하여 구성되었던 네트워크 기술들이, 인터넷 기반 기술과 접목을 통하여 연결성을 제공받고 사용자에게 보다 풍성한 사용환경을 제공하고 있다. 이러한 IoT 기술은 OneM2M 등을 통한 국제 표준화와 함께, AllSeen Alliance, OIC, UPnP+ 등 다양한 컨소시엄으로부터의 표준화가 각 사용 목적에 맞추어 추진 중에 있다. 또한 시장에서도 IoT 기술을 적극적으로 수용하고 있어 이미 AllSeen Alliance 표준이 적용된 제품 및 서비스가 출시되고 있으며, 표준과 무관하게 자체적인 IoT 서비스를 제공하는 기업들도 수년 전부터 시장에 나타나고 있다. 이러한 제품과 서비스를 통하여 사용자들은 손쉽게 자신이 보유하고 있는 IoT 기기의 상태를 원격으로 확인하고

제어할 수 있다. 예를 들어, 사용자는 Fitbit사의 Activity Tracker를 통하여 수집된 개인의 운동 정보를 스마트폰으로 확인할 수 있으며, 또는 SmartThings 사의 Hub를 사용하여 스마트폰으로 맥내 진동 및 진원의 제어, 현관 도어락의 제어가 가능하다. 이렇듯 IoT 기술의 확산은 인터넷을 통한 자원의 접근을 통하여 사용자에게 보다 높은 품질의 사용 경험을 제공하고 있다. 다만, 이러한 IoT 기술의 적용에 있어서 각 사용에 따른 호환성의 확보는 고려해야 할 대상이다. 일부 업계 표준 기구의 경우 표준 문서가 공개되기도 하였으나 아직까지 국제 표준을 비롯한 IoT 표준화는 진행 중에 있으며, 표준과 무관한 제품들도 이미 시장에 출시되어 서비스가 운영 중에 있다. 이러한 각 IoT 서비스간의 독립성은 서비스의 개발과 사용을 모두 어렵게 하는 요인으로 동작한다. 예를 들어 사용자가 운동을 하고 있을 경우 실내 온도에 맞추어 에어컨을 동작시키기 위한 3종 기기간 연동 어플리케이션의 개발을 위해서는 사용자의 운동을 확인하기 위한 Fitbit사의 Activity Tracker API, 실내 온도를 확인하기 위한 NetAtmo사의 Weather Station API, WeMo의 에어컨 제어를 위한 WeMo 사의 API 등 총 3종의 API를 각각 확인하고 제어하여야 한다. 제어의 기법도 각각 독립된 RESTful API를 호출하고 수신되는 JSON 오브젝트를 각 사의 규약에 맞추어 Parsing해야 한다. 이와 별도로 각 서비스에 접근하기 위한 권한 획득을 위한 Authorization도 독립적으로 처리가 필요하다.

본 논문에서는 이러한 문제점을 해결하기 위하여 사용자의 상황 정보 관리를 지원하는 IoT 통합 제어 프레임워크 제안한다. 제안된 프레임워크는 다양한 이종 IoT 서비스에 대한 통합 관리 및 서비스 구성을 위한 단일 API를 제공하여 IoT 서비스의 개발을 지원하며, 사용자에게 단일화된 어플리케이션을 통하여 보다 높은 수준의 사용자 경험을 제공할 수 있다. 또한, 각 단위 IoT 서비스로부터의 사용자의 상황정보를 통합 관리함으로써 사용자의 상황 정보를 종합적으로 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 연구 동향을 검토하고, 3장에서는 제안된 프레임워크를 설명한다. 프레임워크는 IoT 서비스에 대한 통합관리 프레임워크와 해당 프레임워크를 통한 사용자 상황 정보 관리를 포함한다. 프레임워크를 통한 임베디드 시스템에서의 구현 예는 4장에 기술되었다. 구현 예는 서비스의 구성과 함께 제시된 프레임워크의 실사용성을 검토하기 위한 SmartTV,

SmartPhone 등의 임베디드 시스템을 통한 실제 제품간의 연동을 포함한다.

## II. 관련 연구

IoT와 사용자 상황 정보에 관련된 정보는 최근 많은 연구가 진행되고 있다. 이러한 연구는 상황 정보를 분석하여 서비스를 제시하는 형태를 포함한다. Atzori [1]는 센서 네트워크로부터 수집되는 정보를 소셜 정보에 연동시키기 위한 기술과 고려해야 할 정보들을 논한 바 있다. Jaewon Moon [2]은 스마트 홈 환경에서 사용자 컨텍스트에 기반한 Second Screen 기술을 논하고 있다. Ji-Yeon Son [3]은 홈 내 컨텍스트 정보를 기반으로 서비스를 구성하기 위한 기법을 제시하였다. 이러한 연구들은 단일 네트워크 내에서 컨텍스트 정보를 활용하기 위한 기법을 논하고 있다.

IoT 서비스의 구성 및 운영 방법에 대해서는 Service Oriented Architecture를 기반으로 하는 연구들이 진행되고 있으며, 단일 서비스 혹은 서비스간 연동에 대한 구성 방법을 기술하고 있다. Guinard [4]는 엔터프라이즈급 어플리케이션과 실제 물리 기기간의 연동을 구성하는 방법을 제시하였으며, PatRICIA [5]는 기기간 연동을 위한 정보를 Intent로 정의하여 IoT 기기간의 연동을 구성할 수 있는 프레임워크를 제시한 바 있다. 이러한 연구들은 단일 서비스의 구성을 위한 SOA 기반의 아키텍처를 제시하고 있다. Ji-Hoon Lee [6]는 헬스케어 정보를 관리하는 게이트웨이 기반의 서비스를 제시한 바 있다. 호환성의 보장을 위해 Gateway 구조를 사용하는 연구도 다방면에서 진행 중이다. Bosman [7]은 SOA 구조와 홈 네트워크 프로토콜의 호환성 확보를 위한 구조를 제시한 바 있으며, Cirani [8]는 서비스 발견을 위한 게이트웨이 구조를 제안한 바 있다. 해당 기술들은 이종 프로토콜간의 변환 혹은 연동을 위한 특정 하드웨어 및 소프트웨어 구성을 요구하고 있다.

상황 정보의 정량적 표현을 위한 메타데이터 기술 방법에 대한 연구도 지속적으로 진행되고 있다. 특히, MPEG은 멀티미디어 생성 및 소비 환경에 대한 상황 정보를 지속적으로 정의 및 갱신하고 있다. MPEG-7, MPEG0-21, MPEG-V 등의 표준은 멀티미디어 콘텐츠의 생성 및 소비 환경에 대한 정의를 포함하고 있으며, 최근 MPEG 105차 회의에서는 미디어 소비 환경 뿐 아니라 다양한 환경에서의

서비스 추천에 대한 호환성을 제공할 수 있는 MPEG-UD 표준에 대한 Call for Contribution [9]이 승인되었다. MPEG-UD는 111차 회의에서 Committee Draft가 제출되었다.

단일 서비스 구성이 아닌 서비스간 연동을 위하여 SOA 아키텍처 외의 Application Layer에서의 이중망 연동에 대한 연구 및 제품화도 일부 확인되고 있다. Neura [10] 사의 경우 IoT Cloud 서비스에 대한 호환성을 제공하는 서비스를 제한된 형태로 준비 중이다. IFTTT [11]은 클라우드 서비스 및 IoT 기기의 연동을 사용자 설정을 통하여 제공하는 서비스를 시행 중이다. 다만, 상기의 서비스들은 특정 목적에 제한적으로 적용되고 있어 범용 서비스 연동을 위한 프레임워크로의 적용에는 무리가 있다. 본 논문에서는 현행 IoT 서비스간 연동을 구현할 수 있는 프레임워크를 제시함으로써 기기 및 서비스간의 연동 및 확장에 대한 편의성을 제공한다.

### III. 본 론

현행 IoT 기기 및 서비스간의 호환성을 보장하기 어려운 이유는 다음과 같이 분류 할 수 있다. 첫째, 다양한 IoT 기술의 활용에 대한 국제표준 및 단체 표준의 혼재이다. 국제표준으로는 OneM2M이 현재 표준화가 진행 중에 있으며, 이 외에 각 분야별로 목적에 따라 다양한 IoT 기술을 별도로 표준화하고 있다. AllSeen Alliance [12]는 Peer-to-Peer 기반의 Proximal Network를 기반으로 하는 표준을 제시하고 있으며, 이와 유사한 OIC [13] 표준화도 별도로 진행 중에 있다. 전통적인 홈 네트워크 단체 표준인 UPnP의 경우 클라우드를 통한 IoT 기술을 표방하는 UPnP Cloud Architecture [14]를 2014년 가을에 도출하였다. 이러한 표준들은 각 표준이 목표하는 사용 환경에 따라 독립적으로 구성되고 있다. 둘째, 표준과 무관한 제품 및 서비스의 출시이다. 북미의 기업들을 중심으로 이미 다양한 IoT 단말 및 서비스가 출시되고 있으며, 이러한 서비스들은 현행 표준과 무관하게 독립적인 프레임워크를 적용하고 있다. SmartThings [15]의 경우 자사의 Hub를 통한 Zigbee/ Z-Wave 단말의 상태 확인 및 제어를 지원하고 있으며, WeMo [16]의 경우 스마트 전원 등의 IoT 기술 기반 제어를 지원한다. 또한 WeatherStation [17], Withings [18], Fitbit [19] 등의 다양한 서비스들이 자사의 독립 클라우드 서

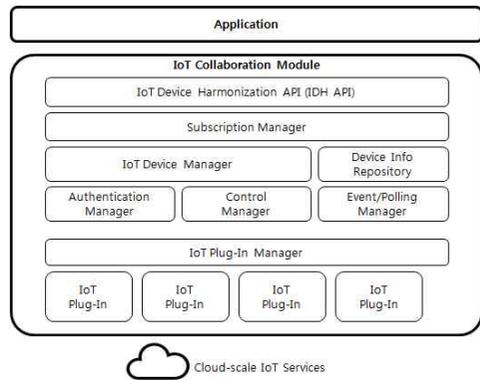


그림 1. IoT Collaboration Framework 구성도  
Fig. 1 IoT Collaboration Framework Overview

비스를 활용하여 사용자에게 다양한 IoT 서비스를 제공하고 있다. 이러한 서비스들은 자사의 서비스 확장을 위하여 제 3자가 서비스 및 어플리케이션을 개발할 수 있는 API를 제공하고 있다. 이러한 API는 Proprietary하게 제공되고 있어 각 서비스별로 독립적으로 구성된다.

본 논문에서는 상기와 같은 문제를 해결하기 위하여, 이중 IoT 단말 및 서비스를 통합 관리할 수 있는 프레임워크와 함께 해당 프레임워크를 통한 사용자 상황 정보 관리 기법을 제시한다. 제시되는 프레임워크는 기존에 제시된 프레임워크 [20]의 개념을 기반으로 현재 시장에 제공되고 있는 각 IoT Service에 대한 통합 관리 기능과 개발 환경을 제공한다.

각 IoT 서비스에 대한 구현은 Plug-In형태로 구성되어 서비스의 확장이 가능하도록 설계되었다. 또한, 각 사별 독립 API에 대한 통합 API를 통하여 개발자에게 단일 API를 통한 이중 IoT 기기 제어 어플리케이션 개발 환경을 제공할 수 있다. 상기의 프레임워크를 통하여 현행 IoT 서비스에 대한 통합 관리 및 제어가 가능해지면, 현행 IoT 서비스로부터 사용자의 상황 정보를 수집한 후 해당 상황 정보에 접근하기 위한 API를 제공함으로써 사용자의 상황 정보에 대한 관리 기능을 제공하게 된다. 시스템의 성능에 격차가 심한 IoT 단말의 특성을 고려하여, 모든 기능은 플랫폼에 독립적인 웹 기반 기술로 설계 및 구현되었다. 본 프레임워크를 통하여 제공되는 장점은 다음과 같다.

1. 개발의 용이성 : 통합 개발환경을 제공함으로써

이중 IoT 단말 및 서비스에 대한 어플리케이션 개발을 단순화할 수 있다.

- 2. 사용의 용이성 : 각 단말 혹은 서비스간 호환성 부재로 인하여 현재의 IoT 서비스는 특정 서비스에 제한된 어플리케이션을 활용하여 해당 서비스에 접근하고 있으나, 본 프레임워크는 단일 어플리케이션을 통한 다중/이중 IoT서비스에 대한 통합 관리를 제공할 수 있다.
- 3. 사용자 상황 정보 관리 : 각 IoT 서비스에서 제공하는 독립된 사용자 상황 정보를 통합 관리할 수 있는 프레임워크를 제공한다.
- 4. 플랫폼 독립성 및 호환성: 상기의 기능들을 웹 표준 기반 기술로 제공함으로써 플랫폼에 대한 개발 제약을 제거하고 서비스/단말간 호환성을 확보한다. 또한 사용자 상황 정보는 국제표준으로 진행 중인 MPEG-UD를 활용하여 호환성을 확보한다.

그림 1은 본 논문에서 제안하는 프레임워크에 대한 구성도이다. 현행 IoT 서비스와 프레임워크의 연동 구성은 그림 2와 같다. 현행 IoT 서비스는 자체적으로 운영되는 Vendor-Specific 클라우드를 통하여 운영되고 있으며, 이러한 서비스들은 Vendor-Specific API를 제공하여 제 3자의 앱 개발을 지원하고 있다. 그림 2의 중간 및 우측 부분은 이러한 Vendor-Specific 서비스를 표현하고 있다. 본 논문에서 제안하는 프레임워크는 좌측부분으로, 각 Vendor-Specific API에 대한 통합 개발환경을 제공함으로써 제 3의 개발자가 HTML5 기반의 단일 API를 통하여 다종의 IoT Cloud에 연결할 수 있는 구조를 가진다.

1. IoT 통합 제어 모듈

IoT 통합 제어 모듈의 구성도는 그림 1에 표시되어 있다. 각 단위 IoT 간의 다양한 구성으로부터 독립성을 보장하기 위하여, 각 단위 IoT 간의 연동은 독립된 IoT Plug-In으로 구현된다. 예를 들어, UPnP Cloud Architecture를 지원하기 위해서는 UPnP Cloud Architecture에 대한 어플리케이션을 구성한 후 해당 어플리케이션을 IoT Plug-In Manager에 해당 Plug-In을 등록하게 된다. Plug-In에 등록되는 기능은 사용자의 접근 권한을 확보하기 위한 Authorization, 특정 자원의 제어를 요청하기 위한 Control, 자원의 상태 변경을 확인하기 위한 Event 기능으로 구분되며, Plug-In의 등록은 해당 기능에 대한 Callback 함수 등록을 통하

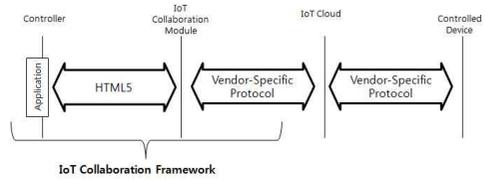


그림 2. 현행 IoT 서비스와의 연동 구성 Fig. 2 Collaboration with IoT Cloud Service

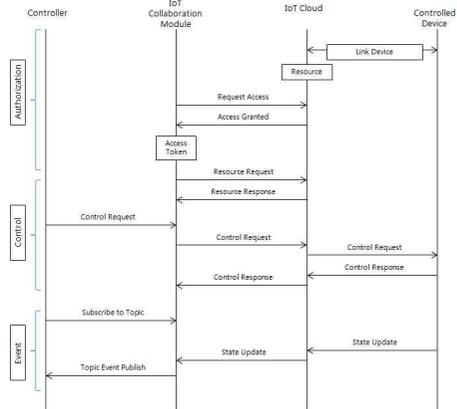


그림 3. IoT Plug-In의 주요 기능 흐름도 Fig. 3 Flow of IoT Plug-In

여 이루어진다. IoT Plug-In Manager는 본 프레임워크에서 지원하는 IoT 서비스들을 관리한다.

Authorization은 각 단위 IoT 서비스에 대한 접근 권한을 요청하고, 그 결과로 반환되는 Access Token을 저장하는 기능을 수행한다. 해당 부분은 각 단위 IoT 서비스에서 정의하는 Authorization 방법을 따라 정의된다. 일반적으로 현재 제공되는 IoT Service들은 OAuth 프로토콜에 의거한 인증방식을 사용하고 있으며, 이 통신은 통합 제어 모듈과 단위 IoT 서비스 내로 국한된다. 즉, 인증 과정에 대한 별도의 외부 API 접근은 제한적이다.

기기의 상태 확인 및 제어의 요청을 위한 Control은 Collaboration API를 통하여 처리된다. Collaboration API는 HTTP 프로토콜을 사용하는 RESTful Architecture로 구성된다. 기기 상태 확인의 경우 상태 정보는 JSON Object로 반환된다.

Event 기능은 IoT 통합 제어 모듈에서 관리하는 기기(자원)의 상태가 변경되었을 경우 변경된 상태를 통보하는 기능을 수행한다. 서버로부터 클라이언트로 전달되어야 하는 특성에 따라 해당 기능은 WebSocket으로 정의된다. 클라이언트는 통합 제어

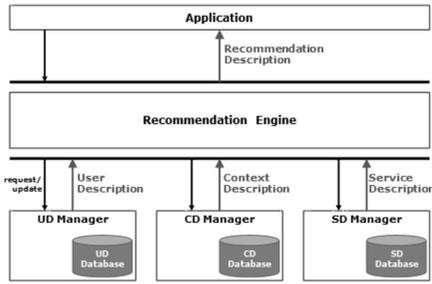


그림 4. MPEG-UD 구성도  
Fig. 4 MPEG-UD Structure

모듈로부터 수신한 자원 중 상태 갱신이 필요한 자원에 대하여 특정 WebSocket 주소에 접근함으로써 자원의 상태 변경 상태를 수신할 수 있다. 상기의 과정은 그림 3에 표시되어 있다.

2. 사용자 상황 정보 수집 및 관리

각 단위 IoT 서비스들은 IoT 자원에 대한 가상화를 통하여 기기의 상태를 표현하고 있다. 해당 자원의 상태는 사용자의 상황 정보를 표시하고 있으며, 본 프레임워크는 해당 자원의 정보를 정형화하고 적절한 사용자 상황 정보에 매핑하여 데이터베이스에 저장한다. 사용자 상황 정보에 대한 호환성을 확보하기 위하여, 본 논문에서 제안하는 프레임워크는 국제 표준화가 진행 중인 MPEG-UD [5]를 통하여 사용자 상황 정보를 기술한다.

MPEG-UD의 개념은 그림 4에 표현되어 있다. MPEG-UD의 목적은 사용자의 상황정보를 고려한 서비스 추천방법을 XML을 통하여 정형화된 방법으로 제공하는 데 있으며, 이를 위하여 사용자의 기본 정보를 정의하는 User Description(UD), 사용자의 상황 정보를 정의하는 Context Description(CD), 사용자가 제공받을 수 있는 서비스를 정의하는 Service Description(SD) 등 3종의 Description을 통하여 사용자의 상황 정보를 구체화한다. 사용자의 상황 정보에 따른 서비스 추천 정보는 Recommendation Description에 따라 기술된다. 본 논문에서 제시되는 프레임워크는 사용자의 기기 정보와 함께 기기로부터 수집되는 상황 정보를 제공할 수 있으며, 해당 정보들은 MPEG-UD의 User Description 중 Device Description 부분과, Context Description 전반에 연관될 수 있다.

수집된 사용자 상황 정보의 접근을 위하여, 제안된 프레임워크는 2가지의 방법을 제안하고 있다. 상

표 1. IoT Plug-In 구현

Table 1. List of Plug-In Implementations

IoT Service	IoT 기기	연관 Context
SmartThings	Contact Sensor	Open/Close
	Power Outlet	ON/OFF
	DoorLock	Lock/Unlock
	Proximity Sensor	Near/ Far
	Camera	Photo
Fitbit	Activity Tracker	Walking Steps
		Walking Distance
		Sleep Time
NetAtmo	WeatherStation	Temperature
		Humidity
		Noise Level
		Co2 Level
WeThings	Smart Scale	Weight
		Body Fat
Hue	Light Bulb/Strip	Brightness
		Hue
		Saturation
Wahoo	Kickr (Bicycle)	Cycling Distance
		Cycling Speed

황정보에 접근하기 위해서는 REST API에 접근함으로써 해당 자원의 상태를 확인할 수 있다. 이는 HTTP GET 등의 프로토콜 메시지를 통하여 수행될 수 있다. 또한 상태의 변경정보를 실시간으로 확인하기 위해서, 제안된 프레임워크는 MQTT 프로토콜의 Publish-Subscribe기술을 사용한다. 각각의 상태 정보는 하나의 Topic으로 구성되어 데이터베이스에 저장된다. 해당 상태 정보(Topic)를 모니터링하려는 객체는 MQTT 프로토콜을 사용하여 해당 Topic에 Subscribe를 진행하고, 상태정보의 변경은 해당 Topic에 대한 MQTT Publish를 통하여 처리된다. MQTT 프로토콜은 Application 프로토콜로써, HTTP 프로토콜과의 호환성을 확보하기 위해 제안된 프레임워크에서는 WebSocket을 통한 EVENT형태의 구성으로 정의한다.

IV. 구현 및 검증

제안된 프레임워크의 타당성을 검증하기 위하여, 본 논문에서는 IoT 기기 및 서비스에 대한 테스트베드를 구축하고 해당 테스트베드에의 프레임워크 실장 및 프레임워크 기반 어플리케이션의 개발을 진행하였다. 실제 사용환경에서의 프레임워크 동작성을 검증하기 위하여 테스트베드에 구축된 IoT 단말 및 서비스는 시뮬레이터가 아닌 실제 시판 중인 IoT 단말과 서비스를 대상으로 하였다. 테스트베드



그림 5. IoT Test bed 구성  
Fig. 5 IoT Test bed construction

<pre> idx : int(11) mem_idx : int(11) device_id : varchar(30) device_name : varchar(15) state : varchar(5) distance : varchar(10) totaltime : varchar(10) device_state : text device_log : text reg_date : timestamp         </pre>	<pre> idx : int(11) mem_idx : int(11) app_id : varchar(75) app_secret : varchar(75) access_token : varchar(75) access_secret : varchar(75) reg_date : timestamp         </pre>	<pre> idx : int(11) mem_idx : int(11) bridge_id : varchar(60) bridge_ip : varchar(30) reg_date : timestamp         </pre>
<pre> idx : int(11) mem_idx : int(11) app_key : varchar(65) app_secret : varchar(65) access_token : varchar(65) access_secret : varchar(65) userid : varchar(15) reg_date : timestamp         </pre>	<pre> idx : int(11) mem_idx : int(11) bridge_ip : varchar(30) lights_id : int(11) module_state : text location : varchar(25) reg_date : timestamp         </pre>	<pre> idx : int(11) mem_idx : int(11) app_id : varchar(65) access_token : varchar(65) refresh_token : varchar(65) reg_date : timestamp         </pre>
<pre> idx : int(11) mem_idx : int(11) module_type : varchar(15) getuser : text getuser : text location : varchar(25) reg_date : timestamp         </pre>	<pre> idx : int(11) mem_idx : int(11) module_name : varchar(10) current_data : text week_data : text all_data : text getuser : text module_id : varchar(65) location : varchar(25) reg_date : timestamp         </pre>	<pre> idx : int(11) mem_idx : int(11) module_version : varchar(30) module_id : varchar(30) module_type : varchar(30) getuser : text calori_data : text steps_data : text distance_data : text activation_date : text floors_data : text location : varchar(25) reg_date : timestamp         </pre>

그림 6. 사용자 상황정보 데이터베이스 구조  
Fig. 6 Structure of context database table

에 구축된 IoT 단말 및 서비스의 종류는 표 1과 같다. 그림 5는 상기의 단말을 통하여 구축된 테스트 베드를 보여주고 있다.

1. IoT 통합 제어 모듈의 구현

프레임워크는 Linux 기반의 시스템의 웹 서비스로 구현되었다. 웹 서비스는 Apache와 MySQL의 연동으로 구성되었으며, 웹 서비스 어플리케이션은 CodeIgniter를 활용한 PHP로 개발되었다. Code Igniter의 MVC 구성을 기반으로 총 6종의 Plug-In이 개발되었다. 각 Plug-In에 대한 접근은 3장에 기술된 바와 같이 Collaboration API로 이루어진다. 각 자원에 대한 접근은 RESTful로 정의된 특정 URI에 HTTP 프로토콜로 구현된다. 자원에 대한 URI는 다음의 구조를 적용한다.

http://[서버주소]/[사용자 ID]/[디바이스 ID]

어플리케이션은 해당 주소에 HTTP GET 메시지를 송신함으로써 해당 기기의 상태를 확인할 수 있다. 해당 기기의 상태는 MPEG-UD 표준에 의거한 JSON Object로 반환된다. 해당 기기의 제어

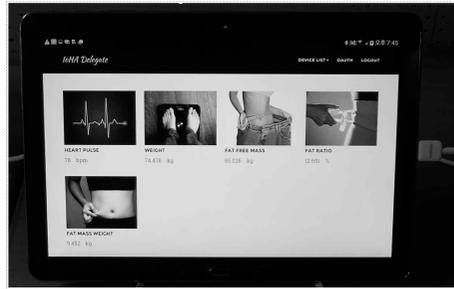


그림 7. 사용자 상황정보 사용자 인터페이스  
Fig. 7 UI for user context management



그림 8. 사용자 상황 정보 기반 어플리케이션  
Fig. 8 User context based application

위해서는 HTTP POST 메시지를 통하여 기기의 제어를 요청한다.

기기의 상태 변경을 수신하기 위해서는 3장에 제시된 바와 같이 MQTT와 WebSocket을 사용한다. 특정 자원의 WebSocket URI는 다음과 같다.

http://[Resource URI]/websocket

자원의 상태가 변경될 경우 해당 WebSocket을 통한 JSON Object가 전송되며, JSON Object의 구성은 상기의 확인 메시지와 동일한 구성을 따른다.

2. 사용자 상황 정보 수집 모듈 및 어플리케이션의 구현

상기에서 제시된 프레임워크를 통하여, 각 IoT 서비스로부터 사용자 상황 정보를 지속적으로 수집할 수 있다. 수집된 사용자 상황 정보는 프레임워크 내의 데이터베이스에 저장된다. 그림 6은 사용자 상황 정보 수집 모듈의 데이터베이스 구조를 표시하고 있다. 각 테이블명에서 해당 데이터베이스와 연관된 IoT 서비스를 확인할 수 있다.

상황 정보의 수집은 3개월간 개인 사용자가 표 1에 나열된 기기를 사용한 상황 정보를 대상으로

이루어졌다. 각 상황 정보는 현행 IoT 서비스의 클라우드로부터 주기적으로 갱신되었으며, IoT Delegate에 저장된 데이터는 서비스 제공자의 IoT Cloud에 저장된 값과 비교하여 그 무결성을 검증하였다.

수집된 상황 정보는 프레임워크에서 제시한 Resource URI를 통하여 접근 가능하며, 해당 자원의 상태 변경 갱신을 위해서는 동일하게 Resource WebSocket URI를 사용할 수 있다. 각 자원의 상태는 MQTT의 Topic에 매핑되어 있으며, 해당 Topic에 대한 PubSub은 WebSocket을 사용한다. 해당 API를 통한 어플리케이션이 그림 7과 8에 각각 표현되어 있다. 스마트TV 및 태블릿 상에 구현된 어플리케이션을 통하여 수집된 사용자 상황 정보가 표시된다. 그림 7은 태블릿 기반의 어플리케이션으로 수집된 사용자 상황 정보를 표현하고 있다. 그림 8은 SmartTV 기반의 어플리케이션으로, 각 IoT Cloud로부터 수집된 정보가 우측 4개의 박스에 표현되고 있으며, 왼쪽의 원형은 Wahoo Kickr로부터 수집되는 실시간 운동 정보를 표현한다.

## V. 결 론

본 논문에서는 사용자 상황 정보 관리를 지원하기 위한 IoT 통합 제어 모듈이 제안되었다. 제안된 기술은 이종 IoT 기술의 제어 및 연동이 가능한 프레임워크를 기반으로 개발이 용이한 어플리케이션 개발 환경을 제공한다. 제공되는 API를 통하여 개발자는 특정 IoT 플랫폼이 아닌 범용의 어플리케이션 개발을 보다 손쉽게 진행할 수 있으며, 기기간 연동을 통하여 사용자에게 보다 풍성한 사용 경험을 제공할 수 있다.

제시된 프레임워크는 실제 IoT 제품을 통하여 컨셉수준을 상회하는 구현으로 검증되었다. 총 6종의 현행 IoT Cloud와 10종의 기기를 대상으로 기기 및 서비스에 대한 Plug-In이 구현되었으며, 이를 통하여 기기간 연동 시나리오를 검증하였다. 또한, 수집되는 사용자 상황 정보는 향후 서비스의 개발을 위한 Big Data 자료로 활용이 가능하며 사용자 상황 정보에의 체계적 접근 및 반응성 확보를 위한 WebSocket Interface가 함께 구현되었다. SmartTV 및 태블릿용 어플리케이션 개발 및 기능 동작을 통하여 다종 IoT 서비스/기기를 제어하기 위한 효율성을 확인할 수 있다.

본 논문에서 제시된 프레임워크 및 구현 예는

향후 각 단위 IoT 서비스간의 연동 및 통합 제어를 위한 기반 기술로의 활용이 가능하다. 향후에는 본 프레임워크를 통하여 수집된 사용자 상황 정보를 활용한 Big Data Application의 개발과 함께 임베디드 시스템에서의 MQTT 프로토콜 기반 구현이 예정되어 있다. 또한 대량의 데이터를 보다 효율적으로 처리하기 위한 MQTT Topic 구조의 정의 및 효율 검증이 진행될 예정이다.

## References

- [1] L. Atzori, A. Iera, G. Morabito, "From 'smart objects' to 'social objects': The next evolutionary step of the internet of things," *IEEE Communications Magazine*, Vol. 52, No. 1, pp. 97 - 105, 2014.
- [2] J. Moon, T.B. Lim, K.W. Kim, S.W. Kum, M. Yoon, K. Cho, "A Framework for Dynamic Context-Awareable User Interfaces of the Second Screen Devices in a Home Network Environment," *Proceedings of IEEE Annual Conference on Computer Software and Applications*, pp. 284 - 285, 2013.
- [3] J. Son, J.-H. Park, K.-D. Moon, Y. Lee, "Resource-aware smart home management system by constructing resource relation graph.," *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 3, pp. 1112 - 1119, 2011.
- [4] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services," *IEEE Transactions on Services Computing*, Vol. 3, No. 3, pp. 223 - 235, 2010.
- [5] S. Nastic, S. Sehic, M. Vögler, H.-L. Truong, S. Dustdar, "PatRICIA-A Novel Programming Model for IoT Applications on Cloud Platforms," *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications*, pp. 53-60, 2013.
- [6] J.-H. Kim, C.-W. Lee, "Design and Implement a Gateway Based on Mobile Device and a Web Monitoring System for u-Healthcare

- Service,” Journal of Embedded Systems and Applications, Vol. 4, No. 3, pp. 126-133, 2009 (in Korean).
- [7] R. Bosman, J. Lukkien, and R. Verhoeven, “Gateway architectures for service oriented application-level gateways,” IEEE Transactions on Consumer Electronics, Vol. 57, No. 2, pp. 453 - 461, 2011.
- [8] S. Cirani, L. Davoli, G. Ferrari, R. Leone, P. Medagliani, M. Picone, L. Veltri, “A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things,” IEEE Internet Things Journal, Vol. 1, No. 5, pp. 508-521, 2014.
- [9] “ISO/IEC JTC1/SC29/WG11/N13879 Call for Proposals on MPEG User Description”, ISO Publication, Aug 2013.
- [10] <http://www.theneura.com>.
- [11] <http://www.ifttt.com>.
- [12] AllSeen Alliance, <http://allseenalliance.org>.
- [13] OIC - Open Interconnect Consortium, <http://openinterconnect.org>.
- [14] UPnP Forum, “UPnP Device Architecture 2.0,” Sep. 2014.
- [15] <http://www.smartthings.com>.
- [16] <http://www.wemo.com>.
- [17] <http://www.netatmo.com>.
- [18] <http://www.withings.com>.
- [19] <http://www.fitbit.com>.
- [20] S.W. Kum, J.W. Moon, K.W. Yuk, T. Lim, Jong I. Park, “A Novel Design of IoT Cloud Delegate Framework to Harmonize Cloud-Scale IoT Services,,” Proceeding of International Conference on Consumer Electronics, pp. 247-248, 2015.

### Seung Woo Kum (금 승 우)



He is a Ph.D. candidate in the Department of Electrical and Computer Engineering, Hanyang University. From 2002 to 2006, He was a senior research engineer LG Electronics, where he had worked on home network applications on DTV. He joined Smart Media Research Center of Korea Electronics Technology Institute in 2006. His research interests include home network and IoT, smart home service and related application on embedded system.

Email: [swkum@keti.re.kr](mailto:swkum@keti.re.kr)

### Tae Beom Lim (임 태 범)



He received Ph.D degree in Computer Science and Engineering in Konkuk University in 2012. From 1997 to 2002, he was with Daewoo Electronics, Seoul, Korea. He joined Korea Electronics in 2002, where he is currently a director of Smart Media Research Center. His research interests include digital broadcasting solutions, personalized digital TV solutions, embedded systems, and operating systems.

Email: [tblim@keti.re.kr](mailto:tblim@keti.re.kr)

**Jong Il Park (박종일)**

He received Ph.D. degrees in Electronics Engineering from Seoul National University, Seoul, Korea, in 1995. From 1996 to 1999, he was with ATR Media Integration and Communication

Research Laboratories, Japan. He joined the department of Computer and Software, Hanyang University, Seoul, Korea, in 1999, where he is currently a Professor. His research interests include computational imaging, augmented reality, 3D computer vision, and HCI.

Email: [jipark@hanyang.ac.kr](mailto:jipark@hanyang.ac.kr)