

IoT Delegate: Smart Home Framework for Heterogeneous IoT Service Collaboration

Seung Woo Kum^{1,3}, Mingoo Kang² and Jong-Il Park¹

¹ Department of Computer and Software, Hanyang University
Seoul, South Korea

² Dept. of IT Contents, Hanshin University, Hanshin University
South Korea

³ Smart Media Research Center, Korea Electronics Technology Institute
Sunnam, South Korea

[e-mail: swkum@keti.re.kr, kangmg@hs.ac.kr, jipark@hanyang.ac.kr]

*Corresponding author: Seung Woo Kum

*Received May 9, 2016; revised July 1, 2016; accepted August 1, 2016;
published August 31, 2016*

Abstract

With Internet of Things (IoT) technology, home environment becomes smarter than ever. Not only smart devices such as smart phone or smart TV, but also various IoT devices including sensor, smart thermostat, and smart scale has now become very common on the market. These devices have connectivity to the Internet, so that user can read data from the device or control the device using Internet technology. However, due to diversity of smart home requirements, device collaboration in smart home remains a challenging task still. Usually smart home is built with various technologies to fulfill its own purpose, and these purposes cover very wide area from controlling low-power sensor devices to controlling high-performance devices like smart TV and smart phone. This variety of smart home requirements makes smart home very complicated due to mixed network architecture, protocol and technology. In this paper, a framework to enable managing and collaborating heterogeneous IoT devices in smart home environment is proposed. Several programming models are defined in the proposed framework to make application development for heterogeneous devices more intuitive. The proposed framework has been implemented as a web service, and a case study with real-world smart home IoT devices is presented.

Keywords: Heterogeneous Device Network, Application-level Collaboration, Programming Model.

A preliminary version of this paper appeared in ICONI 2015, December 13-16, Kuala Lumpur, Malaysia, and was selected as an outstanding paper. This version includes the IoT Delegate framework for horizontal IoT service and expands implementation result to IoT device interaction service. This research was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (R0101-16-0159, "Development of Responsive RRUI Technology based on Device/ Environmental/ Emotional/ Cognitive Information for Information Technology").

1. Introduction

With the advance of Internet of Things (IoT) technology, connectivity has been added to number of home sensors and devices. It's not just smart TV, smart SetTop-Box or smart phones. Sensor devices such as proximity sensor, IR sensor or open/close sensor have low-power Zigbee or Z-Wave connection, but can access the Internet with the help from smart home gateways. For example, door locks have Z-Wave connectivity to let user know its status and control it. It becomes quite common experience to have notification on user smart phone when motion is detected in motion sensor in your living room. User can read temperature, humidity, CO2 level, noise level and much more data from IoT-enabled sensor to his/her smart phone. Also, conventional home appliances such as refrigerator, washing machine and microwave oven have IP-based connectivity to guide user for better experiences. Refrigerator can show its contents in it by list or the camera installed, microwave oven can download recipes from the Internet, coffee maker, heater and air conditioner can be set with smart phone. It's not just control, multi media contents can be shared between audio video home appliances, and with mirroring each screen can be transmitted to the other device in real-time. This connectivity in Things in home makes home much smarter, and provides rich experience to home members.

Smart home is quite a complex environment since lots of different use cases co-exist for each of family members. Smart home is not easy to be defined as a single architecture due to different requirements from various use cases. For example, let's take a look on sharing multimedia contents between audio video appliances. In this case, peer-to-peer ad-hoc network between two smart home devices seems enough, since user is not likely try to send his screen of office PC to a TV in his living room. Also, transmitting contents is confined between neighboring devices, so no Internet connection is required. For Device control use case, requirements become quite different. User may want to know whether the door lock is closed or not in his office, or any unexpected motion is detected in his living room. In this case, the smart home device needs to be connected to the Internet so that user can read data from the device or set the device remotely. So, server-client network and server in the Internet is required. There are requirements other than network architecture for sensors. Sensors shall be small, and its power consumption shall be as low as possible. To meet these requirements, IoT sensors use protocols for low power consumption like Zigbee, Z-Wave or Bluetooth Low Energy instead of IP network. Since these protocols are not based on IP, a gateway device is needed for Internet connection. These are only a few of practical use case in smart home, and these few examples are using all different network architecture and different protocols. This heterogeneous nature of smart home makes device interaction difficult. Some of smart home device manufactures provide their own vendor-specific Open APIs to resolve this problem. But since they are vendor-specific based on independent architecture, building application for collaboration is quite complex task.

To address these issues in smart home IoT environment, IoT Delegate: a framework for heterogeneous Home IoT device collaboration is proposed. The proposed framework aims to provide an ecosystem for smart home application development and provisioning of smart home IoT appliances. To this purpose, two programming models are defined in the proposed framework. Smart Home IoT Programming Model is defined to deliver high-level programming architecture, which encapsulates device-specific knowledge from application development. IoT Device Harmonization Programming Model, which provides methods for

interaction between IoT devices and/or services. Combined with these models, the proposed framework enables easier, faster and intuitive development of heterogeneous home IoT applications.

This paper is organized into sections as follows: Related researches and works are presented in Section 2. In Section 3, characteristics of various home IoT architectures are investigated, followed by overview of IoT Delegate framework. Section 4 describes Smart Home IoT Programming Model and IoT Device Harmonization Programming Model. Section 5 describes the implementation of a service system and applications based on proposed framework. Conclusion remarks are presented in Section 6.

2. Related Works

Recent approach about IoT application development includes researches about data and device integration based on Service Oriented Architecture and WS-BPEL. Guinard et al. [1] proposed a system architecture to enable communication enterprise applications and physical devices. Pintus et al. [2] employed WSDL to abstract Web Services into Things. There are several researches about implementing IoT Services [3] and providing collaboration between devices [4][5][6]. PatRICIA [7] used Intent and IntentScope programming model. These approaches aim to abstract devices and services into Things and enable collaboration between Things in a service.

Various protocols are being employed in IoT [8], and there are researches to harmonize heterogeneous network devices. Brut et al. [9] proposed an infrastructure to provide device interoperability using protocol bridge and gateway device. Bosman et al. proposed gateway architecture for interoperability between SOA and UPnP. Jin et al. [10] proposed a framework to build Smart City using IoT sensors. Cirani et al. [11] proposed IoT Gateway-based architecture for Service Discovery. These approaches employ a specific gateway device to provide interoperability. There are also researches about IoT based on specific network protocol such as XMPP [12] or RESTful [13]. Most recent research on IoT includes finding context-awareness in home network and relationship analysis between activities. Son et al. proposed a system describing resource relation between smart home devices [14]. DU et al. [15] uses user emotion for device collaboration. Jin et al. [16] suggested a method to detect spatio-temporal relationship from a Pub-Sub system. Machine learning algorithms are adopted to find the relationship [17]. These works proposed architectures for harmonization of heterogeneous network in protocol level or middleware level within a single architecture. The proposed framework defines service-level architecture, which enables inter-service application development without prior knowledge about specific protocols.

3. IoT Delegate

3.1 Smart Home IoT Models

Smart home is a mixture of independent technologies (in aspect of architecture and protocol) to accommodate various use cases in home network. To manage multiple network architecture within smart home efficiently, first it needs to classify various network architectures used in home network with respect to its characteristics. Among many characteristics, network topology, network medium, popular use case are considered for the classification, and it covers current IoT services and devices available in market as well as IoT standards.

Peer-to-Peer Model: Peer-to-peer direct connection between two end devices: one is controller, and the other is controlled device. This model provides proximal connectivity. For IP-based devices, UDP-based protocols like SSDP or mDNS are used to create ad-hoc network between devices. Application resides in end device. UPnP MediaServer and MediaRenderer are classified into this model. Also, Bluetooth connection between smart phone and Bluetooth speaker fits this model. All data are stored in the controller. Since it is an ad-hoc network, data stored in controller may not be permanent.

Server-Client Model: Two end devices (controller and controlled device) communicate with server. No direct connection between end devices. User identification and pairing with device are processed on the server side. User data and device data are stored on the server also. Specific APIs and secure delegated access methods are provided for third party developer to access the server.

IP Gateway Model: Gateway is added between server and controlled device in Server-Client Model.

Non-IP Gateway Model: Gateway is added between server and controlled device in Server-Client Model. In this model, controlled device uses non-IP protocol such as Bluetooth and ZigBee.

The last three models are based on server-client architecture, and to process large number of customers and their device information, implemented on cloud architecture. In this paper, these models are referred as cloud-scale IoT services. To our survey, all the existing smart home IoT device or service can be categorized within these models.

According to these classifications, the major requirements for our framework include:

- 1) Providing a programming model to raise the level of abstraction by decoupling device knowledge implementation from its usage,
- 2) Providing a programming model to collaborate heterogeneous smart home IoT devices,
- 3) Providing better performance for application with multiple IoT services.

3.2 Overview of IoT Delegate

Fig. 1 shows structure of the proposed framework. IoT Delegate consists of 3 major blocks: IoT Plug-In Manager Layer, Service Management Layer and Service Collaboration Layer. The main purpose of IoT Delegate is to provide access to heterogeneous IoT devices regardless of its architecture or protocol and this functionality is provided with IoT Plug-In. Each IoT Plug-In is an abstraction of corresponding IoT services. This abstraction removes dependency of specific architecture or protocol of IoT services. From the viewpoint of implementation, IoT Plug-In is an application for corresponding IoT service wrapped by a common API set. For example, if IoT Plug-In A is for peer-to-peer Bluetooth connection with wearable smart band, Bluetooth protocol is going to be implemented in IoT Plug-In A, and the functionality of wearable smart band is presented with common API set. IoT Plug-In Manager layer processes requests and responses to IoT service entity. It has multiple IoT Plug-Ins which are implementations of IoT device communication.

Service Management Layer generates DataChannels which are abstraction entities of device resource. Device management (control and monitoring) is processed with DataChannel. Writing desired value in DataChannel will control device resource, or reading values on DataChannel will return updated value from device resource. Type of DataChannel is URI, and uses HTML messages with JSON-formatted data in it. Methods to manipulating DataChannel are defined with IoT Device Harmonization API (IDH API) [18].

Service Collaboration Layer uses DataChannel to create collaboration of heterogeneous devices. Collaboration between device resources is defined as Intention. Intention along with

DataChannel encapsulates device information and decouples all device/service specific information from application development. This layer provides Intention API [18] for application development using Intention.

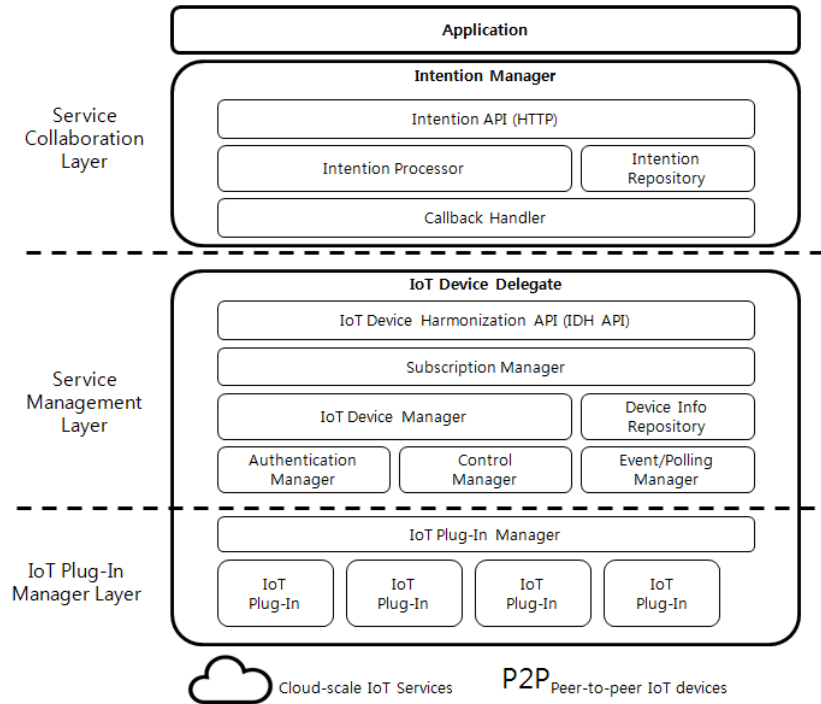


Fig. 1. High-level architecture of IoT Delegate

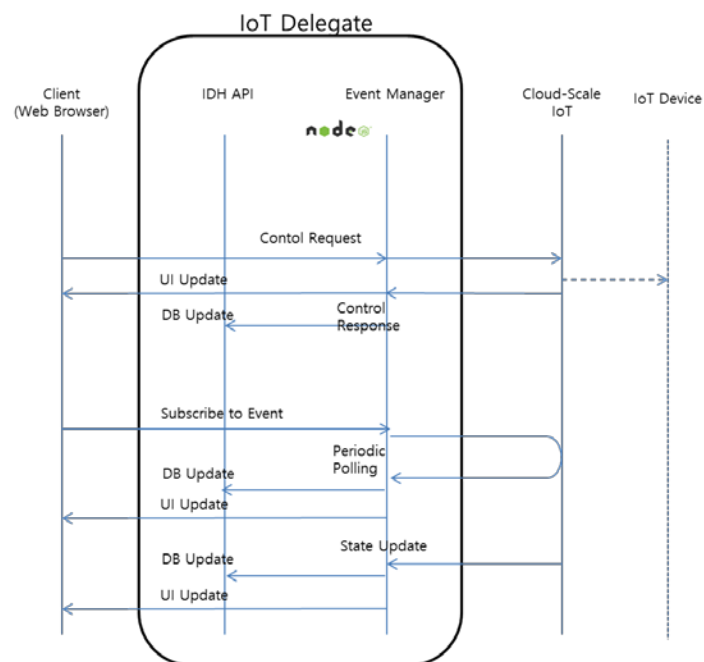


Fig. 2. Message flow between IoT Delegate, cloud-scale IoT Service and service application

Fig. 2 shows how IoT Delegate framework communicates with other cloud-scale IoT services. The proposed framework locates between an application and multiple IoT services, and works as a proxy or gateway. IoT Delegate gathers IoT device state from IoT service or IoT device itself, and save them to a database. This flow can bring faster response time for IoT service application. Since IoT delegate holds data from IoT devices, the application needs to access IoT delegate only, instead of multiple requests to each IoT services.

4. PROGRAMMING MODELS IN IOT DELEGATE

There are two programming models in IoT Delegate. The first one is called Smart Home IoT Programming Model which generates DataChannel to manage device resources. The other one is IoT Device Harmonization Programming Model, and it provides Intention API for device collaboration.

4.1 Smart Home IoT Models

As mentioned above, there are many architectures for IoT, and they use vendor-specific architecture with vendor-specific APIs. The concept of IDH API is to provide a common set API to harmonize heterogeneous devices / services. Device specific APIs vary from vendor to vendor, but there are several common tasks which can be grouped in the same category. IoT Device Harmonization API use 3 categories to build common API set: Authorization, Control and Event. Authorization API handles authorization as its name implies, and also delegation of service access. Existing IoT services uses Internet protocols such as OAuth to delegate resource access to an application securely. After successful delegation, resource can be accessed with various API formats such as JSON object on RESTful architecture or SOAP message over HTTP. This is done with Control API. Control API provides unique URI and data format to control each devices. If device state changes, it can be detected by long polling, event subscription or GENA protocol. This is done by Event API.

These modules enable accessing resource of a device. Resource-related data is stored in database of IoT Delegate. **Fig. 3** shows example database structure of the proposed framework. The boxes in the first row show data structure of Authorization Module. Vendor-specific data for authorization like access_token, bridge_id are located in those tables. The Boxes in the second row show data structure for resource state. Data read from device resource are stored in those tables.

iot_delegate.withings idx : int(11) mem_idx : int(11) app_key : varchar(70) app_secret : varchar(70) access_token : varchar(70) access_secret : varchar(70) userid : varchar(20) reg_date : timestamp	iot_delegate.smarthings idx : int(11) mem_idx : int(11) api_key : varchar(70) api_secret : varchar(70) access_token : varchar(70) access_url : varchar(70) reg_date : timestamp	iot_delegate.weatherstation idx : int(11) mem_idx : int(11) app_id : varchar(70) app_secret : varchar(70) access_token : varchar(70) refresh_token : varchar(70) reg_date : timestamp	iot_delegate.hue idx : int(11) mem_idx : int(11) app_id : varchar(60) bridge_id : varchar(30) bridge_ip : varchar(30) reg_date : timestamp	iot_delegate.fitbit idx : int(11) mem_idx : int(11) app_id : varchar(70) app_secret : varchar(70) access_token : varchar(70) access_secret : varchar(70) reg_date : timestamp
iot_delegate.module_withings idx : int(11) mem_idx : int(11) module_name : varchar(40) getmeas : text reg_date : timestamp	iot_delegate.module_smarthings idx : int(11) mem_idx : int(11) module_id : varchar(50) module_label : varchar(50) module_type : varchar(50) module_value : varchar(50) reg_date : timestamp	iot_delegate.module_weatherstation idx : int(11) mem_idx : int(11) module_name : varchar(30) module_type : varchar(40) dashboard_data : text all_data : text module_id : varchar(70) reg_date : timestamp	iot_delegate.module_hue idx : int(11) mem_idx : int(11) bridge_id : varchar(40) bridge_ip : varchar(40) lights_id : varchar(30) user_name : varchar(40) module_state : text reg_date : timestamp	iot_delegate.module_fitbit idx : int(11) mem_idx : int(11) module_version : varchar(35) module_id : varchar(35) calori_data : text steps_data : text distance_data : text activescore_data : text floors_data : text reg_date : timestamp

Fig. 3. IoT Plug-In data structure.

Device status or data from a device can be accessed via DataChannel entity. DataChannel is representation of device resource in application level. By writing to and reading from the DataChannel, application can control and access specific device resource. To provide device-independent of Smart Home IoT Programming Model, RESTful architecture is used instead of defining methods. Specific URI is defined by combination of the resource name, type and identification for a DataChannel with HTTP commands.

4.2 IoT Device Harmonization Programming Model

The IoT Device Harmonization Programming Model provides framework to setup device collaboration between smart home devices regardless of which IoT technology it's based on. This collaboration is defined as Intention. Intention consists of two parts: Trigger and Action. Intention interprets as 'When Trigger occurs, process this Action'. For example, 'When indoor temperature is higher than 26 degrees Celsius, turn on the air conditioner in COOL mode'. In this example, 'When indoor temperate is higher than 26 degrees Celsius' is Trigger, and 'turn on the air conditioner in COOL mode' is Action. Multiple Actions can reside in one Intention, and Intention can be nested.

IoT Device Delegate provides service abstraction APIs for services. These APIs are called IoT Device Harmonization (IDH) API. IDH API is defined in RESTful architecture. To set up an Intention, application setup Intention in XML format and sends it to the Intention Manager. When Intention Manager receives Intention, it parses DataChannel parameter in Trigger, and subscribe to corresponding data source. And then, it registers callback function to invoke actions described in Action parameter. Multiple Actions for a single Trigger is available. Pseudo code for application using Intention is given in [Fig. 4](#).

```
Function parseIntention (Intention intention)
{
  // Parse Intention
  Trigger trigger = Intention.getTrigger();
  Action[] action = Intention.getAction();

  // Set Data Channel
  DataChannel dchannel = new DataChannel();
  dchannel.setSource( trigger.getDataShource());
  dchannel.setCondition( trigger.getCondition());

  // Set Action(s)
  Action 1stAction = new Action();
  1stAction.setTargetDevice(action[0].getTarget());
  1stAction.setAction(action[0].getDesiredAction());

  // Set Callback on DataChannel
  dchannel.suscribe( Action.getTarget(),1stAction);
}
```

Fig. 4. Sample Psuedo Code of IDH API application

5. Implementation and Evaluation

The aim of IoT Delegate framework is to provide a framework for collaboration between heterogeneous devices for smart home. To evaluate this interaction, a specific service scenario is required. Among many service scenarios, health care service on smart devices was chosen. The proposed framework is for real-world devices rather than conceptual ones. So instead of using simulator or proof-of-concept device, all the implementation is done with devices available on the market.

To verify collaboration based on DataChannels and Intentions, a test bed has been setup. **Fig. 5** and **Fig. 6** shows configuration of test bed. Testbed consists of devices from 4 cloud-scale IoT services and 2 proximity-based IoT services, with one server and two clients (smart TV and smart tablet). Detailed implementation of IoT Delegate and Health applications is described here.



Fig. 5. Test bed setup for IoT Delegate Evaluation

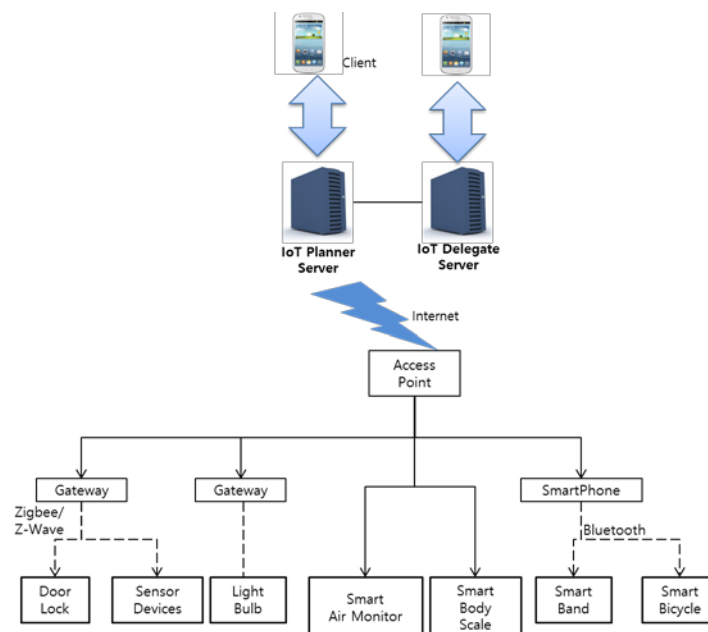


Fig. 6. Test-bed configuration with real-world devices

5.1 Implementation of IoT Delegate

IoT Delegate has been implemented as a Server Application with PHP script language and node.js. For the IoT Plug-in, total six IoT Plug-ins have been implemented. Four of them are cloud-scale IoT services and the others are peer-to-peer services. These IoT Plug-ins are implemented using vendor-specific APIs corresponding to each related service, and provided common API set for Authorization, Control and Event. Characteristics of each IoT Plug-ins are shown in [Table 1](#).

To receive data from each IoT Plug-in, 23 DataChannels are implemented with PHP. [Table 2](#) shows implemented DataChannels from each device in IoT Delegate. Each IoT Plug-in communicates with corresponding device, and all the data read from device are saved in device information repository in IoT Delegate. DataChannel in IoT Delegate is implemented in RESTful architecture with designated URI. Each data channel has specific URI for data submission and retrieval. Data can be updated and read by HTTP PUT and HTTP GET message to designated URI respectively. To receive update from DataChannel, a bi-directional Web Socket is implemented with node.js. Web Socket URI is composed of DataChannel URI appended by “/websocket” string.

User interface for IoT Delegates are designed to display device resource data and control them from/to various DataChannels. All user interfaces are built with HTML5, so that any web browser or hybrid web-app can be used to access IoT Delegate. [Fig. 7](#) shows UI on smart tablet.

Table 1. Details of IoT Plug-In Implementation

Vendor	Underlying Network	Model	Device/ Functions
A	Bluetooth	Gateway	Smart Bicycle
B	IP	Server-Client	Smart Weather Sensor
C	Bluetooth	Gateway	Smart Band
D	IP	Server-Client	Smart Body Scale
E	IP, Zigbee	Gateway	Smart Light
F	IP, Zigbee, Z-Wave	Gateway	Proximity Sensor Open/Close Sensor Motion Sensor Smart Power Door Lock

Table 2. Details of IoT Plug-In DataChannel Implementation

Type	DataChannel	Value
Light	POWER	ON/OFF
	Hue	0~255
	Saturation,	0~255
	Brightness	0~255
Door Lock	LockStatus	LOCK/UNLOCK
Proximity Sensor	Proximity	NEAR/FAR
Open/Close Sensor	Open/Close	OPEN/CLOUSE
Motion Sensor	Motion	0/1
Smart Body Scale	Weight	Integer
	BPM	Integer
	BFM	Integer
Smart Band	WalkingSteps	Integer
	WalkingDistance	Integer

Smart Bicycle	CyclingSpeed	Integer
	CyclingDistance	Integer
Smart Air Conditioner	Fan Speed	1~5
	Mode	COOL/WIND
	POWER	ON/OFF
Smart Power	POWER	ON/OFF
Smart Weather Sensor	Temperature	Integer
	Humidity	Integer
	CO2 level	Integer
	Noise level	Integer

5.2 Health Application with IDH

To evaluate IDH programming model, a web service applications with Intentions has been implemented: HealthCue. This application is built as a Web application on cloud service, with DataChannel and Intention APIs. HealthCue application provides access to 3 clouds-scale IoT services (NetAtmo, Withings and Fitbit), one proximity IoT service (Wahoo Bicycle Trainer) simultaneously. HealthCue application displays data from each cloud-scale IoT services such as temperature, weight and step count via DataChannel API, and displays user activity data from Wahoo Bicycle Trainer in real-time on a single screen.

Implementation of HealthCue is shown on [Fig. 8](#). On the right side of [Fig. 8](#), there are 2x2 blocks each indicating data read from corresponding DataChannels. On the left, there is a circle which indicates Speed DataChannel read from Smart Bicycle device in real-time. Intention of HealthCue is defined as follows:

Trigger: Speed DataChannel changes value 0 from non-zero.

Action1: Turn on Smart air conditioner with respect to temperature value from Temperature DataChannel of Smart Weather Sensor device.

Action2: Turn on Light device.

Action3: Play music on controller device.

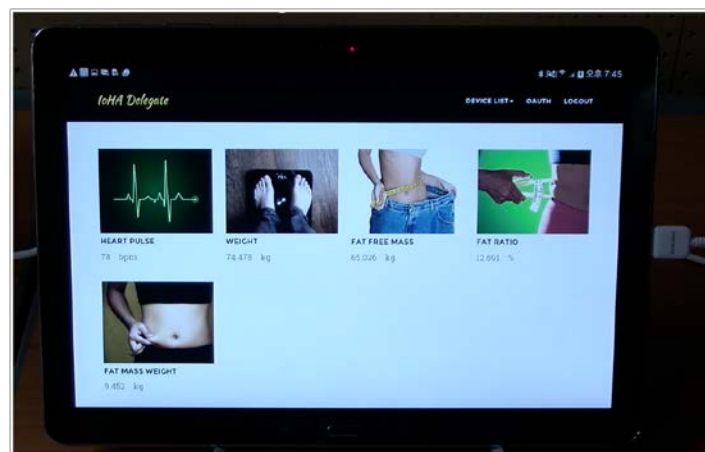


Fig. 7. Horizontal IoT Delegate Application on smart tablet: Data values from various DataChannels are displayed and updated.



Fig. 8. HealthCue Application on smart TV

5.3 Performance Evaluation of IoT Delegate Service

To evaluate performance of IoT Delegate, round-trip delay of IoT service request is measured. IoT Delegate works as a gateway or proxy for individual IoT services, and it can be assumed that at least one hop has been added comparing to normal cloud-scale IoT services. However, IoT Delegate stores data from individual cloud-scale IoT services and this can save significant amount of round-trip delay. Instead of accessing individual cloud-scale IoT service sequentially, only one request is required with IoT Delegate. The performance becomes even better if a service uses more cloud-scale IoT services. This is because most of HTML5 documents are loaded into memory and then run sequentially. Without IoT Delegate, a service page need to access every cloud-scale IoT services to gather data from them. With IoT Delegate, web browser need to access IoT Delegate only once.

Fig. 9 shows round-trip delay of HealthCue application with and without IoT Delegate. HealthCue uses three cloud-scale IoT service as well as one peer-to-peer IoT service. To access each IoT service in HealthCue, it takes average 5,527ms. With IoT Delegate, it takes only 163ms (average) to receive the same amount of data.

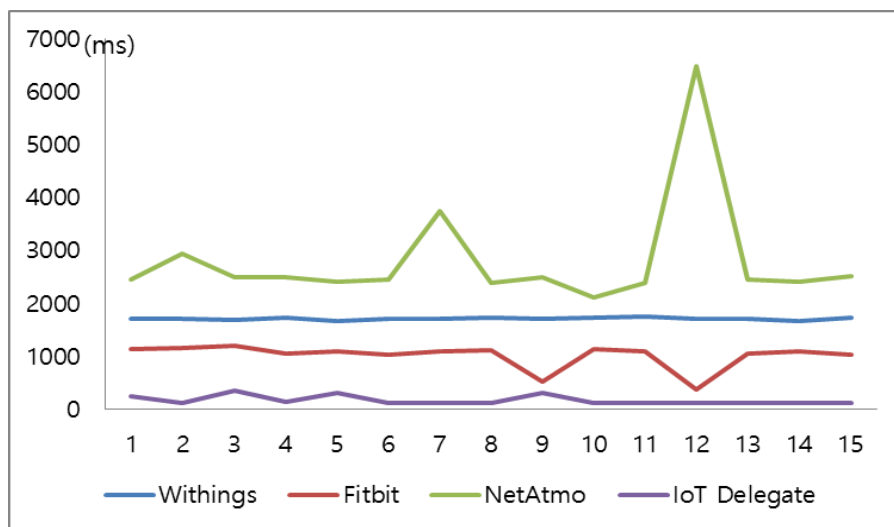


Fig. 9. Round-trip delay of HealthCue Service

6. Conclusion and Remarks

In this paper, IoT Delegate Framework for heterogeneous Smart Home IoT collaboration is proposed. Each device resource is abstracted to the DataChannel. Data Channels provide methods to access device resource with platform-independent RESTful architecture. To harmonize each smart home IoT device (which is DataChannel) more intuitively, an IDH API has been proposed. IDH API provides simple set of APIs for device collaboration: Trigger, Action and Intention. Two applications have been implemented with IDH API on smart TV, and they work properly as directed in Intentions. These two programming models which are defined in the framework provides more intuitive smart home development environment.

Implementation with real-world devices showed a practical example of application development using the proposed framework. Total six IoT Plug-ins are implemented for smart home devices and services. 23 DataChannels are implemented based on these Plug-in's, and used to create Intention for the applications. HeathCue service, showed real-time collaboration between heterogeneous smart home IoT devices. IoT Delegate showed better performance for service delivery time.

In the future, autonomous device collaboration service is going to be implemented based on the proposed framework. One of distinguishing features of the proposed framework is DataModel which describes IoT activity with timing data. By using this model, it can store all the user activities and temporal information regarding IoT. These IoT Activity logs with timing data are very valuable asset to build autonomous IoT collaboration. By adopting various machine learning algorithms to this IoT activity log, temporal relationship between IoT activities can be found and practical autonomous IoT collaboration service can be realized.

References

- [1] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess and 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.
[Article \(CrossRef Link\)](#)
- [2] A. Pintus, D. Carboni, A. Piras and A. Giordano, "Connecting Smart Things through Web Services Orchestrations," in *Proc. of ICWE Workshops*, vol. 6385, no. 41, pp. 431–441, 2010.
[Article \(CrossRef Link\)](#)
- [3] S. Han, G. Lee, N. Crespi, V. Nguyen, K. Heo, M. Brut and P. Gatellier, "DPWSim: A Devices Profile for Web Services (DPWS) Simulator," *IEEE Internet Things Journal*, pp. 1–1.
[Article \(CrossRef Link\)](#)
- [4] A. J. Jara, A. C. Olivieri, Y. Bocchi, M. Jung, W. Kastner and A. F. Skarmeta, "Semantic Web of Things: an analysis of the application semantics for the IoT moving towards the IoT convergence," *International Journal of Web and Grid Services*, vol. 10, no. 2, pp. 244–272, Apr. 2014.
[Article \(CrossRef Link\)](#)
- [5] S. D. Kim, J. Y. Lee, D. Y. Kim, C. W. Park and H. J. La, "Modeling BPEL-Based Collaborations with Heterogeneous IoT Devices," in *Proc. of 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing (DASC)*, pp. 289–294, 2014.
[Article \(CrossRef Link\)](#)
- [6] D. Sui, C. Hongbing, S. Jie and L. Haitao, "A model of task collaboration with simulation for IOT," in *Proc. of 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 1, pp. 331–335, 2011. [Article \(CrossRef Link\)](#)

- [7] S. Nastic, S. Sehic, M. Vögler, H.-L. Truong and S. Dustdar, "PatRICIA -- A Novel Programming Model for IoT Applications on Cloud Platforms," *SOCA '13: in Proc. of the 2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, pp. 53–60, 2013. [Article \(CrossRef Link\)](#)
- [8] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Communication Survey and Tutorials*, vol. 15, no. 3, pp. 1389–1406, Jul. 2013. [Article \(CrossRef Link\)](#)
- [9] M. Brut, P. Gatellier, D. Excoffier and I. Salhi, "When devices become collaborative: Supporting device interoperability and behaviour reconfiguration across emergency management scenario," *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 259 – 264, 2014. [Article \(CrossRef Link\)](#)
- [10] J. Jin, J. Gubbi, S. Marusic and M. Palaniswami, "An Information Framework for Creating a Smart City Through Internet of Things," *IEEE Internet Things Journal*, vol. 1, no. 2, pp. 112–121, May 2014. [Article \(CrossRef Link\)](#)
- [11] S. Cirani, L. Davoli, G. Ferrari, R. Leone, P. Medagliani, M. Picone and 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, Oct. 2014. [Article \(CrossRef Link\)](#)
- [12] R. Klauck and M. Kirsche, "Chatty things - Making the Internet of Things readily usable for the masses with XMPP," in *Proc. of 2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pp. 60–69, 2012. [Article \(CrossRef Link\)](#)
- [13] M. Kovatsch, M. Lanter and S. Duquenooy, "Actinium: A RESTful runtime container for scriptable Internet of Things applications," in *Proc. of 2012 3rd International Conference on the Internet of Things (IOT)*, pp. 135–142, 2012. [Article \(CrossRef Link\)](#)
- [14] J. Son, J.-H. Park, K.-D. Moon and 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. [Article \(CrossRef Link\)](#)
- [15] K.-K. DU, Z.-L. WANG and M. HONG, "Human machine interactive system on smart home of IoT," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, pp. 96–99, Aug. 2013. [Article \(CrossRef Link\)](#)
- [16] B. Jin, W. Zhuo, J. Hu, H. Chen and Y. Yang, "Specifying and detecting spatio-temporal events in the internet of things," *Decision Support Systems*, vol. 55, no. 1, pp. 256–269, Apr. 2013. [Article \(CrossRef Link\)](#)
- [17] S. Bourobou and Y. Yoo, "User Activity Recognition in Smart Homes Using Pattern Clustering Applied to Temporal ANN Algorithm," *Sensors*, vol. 15, no. 5, pp. 11953–11971, May 2015. [Article \(CrossRef Link\)](#)
- [18] Seung Woo Kum, JaeWon Moon, KunWoong Yuk, Taeboem Lim and Jong Il Park, "A Novel Design of IoT Cloud Delegate Framework to Harmonize Cloud-Scale IoT Services.," in *Proc. of International Conference on Consumer Electronics, ICCE '15*, pp.247-248, 2015. [Article \(CrossRef Link\)](#)



Seung Woo Kum received B.S degree in Electrics and Computer Engineering in Hanyang University, Ansan, Korea in 2000, and M.S degree in Hanyang University, Seoul, Korea in 2002. Currently, 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.



MinGoo Kang is a professor in the dept. of IT contents at Hanshin University, Osan South Korea from 2000. He has received the B.S., M.S., and Ph.D. degrees from Yonsei University, Seoul, Korea all in Electronic Engineering in 1986, 1989 and 1994, respectively. He was a research engineer at Samsung Electronics from 1985 to 1997. His research interests include wireless communication algorithm, mobile devices, and Smart UX for Mobile TV & DTV. He also served the ICONI 2014 hosted by the Korean Society of Internet Information, as the c



Jong-Il Park received the B.S., M.S. and Ph.D. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1987, 1989, and 1995, respectively. 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.