International Journal of Smart Home
Vol. 10, No. 9 (2016), pp. 275-286
http://dx.doi.org/10.14257/ijsh.2016.10.9.26

A Platform of Service Registry to Discover Service Resources in Ubiquitous Network

BenYan[1], Hua-Ping Yao[1], Masahide Nakamura[2], Shinsuke Matsumoto[2] and Hideharu Seto[2]

[1] LuoYang Institute of Science and Technology
No. 90 Wangcheng Road, Luolong District,
Luoyang City, Henan Province, 471023, China
[2] Kobe University
1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan
[1] {yanbenjp, yhplisajp}@gmail.com
[2] masa-n@cs.kobe-u.ac.jp

Abstract

In a ubiquitous network, a variety of things such as environmental sensor, home appliance and mobile phone are connected to an information network, which can be regarded as a service resource. Combined via network, these service resources can provide value-added ubiquitous services anywhere at anytime. In a future ubiquitous network, it is expected to achieve an adaptive service platform, which can dynamically integrate various service resources to provide adaptive and context-aware services.

This paper presents a new service registry platform named UBI-REGI, which is used to support efficient and dynamic discovery of service resources in a ubiquitous network. To define the scope of operation within the real world and IT world， UBI-REGI divides every service operation into three categories: source service, transformation service and sink service. Furthermore, UBI-REGI specifies meta-data like physical location or device owner to manage physical devices, in addition to the conventional meta-data like service name, purpose and description.

In order to enable UBI-REGI to be easily used by external program, this paper designs and implements UBI-REGI API by using Web service technologies. With this API, external applications can find service resources by queries of service category, location, purpose keywords， and so on. As a case study, this paper demonstrates an environment continue service in a home network system to prove the feasibility of this proposal.

***Keywords**: service resources, service discovery, data model, home network system, UDDI, Web services, home network system*

1. Introduction

A ubiquitous network， connecting a variety of things such as environmental sensor, home appliance and mobile phone to an information network， aims to provide various network services anywhere at anytime [1]. The study of such ubiquitous network is becoming more and more notable in recent years.

In a ubiquitous network, each feature of these devices can be regarded as a service resource. Such network achieves value-added ubiquitous services by combining these service resources via network.

For the conventional services, service resources are statically coupled within only one service (for example, a motion sensor embedded in a motion-sensing smart light cannot be accessed by other external services). But in the future ubiquitous network, it is expected to achieve an adaptive service platform, which can dynamically integrate various service resources to provide adaptive and context-aware services depending on the surrounding environment and private requirements.

A challenging issue to realize such platform is how to find appropriate service resources from huge number of services in a ubiquitous network. For example, suppose a user wants a service to broadcast a weather forecast when he goes out. The service resources [*SR1: a sensor detecting a user going out*], [*SR2: a service obtaining the weather forecast*], [*SR3：a service synthesizing voice from the forecast*], and [*SR4: a speaker playing back the voice*] are required. However, since there're large amount of devices in a ubiquitous network, it's difficult for a user to find, use and manage these service resources.

To solve this problem, this paper presents a new service registry named UBI-REGI to support efficient and dynamic discovery of service resources in a ubiquitous network. UBI-REGI divides every service operation into three categories: source service, transformation service and sink service, so as to define the scope of the operation within the real world and IT world. In addition to the conventional meta-data like service name, purpose and description, UBI-REGI also specifies meta-data like physical location or device owner to manage physical devices.

In order to enable UBI-REGI to be easily used by external program, this paper designs and implements UBI-REGI API by using web service technologies in the research. With API, external applications can find service resources by queries of service category, location, purpose keywords, and so on. As a case study, this paper demonstrates an environment migration service in a home network system to prove the feasibility of this proposal.

2. Preliminaries

2.1 Ubiquitous Service, Service Resource and Service Operation

A ubiquitous service in this paper is defined as a value-added service by using various devices connected with a network. The appliance integrated service of Home Network System (HNS) is a typical ubiquitous service [6]. HNS is a system consisting of multiple network household appliances and sensors. Each appliance or sensor can be controlled and managed via the home network. It is one of the most promising applications about the emerging ubiquitous technologies. The biggest advantage of HNS is that it provides value-added and more powerful HNS services by integrating multiple appliances and various sensors [11]. For example, integrating a TV, a DVD player, lights, sound system and curtains implements a *DVD theater service*, which allows a user to watch movies in a theater-like atmosphere at home. Also we can create a "*coming home service*" by integrating room lights and door sensor, which can turn on a room light automatically when a door sensor detects user coming home.

Ubiquitous services are realized by using or integrating certain service resources. Service resources include not only the above physical devices but also information systems, such as translation system, e-mail application etc. For HNS, service resources include all home appliances and sensors such as TV, DVD, curtain and sensor.

Generally speaking, a single service resource has several functions, which are called service operations in this paper. For example, an air conditioner (service resource) has service operations like "heating", "cooling", "dehumidifying', etc. Ubiquitous services can be realized with integrating these service operations provided by service resources.

2.2 Future Ubiquitous Service

Current ubiquitous services including service resources and service operations are statically predetermined by service provider. In the near future, more and more varied services will be included in a network. Future ubiquitous service will be able to integrate service with these service resources dynamically to satisfy users' request.

In the earlier study, service-oriented architecture (SOA) technology was selected to integrate service resources. Service operations of each service resource were designed as web services, and the developer can integrate service resources by accessing web service APIs. Depending on this design, an experimental HNS named CS27-HNS was developed. In this environment, a developer can operate each appliance and sensor by using standard HTTP protocols without relying on vendor-dependent control ways. The developer can also integrate other non-physical service resources such as weather forecast provided in the worldwide internet.

3. Research Goal and Approach

3.1 Research Goal

In the previous study of CS27-HNS, it was supposed that a developer knew the details of every ubiquitous service, so the endpoints of service operations were statically hard-coded in each program. Therefore, whenever new service resources are published, service developers need to fix and revise their programs. It is impossible to add service resources or to integrate service resources dynamically.

This paper focuses on how to implement a platform to search and manage numerous service resources efficiently and dynamically in a ubiquitous network. To achieve this purpose, a service registry named UBI-REGI was developed, which supports efficient and dynamic discovery of service resources in a ubiquitous network.

3.2. Key Idea

Considering the characteristic of ubiquitous service resources, a data modeling of UBI-REGI was conducted based on the following key ideas.

- **(K1) Categorize service operations:** Categorize all the service operations into three types: the source type, the transformation type and the sink type.
- **(K2) Associate physical location:** Associate the source and sink service with physical location.
- **(K3) Associate purpose keyword:** Associate every service operation with a set of keywords representing its purpose.

3.2.1 Categorize Service Operations: In a ubiquitous network, many service resources are involved in both real and IT worlds. The real world includes appliances, sensors, etc., and the IT word includes ubiquitous services, service operations, etc. The sensors of the real world provide the value to services of the IT world. The appliance of the real world is operated by the services of IT world depending on the value of sensors. Value-added service is provided to a user who is in the real world. Figure 1 depicts the information flow among ubiquitous resources. It can be seen that the source services including sensors and weather forecast are placed at the boundary from the real world to the IT world. Based on the observation, it is proposed to categorize every service operation into the following three types.

- **(C1) Source Service:** A service that obtains physical information of the real world, and imports it as digital data to the IT world. Typical source services include sensor services measuring temperature and humidity as well as RSS feeds providing news as digital data.

- **(C2) Transformation Service:** A service that converts data into another data (usually with more value) and outputs it into the IT world again. Typical services include text-to-speech service converting text data to voice, or language translation service translating English text into Japanese.
- **(C3) Sink Service:** A service that causes a physical impact to the real world based on data in the IT worlds. Typical services include the air-conditioner service heating a room, the email service sending a message to a human user, or the speaker service playing the voice data in a room.

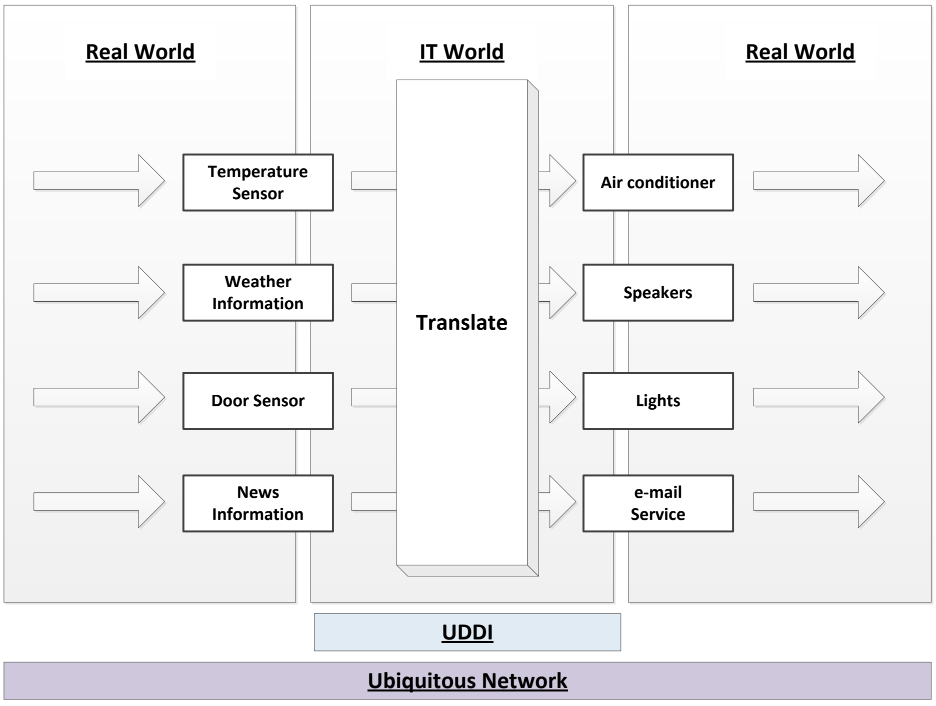


Figure 1. Different Scope of Existing Services (UDDI) and Ubiquitous Services

Transformation service is shown in the middle, which is within the IT world. Sink services including appliances and email are located in the exit from the IT world to the real world. The proposed three categories clarify the role of service operations. Generally, a ubiquitous service can be implemented in this order like the source, transformation, and sink services. A user can search appropriate service resources/operations by categories, and use them as components of the desired ubiquitous service. For example, if a user wants sensor services, he searches them from the source services. If he wants output devices, he searches them from the sink services.

3.2.2 Associate Physical Location: According to the definition above (in section 3.1.1), it's known that source service and sink service are tightly linked to the physical environment (location information). Therefore, the location information is needed to show where the obtaining information is from and where the service affects. For example, there are several air-conditioner services in a building. If there is no information about where each air-conditioner is installed, a user can not identify the most suitable service from them. Thus it is proposed to associate every source service or sink service with physical location to show where the service resource is installed.

Referenced the spatial modeling method [3], the physical location is defined by a tree model of location entities. The tree model is described by spatial inclusion relations such as buildings, rooms and devices. For example, Figure 2 shows a tree model of physical location of a house. There is a living room, a kitchen and a bedroom in the house. Also, there is the location of a TV in the living room.

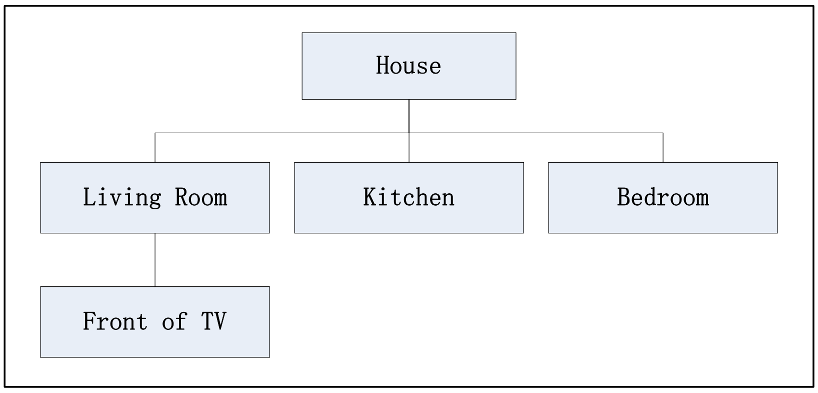


Figure 2. Tree Model of Physical Location

By associating every service resource with a location entity, it is possible to search service resources by physical location. In the example of air conditioner services, a user can identify the suitable air-conditioner by a location where the user is currently staying. This tree model also enables the search queries with the inclusion relations. For example, in Figure 2, all air conditioners installed in the house can be searched by traversing the tree from the root to the leaves. Even if a motion sensor is not directly associated with the living room, we can use the sensor in front of the TV as a substitute.

3.2.3 Associate Purpose Keywords： As seen in Web search, when a user searches a service operation whose name is unknown to him, it would be convenient for him to search it by keywords closely related to the purpose and feature of the service operation. Therefore, purpose keywords are defined to represent the purpose of service operations.

In this proposal, input/output attribute name and function name are used as purpose keywords. For the source or sink services, keywords are defined by the names of environment properties affected by the service operation. For example, *air-conditioner.heating*() is associated with keywords like temperature, air-flow, humidity. A user can identify *air-conditioner.heating*() by specifying “*temperature*” as the purpose, and “*sink*” as the category. He does not need to specify the concrete device name.

3.3 Data Model of the UBI-REGI

Based on the above key idea, a data modeling of UBI-REGI that consists of three portions are conducted. Figure 3 shows the ER diagram. A box represents an entity (i.e., table) consisting of an entity name, a primary key and attributes. Instances below each entity are enumerated. A line represents a relationship between entities, where △ denotes a parent-children relationship, and ▲ denotes a reference relationship.

3.3.1 Definition of Physical Location: The physical location is managed by a LOCATION table which has three columns: a location ID as the primary key, a reference to parent location, and a set of location descriptions.

3.3.2 Definition of Purpose Keyword: The purpose keywords are managed by a KEYWORD table which has two columns: a keyword ID as the primary key and a set of keywords. An instance K002 in Figure 3 represents a set of keywords describing the temperature. By registering synonyms in the same set, it is possible to search service operations by various words like “air temperature, room air temperature and ambient temperature”.

3.3.3 Definition of Service Resources: The service resources are managed by three tables: the SERVICE RESOURCE table, the SERVICE OPERATION table and the PARAMETER table.

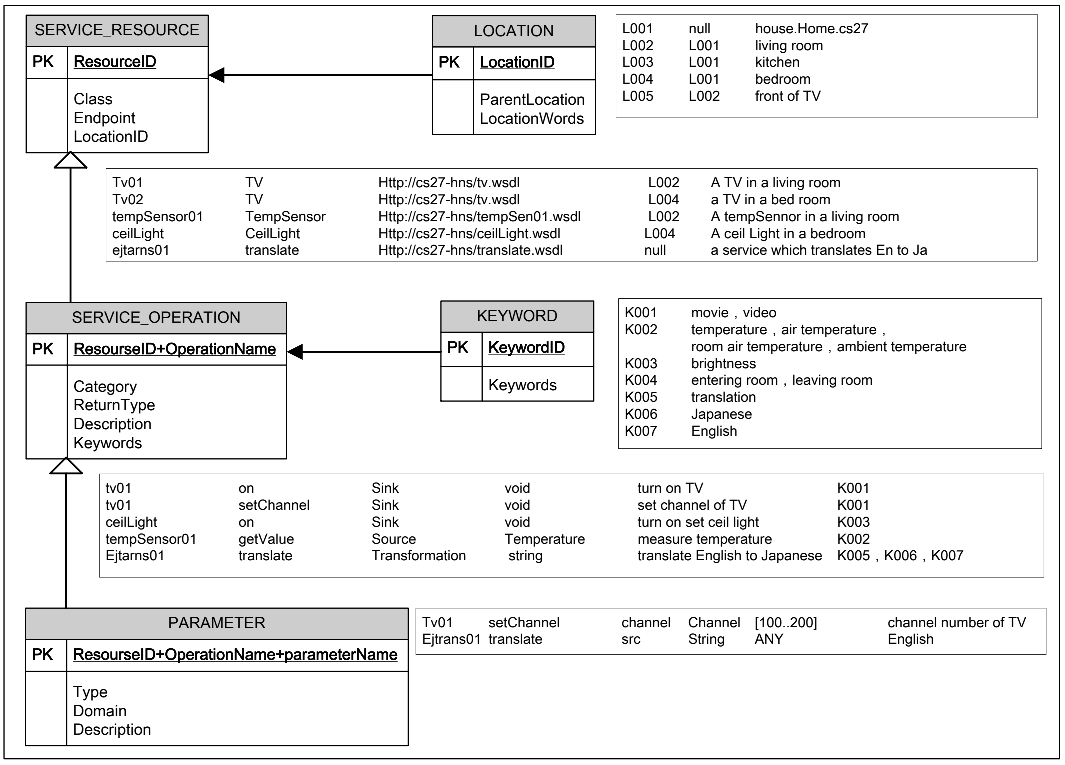
The SERVICE RESOURCE table has five columns: a resource ID, a device class, an endpoint, a physical location and a description. The device class represents an appliance or an abstract type of the software. Note that a unique resource ID is given to every instance of resource. If there are two TVs in a house, they must be distinguished as “tv01” in the living room, and “tv02” in the bed room.

The device classes are determined by standardized data model for home network system proposed in [4]. The instance tv01 in Figure 3 represents a TV in a living room, whose physical location is L002 and the endpoint of the Web service is http://cs27-hns/tv.wsdl.

The service operations provided by each service resources are managed by SERVICE OPERATION table. A resource ID and an operation name form a composite primary key. There is a service category (source, transformation or sink), a type of return value, an operation description and the purpose keyword. One service operation is associated with a set of keywords. An instance “tv01.on” in Figure 3 represents that the operation provides a sink service, and that the type of return value is void and associated with keywords like “movie and video”.

On the other hand, “*tempSensor01.getValue*” is a source service, which acquires data from the real worlds. The type of return value is temperature. The associated keywords are temperature, air temperature, room air temperature and ambient temperature.

The parameters are managed by PARAMETER table with 6 columns. A primary key is composed of a resource ID, an operation name and a parameter name. As for the attributes, there is a parameter type, a parameter domain and a parameter description. An instance “tv01.setChannel.channel” in Figure 3 represents that the parameter is typed by Channel and takes a value from 100 to 200.



L001	null	house.Home.cs27
L002	L001	living room
L003	L001	kitchen
L004	L001	bedroom
L005	L002	front of TV

Tv01	TV	Http://cs27-hns/tv.wsdl	L002	A TV in a living room
Tv02	TV	Http://cs27-hns/tv.wsdl	L004	a TV in a bed room
tempSensor01	TempSensor	Http://cs27-hns/tempSen01.wsdl	L002	A tempSennor in a living room
ceilLight	CeilLight	Http://cs27-hns/ceilLight.wsdl	L004	A ceil Light in a bedroom
ejtarns01	translate	Http://cs27-hns/translate.wsdl	null	a service which translates En to Ja

K001	movie，video
K002	temperature，air temperature，room air temperature，ambient temperature
K003	brightness
K004	entering room，leaving room
K005	translation
K006	Japanese
K007	English

tv01	on	Sink	void	turn on TV	K001
tv01	setChannel	Sink	void	set channel of TV	K001
ceilLight	on	Sink	void	turn on set ceil light	K003
tempSensor01	getValue	Source	Temperature	measure temperature	K002
Ejtarns01	translate	Transformation	string	translate English to Japanese	K005，K006，K007

Tv01	setChannel	channel	Channel	[100..200]	channel number of TV
Ejtrans01	translate	src	String	ANY	English

Figure 3. Data Model of the UBI-REGI

3.4 UBI-REGI API

In order to enable UBI-REGI to be easily used by external program, UBI-REGI API "*getService*()" is defined by using Web service technologies. Figure 4 shows the details about the API *getService*().

The *getService*() has argument that includes service *category*, resource *class*, physical *location* and purpose *keyword*. The API returns a list of service operations objects. The fields of the return object include *method*, *endPoint*, *locationIDpath*, *locationNamePath*, *classID*, *returnType*, *description* and *parameterObject*. The fields of *parameterObject* include *type*, *name*, *domain* and *descriptions*.

```
* Interface:
        getService(category, class, location, keyword);

* Query Parameters:
        - category: Source, Transformation or Sink
        - class: Device class (device name, service name, etc.)
         - location: Physical location
         - keyword: Purpose keyword

* Return:
      A list of operations matching queries
```

Figure 4. The Details about API getService()

Table 1. The Details of returnObject

fields	detail	example
method	The name of method	setChannel
endPoint	End point	http://cs27-hns/tv.wsdl
locationIDPath	The path from location ID	L001/L002
locationNamePath	The path from location name	Home/Living room
class	The name of resource class	Tv
returnType	The type of return value	Void
description	The details about service	Setting TV channel
parameterObject[]	The details about parameter	-

Table 2. The Details of ParameterObject

fields	detail	example
type	The type of parameter	int
name	The name of parameter	Channel
domain	The value of the parameter	[100 .. 200]
description	The description about parameter	The channel of TV

4. Case Study

4.1 Implementation of UBI-REGI

Depending on the above discussion, UBI-REGI was implemented in our testing environment CS27-HNS. The developing environment and the technique we used are as follows:

■ Development Tools:	Eclipse3.4.0
■ API Development Language:	Java
■ Database:	MySQL 5.1
■ Web Server & Engine:	Apache Tomcat 5.5 & Apache Axis 2.1.3

4.2 Case Study

In this section, a case study is used to prove the effectiveness of this proposal. The case study selected [*environment continue service*] of CS27-HNS as an implementing object. This service means if the user moves from living room to bedroom, the [*environment continue service*] will translate the same environment of the living room in the bedroom, such as temperature, brightness, etc. The scenario of service is as follows.

(1) When the service detected the user leaving the living room, it will acquire the temperature and the brightness of the living room.

(2) After (1), when the service detected the user entering the bedroom, the service will operate the equipments of the bedroom to reproduce the same environment in the living room.

Depending on the key idea (K1) in section3.2, the [*environment continue service*] should be achieved by three kinds of services: the source service, the transformation service and the sink service. In this case study, the source service is to obtain the temperature and the brightness of living room, to detect the user is leaving the living room or not, and to detect the user is entering the bedroom or not. After obtaining these data, the source service will import them to the IT world (transformation service). The sink service is to set the same environment in the bed room based on the data and logic which are imported and implemented in the transformation service (IT world).

Based on the analysis of the environment continue service's scenario above, it's known that six source services and two sink services are needed to achieve the service (Table 3).

Table 3. The Details of Source Service and Sink Service

environment continue service	
source services	
[*SO1*]	The service to acquire the temperature of the living room.
[*SO2*]	The service to acquire the brightness of the living room.
[*SO3*]	The service to acquire the temperature of the bedroom.
[*SO4*]	The service to acquire the brightness of the bedroom.
[*SO5*]	The service to detect people entering or leaving the living room.
[*SO6*]	The service to detect people entering or leaving the bedroom.
sink services	
[*SI1*]	The service to change the temperature of the bedroom.
[*SI2*]	The service to change the illumination of the bedroom.

Now the biggest issue is how to search and select suitable resource service and sink service to achieve the [*environment continue service*]. In this case study, we use UBI-REGI platform to search the suitable resource service with the following queries (Figure 5).

- **For the six resource services:**

qo1 = {category=> “Source”, class=> “ANY”, location => “living room”, keyword=> “temperature”}

qo2 = {category=> “Source”, class=> “ANY”, location => “living room”, keyword=> “Light”}

qo3 = {category=> “Source”, class=> “ANY”, location => “bed room”, keyword=> “temperature”}

qo4 = {category=> “Source”, class=> “ANY”, location => “bed room”, keyword=> “Light”}

qo5 = {category=> “Source”, class=> “ANY”, location => “living room”, keyword=> “leaving”}

qo6 = {category=> “Source”, class=> “ANY”, location => “bed room”, keyword=> “entering”}

- **for the two sink services:**

si1 = {category=> “sink”, class=> “ANY”, location => “bed room”, keyword=> “temperature”}

si2 = {category=> “sink”, class=> “ANY”, location => “bed room”, keyword=> “Light”}

Figure 5. The Details of the Queries

In figure 5, the queries qo1 to qo4 is to obtain the temperature and the brightness of the leaving room (or the bedroom); Qo5 to qo6 is to detect a person is leaving (or entering) the living room (or the bedroom); the si1 to si2 is to change the temperature and brightness of the bedroom.

environment continue service		
source services		
query	**Service Operation**	**Service Resource**
[*SO1*]	tempSensor01.getValue()	http://cs27-hns/tempSen01.wsdl
[*SO2*]	lightSensor01.getValue()	http://cs27-hns/lightSen01.wsdl
[*SO3*]	tempSensor02.getValue()	http://cs27-hns/tempSen02.wsdl
[*SO4*]	lightSensor02.getValue()	http://cs27-hns/lightSen02.wsdl
[*SO5*]	doorSensor01.getValue()	http://cs27-hns/doorSen01.wsdl
[*SO6*]	doorSensor02.getValue()	http://cs27-hns/doorSen02.wsdl
sink services		
query	**Service Operation**	**Service Resource**
[*SI1*]	airCon.on()	http://cs27-hns/aircon.wsdl
[*SI2*]	ceilLight.on(), tableLight.on()	http://cs27-hns/ceilLight.wsdl http://cs27-hns/tableLight.wsdl

Table 4. The Searching Result

Table 4 shows the searching result based on the data molding in section 3.3. After this, the developer only needs to implement service logic such as “if the [*brightness of the living room > brightness of the bedroom]*, call the *ceilLight.on*()”. The [*environment continue service*] can be realized by using these resource service operations. Just like this, a developer can dynamically search and select the suitable service resources in a ubiquitous network by appointing key words like service category, physical location and the purpose key word.

5. Discussion

5.1. Summary of Contribution

In this paper, a new service registry platform called UBI-REGI is presented to support the efficient and dynamic discovery of service resources in a ubiquitous network. In this proposal, UBI-REGI categorizes service operation into source service, transformation service and sink service, which can define the scope of the operation within the real and the IT worlds.

The Source Service is to obtain physical information of the real world, and import it as digital data to the IT world. The Transformation Service is to convert data into another data (usually with more value) and output it into the IT world again. The sink service is to cause the physical impact to the real world based on data in the IT worlds.

Furthermore, in order to support a developer to search and select suitable service resources in a ubiquitous network, meta-data are specified like physical location, device owner, service name, purpose and description in the data modeling of UBI-REGI. An UBI-REGI API is also designed and implemented to enable UBI-REGI to be used easily by external program. With the API, the external applications can dynamically find service and select suitable service resources with the queries of service category, location, purpose keywords, and so on.

At last, as a case study, an environment migration service is demonstrated in a home network system to prove the feasibility of this proposal.

5.2. Related Work

In the area of SOA, UDDI (universal description, discovery and integration) has been a standard of service registries, with which clients can find and manage the service resources [2]. However, in a ubiquitous network, every service resources are heavily associated with the physical world. For example, if a user wants to use a speaker in a living room, the user must find an appropriate one from a lot of speaker services in the network. The conventional UDDI does not cover the physical system, so it cannot be directly used.

5.3. Future Work

As for the future work, the definition of resource class will be elaborated to improve the quality of resource search. The effectiveness of UBI-REGI will also be evaluated through other applications, including a ubiquitous cloud. Finally, it is interesting to introduce the ontology in the purpose keywords, which enables semantic-based service discovery.

Acknowledgements

This research was partially supported by the Scientific Research Start Funds of LuoYang Institute of Science and Technology (No.14308051); the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (B) (No.26280115, No.15H02701), Young Scientists (B) (No.26730155)]; and the International scientific and technological cooperation project (2015) (152102410053) supported by Science and Technology Department of Henan Province, China.

References

[1] Bill N. Schilit and Norman Adams and Roy Want. Context-Aware Computing Applications. Proceedings of the 1st IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), (1994) December 8-9; Washington, DC, USA.
[2] Anind K. Dey and Gregory D. Abowd. Towards a Better Understanding of context and context-awareness. Proceesing of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC), (1999) September 27-29; Karlsruhe, Germany.
[3] Hiroyuki SAKAMOTO, Hiroshi IGAKI, Masahide NAKAMURA, A Sensor Service Framework for Context-aware Applications, TECHNICAL REPORT OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, (2009) March, Vol.108, no.458, pp.381-386.
[4] Thomas Erl, Service-Oriented Architecture: Concepts, Technology and Design, PRENTICE HALL (2008).
[5] Hiroyuki SAKAMOTO, Hiroshi IGAKI, Masahide NAKAMURA, SMuP:A Service-oriented Platform for Sensor Service Mashups, Winter workshop 2010 in KURASIKI, (2010) January,Vol.2010, no.3, pp.73-74.
[6] Akihiro TANAKA, Masahide NAKAMURA, Hiroshi IGAKI, Ken-ichi MATSUMOTO, Adapting Conventional Home Appliances to Home Network Systems Using Web Services, TECHNICAL REPORT OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, (2006) March, Vol.105, no.628, pp.67-72.
[7] Masayuki FUKUDA, Hideharu SETO, Hiroyuki SAKAMOTO, Hiroshi IGAKI, Masahide NAKAMURA, A Looking Back Service for Power Consumption Logs in Home Network System, TECHNICAL REPORT OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, (2009) November Vol.109, no.272, pp.29-34.
[8] Masahide NAKAMURA, Hiroshi IGAKI, Haruaki TAMADA, Ken-ichi MATSUMOTO, Implementing Integrated Services of Networked Home Appliances Using Service-oriented Architecture, the Journal of Information Processing Society of Japan, (2015) February Vol.46, no.2, pp.314-326.
[9] Phidgets Inc. Unique and Easy to Use USB Interface,http://www.phidgets.com/.
[10] MATSUO Syuhei, SETO Hideharu, SAKAMOTO Hiroyuki, Hiroshi IGAKI, Masahide NAKAMURA, Sensor search with spatial information and support by showing similar parameter for building sensor context, TECHNICAL REPORT OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, (2009) December ,Vol.109, no.327, pp.59-64.
[11] Ben Yan, Masahide Nakamura, Lydie du-Bousquet, and Ken-ichi Matsumoto, Validating Safety for Integrated Services of Home Network System Using JML, Journal of Information Processing (JIP), (2008) June, Vol.49, no.6, pp.1751-1762.
[12] Ben Yan, Masahide Nakamura, and Ken-ichi Matsumoto, Deriving Safety Properties for Home Network System Based on Goal-Oriented Hazard Analysis Model, International Journal of Smart Home, (2009) January, Vol.3, no.1, pp.67-80.
[13] Ben Yan, Masahide Nakamura, Lydie du Bousquet, and Ken-ichi Matsumoto, Improving Reusability of Hazard Analysis Model with Hazard Template for Deriving Safety Properties of Home Network System, International Journal of Smart Home, (2009) April, Vol.3, no.2, pp.71-88.
[14] Lydie du-Bousquet, Masahide Nakamura, Ben Yan, and Hiroshi Igaki, Using Formal Methods to Increase Confidence in a Home Network System Implementation: a Case Study, Innovations in Systems and Software Engineering (ISSE Journal), (2009) September, Vol.5, no.3, pp.181-196.
[15] M. Weiser. The computer for the 21st century. Scientific American, 265(3):94–104, 1991. [2] Uddi. http://uddi.xml.org/.
[16] Ichiro Satoh. A programming model for smart spaces (ubiquitous computing). Information Processing Society of Japan, 45(12):2655–2665, 2004.
[17] Kouichi Egami, Hiroshi Igaki, and Masahide Nakamura. A standardized data model for networked appliances in home network system. Technical Report 11, 2009 (in Japanese).

Authors

Ben Yan. Ben Yan received the B.E. degree in Henan University of Science and Technology, China, in 1999, M.E. degree in Department of Information Science Okayama University of Science, Japan, in 2006, and Ph.D. degree in the Graduate School of Information Science at Nara Institute of Science and Technology, Japan, in 2008. From 2009 to 2014, he worked for Panasonic Group, SANYO Information Technology Solutions Co., Ltd, Osaka, Japan. He is currently a professor in the Department of Computer and

Information Engineering at Luoyang Institute of Science and Technology (LIT). His main research interests include the service-oriented architecture, the V&V of home network systems, and requirements engineering for safety critical systems.

HuaPing Yao. HuaPing Yao received the B.E. degree in Henan University of Science and Technology, China, in 1999, M.E. degree in Department of Information Science Okayama University of Science, Japan, in 2006. From 2006 to 2014, he worked for CSI and Trend Creates Co., Ltd, Osaka, Japan. She is currently a lecturer in the Department of Computer and Information Engineering at Luoyang Institute of Science and Technology (LIT). Her main research interests include the e-learning, software engineering and home network system.

Masahide Nakamura. Masahide Nakamura received the B.E., M.E., and Ph.D. degrees in Information and Computer Sciences from Osaka University, Japan, in 1994, 1996, 1999, respectively. From 1999 to 2000, he has been a post-doctoral fellow in SITE at University of Ottawa, Canada. He joined Cyber media Center at Osaka University from 2000 to 2002. From 2002 to 2007, he worked for the Graduate School of Information Science at Nara Institute of Science and Technology, Japan. He is currently an associate professor in the Graduate School of System Informatics at Kobe University. His research interests include the service /cloud computing, smart home, smart city, and life log. He is a member of the IEEE, IEICE and IPSJ.

Shinsuke Matsumoto. Shinsuke Matsumoto received the B.E. degree in computer science from Kyoto Sangyo University in 2006. He received M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology in 2008 and 2010, respectively. He is currently an assistant professor in the Graduate School of System Informatics at Kobe University. His research interests include software engineering, mining software repository and cloud computing.