UBI-REGI: Service Registry for Discovering Service Resources in Ubiquitous Network

Hideharu SETO, Shinsuke MATSUMOTO, Masahide NAKAMURA
Graduate School of System Informatics, Kobe University
1-1, Rokkodai-cho, Nada, Kobe, Hyogo 657-8501, Japan
seto@ws.cs.kobe-u.ac.jp, {shinsuke, masa-n}@cs.kobe-u.ac.jp

ABSTRACT

In the future ubiquitous network, it is expected to realize an adaptive service platform, which dynamically integrates various service resources to provide adaptive and context-aware services. This paper presents a service registry, called *UBI-REGI*, in order to support efficient and dynamic discovery of service resources in the ubiquitous network.

Categories and Subject Descriptors

H.2.1 [**Database Management**]: Logical Design—*Data models*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-Based services*

General Terms

Design, Experimentation

Keywords

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

1. INTRODUCTION

The *ubiquitous network* [1] aims to provide various network services at anytime and anywhere. In the ubiquitous network, a variety of *things* such as environmental sensor, home appliance and mobile phone are connected to the information network. Every feature of such devices can be regarded as a *service resource*. Collaborating these resources via network achieves value-added *ubiquitous services*.

In the conventional services, service resources are statically coupled within one service. For example, a motion sensor embedded in a motion-sensing smart light cannot be accessed by other external services. In the future ubiquitous network, it is expected to achieve an *adaptive service platform*, which dynamically integrates various service resources to provide adaptive and context-aware services. For such dynamic and flexible integration of the heterogeneous things, the *service-oriented architecture (SOA)* and the *cloud computing* are promising technologies.

A challenging issue to realize the platform is how to find appropriate service resources from enormous services in the ubiquitous network. For example, suppose that a user wants a service speaking a weather forecast when he goes out. Then, the following service resources are required: a sensor detecting the user going out, a service obtaining the weather forecast, a service synthesizing voice from the forecast, and a speaker playing back the voice. However, in the ubiquitous network with enormous devices, it is difficult to manage these resources manually, and to discover appropriate ones.

In the area of SOA, the *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 the 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.

This paper presents a novel service registry, called *UBI-REGI*, in order to support efficient and dynamic discovery of service resources in the ubiquitous network. The UBI-REGI categorizes every service operation into three types, *source, transformation or sink*, to define the scope of the operation within the real and IT worlds. Furthermore, the UBI-REGI specifies meta-data like *physical location* or *device owner* for managing physical devices, in addition to the conventional meta-data like service name, purpose and description.

In this paper, we also design and implement the *UBI-REGI API* using Web service technologies. With the API, the external applications can find service resources by queries of service category, location, purpose keywords and so on.

2. PRELIMINARY

2.1 Ubiquitous Service, Service Resource and Service Operation

In this paper, we define a *ubiquitous service* as a value-added service which uses various ubiquitous devices connected to the ubiquitous network. An example of a ubiquitous service, there is “appliance cooperation service” in *home network system (HNS)*. HNS is intended to provide more convenient and comfortable living for home users. HNS consists of some appliance services, sensor services and a home server. Each of appliance and sensor can be controlled and managed via home network.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
iiWAS2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam..
Copyright 2011 ACM 978-1-4503-0784-0/11/12 ...$10.00.

The "appliance cooperation service" provides value-added services by orchestrating two or more networked appliances and sensors. For example, orchestrating TV, DVD, curtain, light and speaker provides a "DVD theater service", where a user can watch movies in a theater-like atmosphere at home. Also we can create a "coming home service", which turn on a room light when a door sensor detects user's coming home.

These ubiquitous services are realized by using some *service resources*. In the HNS environment, the service resource includes all of home appliances and sensors such as TV, DVD, curtain, door sensor and motion sensor. The service resource includes not only above physical devices but also information systems like a translation system and e-mail application.

Generally, a single service resource has several functions. In this paper, we call these functions *service operations*. For example, an air conditioner, which is a service resource, has following service operations: heating, cooling, dehumidifying. Ubiquitous services can be realized by cooperating with these service operations provided by service resources.

2.2 Future Ubiquitous Service

Current ubiquitous services assume that service resources and service operations, used by ubiquitous service, are statically predetermined by service provider. In the near future, as more varied services are connected to the information network, it is expected to dynamically integrate various service resources and service operations to provide adaptive context-aware services

In order to achieve the above future ubiquitous service, *service-oriented architecture (SOA)* is a smart solution for the dynamic service resource integration. Wrapping a service resource as a Web service provides loose-coupling and platform-independent access methods. Furthermore by integrated these Web services, we can dynamically create a ubiquitous service.

By applying the SOA to HNS, we can operate each appliance and sensor using standard HTTP protocols without relying on vendor-dependent control ways. We can also cooperate other non-physical service resources such as weather forecast provided in the worldwide internet.

In this paper, we assume a future ubiquitous network, where every service resources are wrapped as a Web service. We also assumes these Web services are defined by standard interface description (e.g., WSDL) and can be accessed by HTTP-based protocols (e.g., REST and SOAP).

2.3 Scope of This Research

Challenges for the future ubiquitous services, we need to find and manage numerous service resources efficiently and dynamically. In this paper, we focus on *how to search and manage various service resources in the ubiquitous network*. In order to achieve this purpose, we develop a service registry, called *UBI-REGI*, which supports efficient and dynamic discovery of service resources in the ubiquitous network.

In our previous work of CS27-HNS, we have assumed every ubiquitous services are known at service development. 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 program. Accordingly, it is impossible to dynamically add service resources and to integrate service resources. Using UBI-REGI, we think these problems are solved because every ubiquitous services can dynamically search and obtain each service resource.

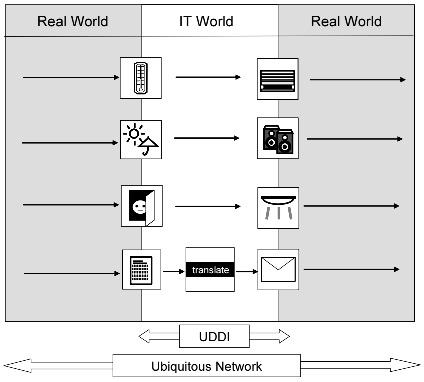


Figure 1: Different Scope of Existing Services and Ubiquitous Services

3. UBI-REGI: UBIQUITOUS SERVICE RESOURCE REGISTRY

3.1 Key Idea

Considering the nature of the ubiquitous service resources into account, we conduct data modeling of the UBI-REGI based on the following key ideas.

(K1) Categorize service operations: Categorize every service operations into one of three types, *source*, *transformation* or *sink*.

(K2) Associate physical location: Associate the source and sink services with *physical location*.

(K3) Associate purpose keyword: Associate every service operation with a set of keywords representing its purpose.

3.2 Categorize Service Operations

Many service resources in the ubiquitous network are involved in both real and IT worlds. Based on the observation, we propose to categorize every service operation in 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 in the IT world again. Typical services include the text-to-speech service converting text data to voice, the language translation service translating English text into Japanese.

(C3) Sink Service: A service that causes a physical impact to the the real world based on data in the IT worlds. Typical services include the air-conditioner

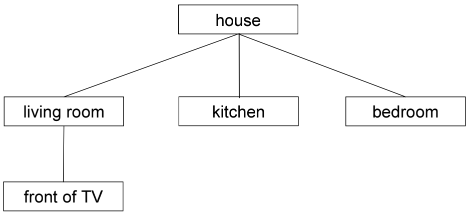


Figure 2: Tree Model of Physical Location

service heating a room, the email service sending a message to a human user, the speaker service playing the voice data in a room.

Figure 1 depicts the information flow among ubiquitous resources, where we can see the difference of scope between the conventional UDDI and the proposed UBI-REGI. It can be seen that the source services including sensors and weather forecast are placed in the boundary from the real world to the IT world. The transformation service is shown in the middle, which is within the IT world. The 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 the service operations, which helps efficient resource discovery. Generally, a ubiquitous service can be implemented by concatenating the source, transformation, and sink services in this order.

A client of the UBI-REGI 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 source services. If he wants output devices, he searches them from sink services. Note that the conventional Web services managed by the UDDI are categorized as transformation services.

3.3 Associate Physical Location

The source and sink services are tightly linked to the physical environment. Therefore, we need the location information where the service affects. For example, suppose that there are several air-conditioner services in a building. If there is no information about where each air-conditioner is installed, the user would not be able to identify the most suitable service from them.

Accordingly, we propose to associate every source or sink service with *physical location*, which explains where the service resource is installed. Inspired by a spatial modeling method [3], we define the physical location by a *tree model* of location entities. The tree model is described by the spatial inclusion relations such as buildings, rooms and devices. For example, Figure 2 shows a tree model of physical location of a house. There are a living room, a kitchen and a bedroom in the house. Also, there is the location of a TV in the living room.

By associating every service resource with a location entity, it is possible to search service resources by the physical location. In the example of the air conditioner services, the user can identify the suitable air-conditioner by a location where the user is currently staying. Also, the tree model enables the search queries with the inclusion relations. For example, in Figure 2, we can search all air conditioners installed in the house by traversing the tree from the root to the leaves. Also, 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.

3.4 Associate Purpose Keywords

As seen in the Web search, when a user searches a service operation whose name is unknown for the user, it would be convenient for him to search it by keywords that are closely related to the purpose and feature of the service operations. Therefore, we define *purpose keywords* representing the purpose of service operations.

For a service operation, there are many ways to define the associated purpose keywords. In the UBI-REGI, as for the transformation service, we use *input/output attribute names* and the *function name*. As for the source or sink services, we use the *names of environment properties* affected by the service operation. For instance, `air-conditioner.heating()` is associated with keywords like *temperature, air-flow, humidity*. A user, who wants to warm a room, can identify `air-conditioner.heating()` by specifying "temperature" as the purpose, and "sink" as the category. It is not necessary for the user to specify the concrete device name.

3.5 Data Model of the UBI-REGI

Based on the above discussion, we have conducted data modeling of the UBI-REGI. Figure 3 depicts the ER diagram. A box represents an entity (i.e., table), consisting of an entity name, a primary key and attributes. We enumerate instances below each entity. A line represents a relationship between entities, where +—∈ denotes a parent-children relationship, and +—· · · denotes a reference relationship. The data model consists of the following three portions.

Definition of Physical Location

The physical location discussed in Section 3.3 is managed by the LOCATION table. This table has three columns: a location ID as the primary key, a reference to parent location, and a set of location descriptions. The instances represent the tree model shown in Figure 2.

Definition of Purpose Keyword

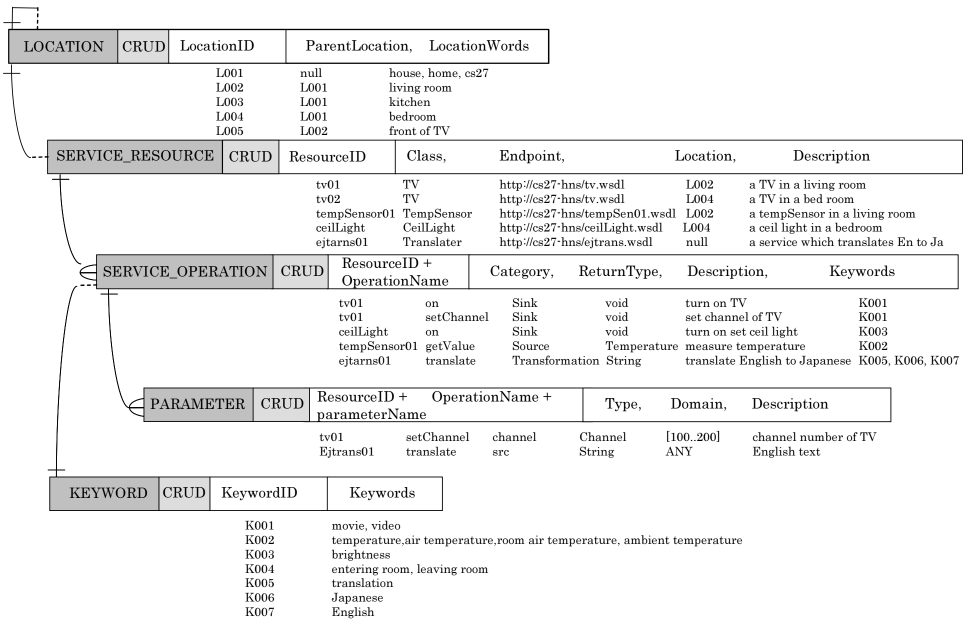
The purpose keywords are managed by the KEYWORD table. This table has two columns: a keyword ID as the primary key and a set of keywords. An instance K002 in Figure 3 represents a set of keywords describing the temperature. By registering synonyms in the same set, it is possible to search service operations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

Definition of Service Resources

The service resources are managed by three table: 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 a kind of the appliances or an abstract type of the software. Note that a unique resource ID is given for every instance of resource. When there are two TV in a house, these two must be distinguished as "tv01" in the living room, and "tv02" in the bed room.

The device classes have been determined by the standardized data model for home network system proposed in [4].



LOCATION	CRUD	LocationID	ParentLocation,	LocationWords
		L001	null	house, home, cs27
		L002	L001	living room
		L003	L001	kitchen
		L004	L001	bedroom
		L005	L002	front of TV

SERVICE_RESOURCE	CRUD	ResourceID	Class,	Endpoint,	Location,	Description
		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 tempSensor in a living room
		ceilLight	CeilLight	http://cs27-hns/ceilLight.wsdl	L004	a ceil light in a bedroom
		ejtarns01	Translater	http://cs27-hns/ejtrans.wsdl	null	a service which translates En to Ja

SERVICE_OPERATION	CRUD	ResourceID +	OperationName	Category,	ReturnType,	Description,	Keywords
		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

PARAMETER	CRUD	ResourceID +	OperationName +	parameterName	Type,	Domain,	Description
		tv01	setChannel	channel	Channel	[100..200]	channel number of TV
		Ejtrans01	translate	src	String	ANY	English text

KEYWORD	CRUD	KeywordID	Keywords
		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

Figure 3: Data Model of the UBI-REGI

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 the 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 the temperature. The associated keywords are temperature, air temperature, room air temperature, ambient temperature).

The parameters are managed by the PARAMETER table with 6 columns. A primary key is composed of a resource ID, an operation name and parameter name. As for the attributes, there are 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.

3.6 UBI-REGI API

We then define the UBI-REGI API `getService()` so that external applications finds the service operations easily. The API takes queries of the service category, the resource class, the physical location and the purpose keyword. The API returns a list of service operations matching these queries.

```
* 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
```

4. CONCLUSION

In this paper, we have presented a novel service registry called *UBI-REGI*, in order to support efficient and dynamic discovery of service resources in the ubiquitous network.

As for the future work, we plan to elaborate the definition of resource class to improve the quality of resource search. Also, we will evaluate the effectiveness of the UBI-REGI through other applications, including the ubiquitous cloud. Finally, it is interesting to introduce the ontology in the purpose keywords, which enables semantic-based service discovery.

5. ACKNOWLEDGMENTS

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (B) (No. 23300009), Young Scientists (B) (No. 21700077), Research Activity Start-up (No. 22800042)], and Hyogo Science and Technology Association.

6. REFERENCES

[1] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[2] Uddi. `http://uddi.xml.org/`.

[3] Ichiro Satoh. A programming model for smart spaces (ubiquitous computing). *Information Processing Society of Japan*, 45(12):2655–2665, 2004.

[4] Kouichi Egami, Hiroshi Igaki, and Masahide Nakamura. A standardized data model for networked appliances in home network system. Technical Report 11, 2009 (in Japanese).