2015 IEEE World Congress on Services

A Social Network based Approach for IoT Device Management and Service Composition

Guo Chen, Jiwei Huang*, Bo Cheng, Junliang Chen
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing 100876, China
Email: guochen09@gmail.com, {huangjw, chengbo, chjl}@bupt.edu.cn

***Abstract*—Nowadays, with the rapid development of hardware and network technologies, various types of smart devices are released by different vendors, resulting in the emergence of Internet of Things (IoT). However, most of the existing approaches for IoT device management are designed in a centralized way, whose efficiency meets challenges recently because of the large scale of heterogeneous device modules and highly dynamic essence of the networks. To tackle this challenge, we propose a distributed approach for IoT device management and service composition from a social network point of view. Specifically, we introduce RESTful web service to encapsulate heterogeneous IoT device modules, providing uniform interfaces for IoT service invocation. According to the relationships between IoT devices, social network theory is applied to model IoT services in different dimensions. After fully considering the social properties, a flexible and scalable schemes for IoT service composition is designed to satisfactorily meet users' requirements from several aspects. Finally, simulation experiments based on dataset from reality are conducted to validate the effectiveness of our approach.**

***Keywords*-Web service; Internet of Things; social network; device management; service composition.**

I. INTRODUCTION

The Internet of Things (IoT) is the interconnection of embedded computing and sensor devices within the existing Internet infrastructure. It is expected to offer advanced connectivity of devices, systems and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications [1]. With the rapid development of microchips, sensor devices, networks and software in recent years, IoT is becoming more and more popular in several places. An example is the industrial logistics system, where IoT has been widely applied in many aspects such as package tracing, delivery localization and vehicular monitor. Another scenario is smart home, which is one of the hottest applications in people's daily life. Taking advantages of IoT techniques, various home automation systems are changing people's life, and is perfectly used to control lighting, heating, ventilation, air conditioning, appliances, communication systems, entertainment and home security devices to improve convenience, comfort, energy efficiency, and security [2].

*Jiwei Huang is the corresponding author. Email: huangjw@bupt.edu.cn.

With the growing popularity of IoT, more and more smart products or devices are involved, resulting in the problem of heterogeneity among different devices, which brings more complexity to management in IoT. In order to improve the efficiency of IoT management, heterogeneity among different devices should be encapsulated. A widely adopted way is to encapsulate heterogeneous devices by web services, which regards all different components as uniform service modules with input and output interfaces [3]. With the popular RESTful style, all scoping information from/to IoT devices goes into the Uniform Resource Identifier (URI) [4], making it possible for the users or upper layer applications to access the devices via HTTP/HTTPS protocol in a unified and flexible way.

Besides heterogeneity, another challenge that IoT currently faces is the rapidly increasing number of devices. It has been reported that the Internet of Things (IoT) is forecasted to reach 26 billion installed units by 2020, up from 0.9 billion in 2009 [5]. Even though heterogeneous devices are encapsulated as homogeneous web services for better efficiency, the management of the services still needs to be treated accordingly, because of the following reasons. The traditional triangular SOA operational model is commonly used in most of service systems to manage web services, in which services are registered in a centralized Universal Description, Discovery, and Integration (UDDI) registry [3]. With exponentially increasing number of services added into the single UDDI registry, the efficiency of service management and service search decreases dramatically. Hence service management scheme should be designed from centralized to distributed, and thus brings about scalability and robustness. Also, according to the characteristics of distributed system, it should appear to the applications and users that the system with independent service management nodes is a single coherent one [6].

In order to meet users' demands, most of the devices in IoT are not working alone. Like relationships among people in real world, there are many types of relationships between IoT devices. We take a typical smart home scenario for example. TV and Xbox are often placed together in a living room, and thus their relationship is defined as "devices in the same place". Heating units in different rooms may sometimes have to work together to keep the

978-1-4673-7275-6/15 $31.00 © 2015 IEEE
DOI 10.1109/SERVICES.2015.9

entire house warm, and their relationship is denoted as "the same device type". Besides, another relationship is defined as "correlation of devices", which means different type of devices in different locations may also work cooperatively or sequentially, e.g., after garage doors are closed, lights in the living room may be turned on to welcome the hostess's arrival. With all these complex relationships, IoT devices reveal characteristics of modern social networks, resulting in a social structure of IoT [7]. And atomic IoT services sometimes have to composite into certain complex workflows to meet the requirements of upper layer applications. How to make full use of the social relationships between IoT devices to manage and composite IoT services remains unexplored.

To attack these challenges, we propose a social network based approach for IoT device management and service composition. More specifically, (1) we encapsulate heterogeneous IoT devices into web services. Using RESTful style, the users or upper layer applications are able to invoke heterogeneous devices in a homogeneous way, which makes it possible for the devices to collaborate with each other to form a comprehensive composite business process. (2) We model the relationships between IoT services by social networks, and classified them into three dimensions which are location, type and correlation. Schemes of service registration, selection and management are carefully designed for each of these three dimensions, and thus the social network can be divided into three separate subnetworks for efficient management and accelerated service search. (3) A mechanism for IoT service composition is proposed. Taking the social relationships between IoT services into account, it allows for the automatic collaboration of heterogeneous IoT devices in order to meet user requirements. Detailed designs and analysis will be presented in this paper, and the effectiveness of our approach is validated by simulation experiments based on real world dataset.

The remainder of this paper is organized as follows. Section II discusses the related work most pertinent to this paper. In Section III, the architecture for social network based IoT device management and service composition is proposed. More specifically, Section IV introduces the detailed mechanism of web service encapsulation for IoT devices; Section V describes the modeling of 3-dimensional social relationships between IoT services and designs detailed data structures and algorithms for service search in each of the dimensions; Section VI proposes schemes of service selection and composition. Simulation experiments are conducted to validate the efficacy of our approach in Section VII. Finally, we conclude the paper in Section VIII.

II. Related Work

A. Web Service based IoT Device Management

IoT network has been a hot topic recent years, and device management over IoT has aroused more and more concentrations. Many researches are now concerning on applying web service techniques to the management of IoT devices. Various approaches have been put forward, most of which regard all heterogeneous IoT devices as uniform service modules such that they can be accessed via common communication protocols by upper layer applications.

Spiess et al. [2] intended to apply service integrated methods pervasively to IoT scenarios in business environment. Liu et al. [8] proposed a web service based framework to manage heterogeneous computing resources coming from various different platforms and systems in an enterprise environment. However, relatively different from business environment, a common IoT network often contains devices that are quite heterogeneous, like those in a smart home. Thus these approaches may face challengers in dealing with heterogeneous devices.

In the seminal work by Son et al. [9], a complete solution has been advocated that web service can be used to encapsulate household smart devices like smart TV, and its efficiency can be guaranteed by constructing resource relationship graph. However, as the scale of IoT network keeps increasing rapidly, a more efficient approach for IoT service discovery, management and composition remains unexplored.

B. Applying Social Network in Web Service Search and Composition

With the increasing number of interactive web services appearing on the Internet, researches have applied social network model to web service search and composition. Asl et al. [10] conducted an experiment in which they recorded co-presence times of different devices in term of users' activity data that comes from an open data source, and they found that the small world phenomenon exists in an IoT network even if mobility of devices are taken into consideration. Their empirical study has provided a strong support for the application of social network to IoT service search and composition.

Atzori et al. [7] presented a novel paradigm of IoT organization in form of social network, and proposed a corresponding model. They also noted that service discovery and composition are also needed as basic functions for the development of IoT. However, they merely gave some simple and brief descriptions of details in the design.

On the other hand, Chen et al. [11] studied web service discovery based on the social relationships between service modules, mainly focusing on the satisfaction of web service quality. However, as for an IoT service, various requirements besides quality have to be considered, such as the correlation between two services and the location of the services. Besides, it is not quite appropriate to use quality only to judge whether a service is satisfactory or not.

III. Architecture

Comprehensively considering both IoT device management and service provision, we design an approach based

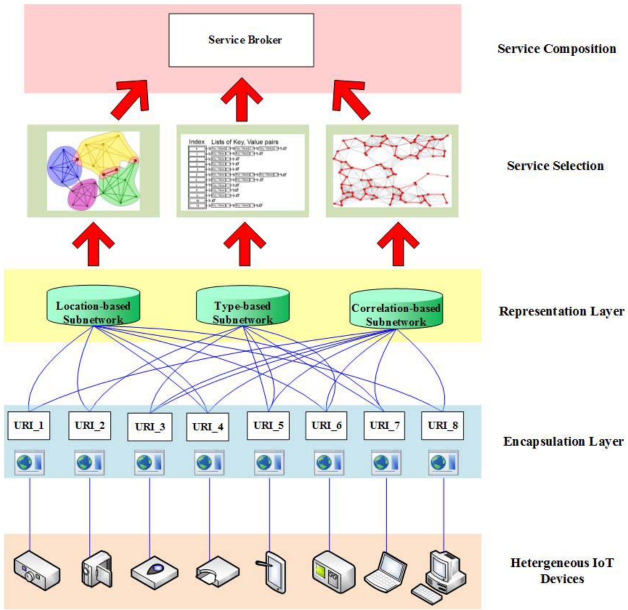

Figure 1. The architecture for IoT device management and service composition.

on social network models. In our approach, all heterogenous IoT devices are encapsulated by web services, transforming IoT device management problem to a homogenous service management problem. 3-dimensional social network models are presented to describe the relationships between IoT devices, and a parallel approach for IoT service search and composition is proposed. The architecture of our approach is illuminated by Fig. 1. According to the functionalities, our approach includes four layers, i.e., encapsulation layer, representation layer, service selection and service composition.

In order to encapsulate heterogeneity of different IoT devices, all IoT device modules are wrapped by RESTful web services once they access the system. With uniform interfaces to upper layer applications, the encapsulated IoT services can be controlled and monitored through Uniform Resource Identifier (URI). And thus the IoT devices can be managed in a convenient and efficient way. The encapsulation of IoT services will be described in detail in Section IV.

After encapsulation, we manage the IoT services in a parallel way, shown by the representation layer in Fig. 1. Since there are various relationships among different devices, a social network model is applied to represent and store such social relationships. Specifically, such model is denoted in three dimensions, representing three types of relationships between IoT devices, which are location, type and correlation. For different features in each dimension, different data structures are applied to denoting social network model in different dimensions. Details of the design for this layer can be found in Section V-A.

A service broker is responsible for searching for service nodes and making service composition to meet user demands. According to the multi-dimensional social network model, the searching processes are implemented in a parallel way. Based on the different data structures, searching for each dimension can be done with highly efficiency. Detailed algorithms can be found when we describe the service search processes in Section V-B.

A broker is responsible for aggregating the results of multi-dimensional service search at the service composition layer. The broker composites the services and finds a composite group which best meets the requirement. Like what our mechanism does to original IoT devices in the encapsulation layer, the broker finally generates a WSDL document describing such service composition for upper applications. We defer readers to Section VI for the detailed schemes of service composition in each broker.

IV. Encapsulation

There can be several types of devices in an IoT, resulting in the heterogeneity problem for data collection and device management. The heterogeneity comes from several aspects, such as different types, scales and values of data, different patterns and manners of the actions and management, etc. It appears to be extremely expensive and time-consuming to design and develop individual modules for each type of the devices. Hence, the heterogeneous IoT devices should be encapsulated as uniform interfaces.

Web services are applied for encapsulation, which encapsulate the functionality implementation details of the IoT devices and can only be accessed through their published uniform interfaces [3]. As for guidelines and practices for creating scalable IoT services, we use Representational State Transfer (REST) style [12] for a light-weight encapsulation. There are several advantages of such an approach. First, the heterogeneous IoT device management is transformed into homogeneous service management, which has been studied for years. The IoT device modules can be maintained and searched via URI, and invoked via HTTP/HTTPS, with highly efficiency and convenience. Second, the data generated by IoT devices can be described by a standard XML configuration file, and different data structure will bring no extra workload to the design and implementation of our system architecture. Third, upon arrivals of new IoT devices, only the communication protocols need to be implemented and thus the devices can be operated and managed in a uniform manner. The light-weight implementation of REST style can further guarantee the scalability in performance aspect.

Fig. 2 shows an example of the IoT device encapsulation in a smart home environment. A temperature sensor is encapsulated by a web service which is described by an XML configuration file. The basic data for interaction includes the sensor ID, sensor type, the coordinates of the sensor, and the value of temperature. The ID and type of the sensor are represented by two variables with string-type, and the

```xml
<types>
    <schema targetNamespace="http://example.com/stockquote.wsdl"
                xmlns="http://www.w3.org/2001/XMLSchema">
        <xs: element name="id" type="xsd:string"/>
        <xs: element name="type" type="xsd:string"/>
        <xs: element name="coordinate-X" type="xsd:float"/>
        <xs: element name="coordinate-Y" type="xsd:float"/>
        <xs: complexType name="location">
            <xs: element ref="tns:coordinate-X">
            <xs: element ref="tns:coordinate-Y">
        </xs:sequence>
        <xs: complexType name="TemperatureSensorInfo">
            <xs: element ref="tns:id">
            <xs: element ref="tns:type">
            <xs: element ref="tns:location">
        </xs:sequence>
    </xs:schema>
</types>
<message name = "getTemperatureRequest">
    <part name = "temperatureSensorInfo" type = "xs:TemperatureSensorInfo" />
</message>
<message name = "getTemperatureResponse">
    <part name = "temperatureValue" type = "xs:float"/>
</message>
<portType name = "temperatureSensor">
    <operation name = "getTemperature">
        <input message = "getTemperatureRequest"/>
        <output message = "getTemperatureResponse"/>
    </operation>
</portType>
```

Figure 2. An encapsulated temperature sensor management module.

coordinates are described by a pair of float variables. The ID, type and location constructs a complex data structure named *TemperatureSensorInfo* containing all the basic information of the sensor. The value of temperature is defined as a float variable. Upon arrival of an invoking request with input message “getTemperatureRequest”, the web service returns temperature value together with *TemperatureSensorInfo* in response. All the details of the data structures, including data definitions and invoking processes, are defined according to the XML configuration, which is strictly followed by both service invoker (user) and the service provider (sensor).

V. Service Representation and Service Selection

Instead of working alone, IoT devices often cooperate with each other in many application scenarios. Therefore, the relationships among them should be taken into consideration in the process of device management and process of service provision. In this section, a social network model is presented to represent and store such social relationships, based on which of novel service selection schemes, are proposed according to the structure of social networks.

A. Service Representation

To give a complete representation of various relationships between service nodes in the real world, the whole social network is classified into three dimensions, which are location, type and correlation, thus the social network can be divided into three separate subnetworks for efficient management and rapid service search. For every service node represented in the representation layer, it has to provide its basic information including location, type and correlation with other nodes, and a 3-dimensional social network will be built later then.

Not only does the social network model provide us with a profound understanding of the relationships between IoT services, but the model also makes it possible to accelerate the processes of service search. It has been proved that a fast search can be implemented by changing the storage structure of networks [13], and the time complexity would meet $O(\log n)$ if the nodes in a social network are organized in a hierarchy [14].

According to characteristics of different dimensions, the storage structure of these three subnetworks are designed separately and differently. As for the dimension of location, position information is represented by an aggregated data group containing the identity number as well as position coordinate value pairs of the service node (IoT device). In the dimension of type, we use a hash table in which device types are used as hash keys to store corresponding identity numbers of service nodes. And for the subnetwork in the correlation dimension, an undirected graph is applied in which the vertices represent the service nodes while the correlation among them are regarded as the edges connecting them. There is a weight assigned for each edge representing the probability that two connected service nodes cooperate with each other.

B. Service Selection

Based on the social network model, we design service selection schemes in each of these three dimensions. In this subsection, we present three service search algorithms working separately in the dimension of location, type and correlation, each of which returns local selection results to the service broker in a parallel way.

1) Location-Based Service Selection:

The IoT devices are geographically dispersed in different places. Without any additional supervised information, DBSCAN clustering algorithm [15] is applied on the coordinates of IoT devices to figure out devices in different places. Thereafter, service selection is conducted based on the classification results obtained by clustering. Such a procedure is shown by Algorithm 1.

Algorithm 1 Location-based service selection

Input: $loc, V, \epsilon, minPts$

Output: $\hat{C}$

$S = \{\langle sp_i, C_i\rangle\} \leftarrow \text{DBSCAN } (V, \epsilon, minPts)$

$\langle \hat{sp}, \hat{C}\rangle \leftarrow \underset{\langle sp, C\rangle \in S}{\arg\min} |loc, sp|$

return $\hat{C}$

In Algorithm 1, the inputs include the location information denoted as loc given by the service broker, the set of all services V, and two constant parameters ϵ and $minPts$ that are used in DBSCAN clustering. At first we apply DBSCAN algorithm [15] to make these uniform scattering points divided into different clusters, from which the complete set $S = \{\langle sp_i, C_i \rangle\}$ containing all clusters C_i and their respective starting point sp_i is returned. The algorithm then searches from S for the cluster $\hat{C}$ whose starting point $\hat{sp}$ is nearest to the location loc, and finally return $\hat{C}$ to the upper layer service brokers.

2) Type-Based Service Selection:

It is quite a common scenario where we need all devices of one type to work together at the same time, such as the control for heating systems or lighting systems in a smart home environment. A fast searching algorithm is designed for such type-based service selection, elucidated by Algorithm 2. A hash table is maintained, whose key is defined by the device type and value is denoted by all service entry identities of each type. The procedure of type-based service selection is to visit the hash table with the hash key $type$ (types of the devices) and finally obtain the set of all corresponding services of type $type$.

Algorithm 2 Type-based service selection

Input: $type$
Output: $V_{devices}$

$V_{devices} \leftarrow h(type)$
return $V_{devices}$

3) Correlation-Based Service Selection:

In some certain complex scenarios, heterogeneous devices in different places need to cooperate with each other to finish certain tasks. Such a requirement brings difficulty to service selection if we always want to find out those devices that are strongly associated with an assigned one. Therefore, we propose a scheme for such correlation-based service selection. In our approach, a positive value is assigned to each pair of services to describe the correlation between them. With a user-preferred threshold, all services with satisfied correlation value are selected and returned to the upper layer brokers.

In specific, correlation value between services s_i and s_j is defined as (1), where $N(s_i \cap s_j)$ indicates number of times that both s_i and s_j work cooperatively while $N(s_i \cap s_j)$ represents number of times when at least one of them is at work in all scenarios. Then we model the services correlations by an undirected graph $G = (V, E)$ where a vertex V represents all services while an edge E denotes the correlation. Based on the correlation value, we assign a weight w_{ij} to each edge (i, j) calculated by $w_{ij} = 1 - v_{ij}$. Initially, the weights can be set to 0, or determined by some prior knowledge on the correlation relationships. And they are automatically updated during service processes according to their definitions.

$$v_{ij} = \frac{N(s_i \cap s_j)}{N(s_i \cup s_j)}. \tag{1}$$

Algorithm 3 Correlation-based service selection

Input: V, E, sp, θ
Output: $\langle V_{st}, E_{st} \rangle$

$V_{st}^0 \leftarrow \{sp\}$, $E_{st}^0 \leftarrow \emptyset$
while $V_{st}^i \neq V_{st}^{i-1}$ **do**
  $\langle p, q \rangle \leftarrow \underset{p \in V_{st}^i, q \notin V_{st}^i}{\arg\min} \; w_{pq}$ s.t. $w_{pq} < \theta$
  $V_{st}^{i+1} \leftarrow V_{st}^i \cup \{q\}$
  $E_{st}^{i+1} \leftarrow E_{st}^i \cup \{(p, q)\}$
  $i \leftarrow i + 1$
end while
$V_{st} \leftarrow V_{st}^i$, $E_{st} \leftarrow E_{st}^i$
return $\langle V_{st}, E_{st} \rangle$

The procedure of correlation-based service selection is shown by Algorithm 3, which extracts a satisfactory subgraph of G from a given service trigger sp. The basic idea is to iteratively generate a spanning tree of G in which the weights of all edges are below a user-defined threshold θ. In i-th iteration, V_{st}^i and E_{st}^i represent the set of vertices and edges in the spanning tree respectively, and our algorithm expands the tree with the vertex not in V_{st}^i and its corresponding edge which has the smallest weight value but still less than the threshold θ. As V_{st}^i becomes stable, we obtain the spanning tree including all satisfactory service nodes.

VI. Service Composition

In order to make the IoT services cooperate with each other to provide complex functionalities, the service broker composites series of feasible services according to user requirements. In such a service composition process, the social relationships between services should be fully considered. After searching for the candidate services in each dimension, according to the procedures that have been introduced in the previous section, the broker collects all the results of searching at the service composition layer, and then selects proper services and aggregates them into a composite one.

Fig. 3 demonstrates the complete relationships between upper applications and the service broker as well as the responsibility range that service broker lies on. As is shown in the figure, just like how our mechanism deal with original IoT in the encapsulation layer, the service broker finally generates a WSDL document describing such service composition to upper applications.

Fig. 4 illustrates an example of service composition in a smart home. Three simple service nodes are involved in this example, i.e., a door lock, a lighting bulb and a heating unit. This WSDL configuration file describes not only basic information of these service nodes such as the interfaces, input and output parameters, but also how these services cooperate with each other to form a service composition. It describes such an application scenario in which a hostess opens the door of the living room after she comes back home; and then the unlocking action of the door lock triggers

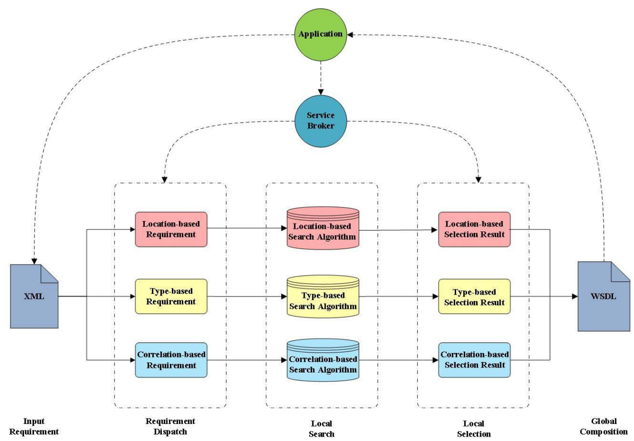

Figure 3. The procedures of service selection and service composition.

two other services which turns on the bulb and the heater respectively.

VII. Evaluation

In this section, we present an empirical evaluation of our approach. Based on both IoT dataset from real world and simulations, we conduct experiments to validate the effectiveness as well as efficiency of service search and composition.

A. Dataset Description

In our experiment, we adopt an open-source IoT dataset from a real-world smart home environment [16]. It consists of several types of sensors/devices whose invoking data for nearly 2 weeks is recorded. The dataset contains basic information such as the types, locations and cooperation relationships of IoT devices in different scenarios. To give a brief but complete description of service search and composition process and time complexity analysis of the algorithms in our approach, we pick up the dataset from a dinner preparation scenario to conduct our experiments.

B. Functionality Validation

1) Location-Based Service Selection:

In the dinning preparation scenario, a hostess may need to use some household cookers in dining room and kitchen. To complete such procedures, IoT services located in the dining room and kitchen should be selected. From the dataset, we obtain the ranges of coordinates for each room in the house shown as Fig. 5. Since the concrete coordinates are not provided, we generate them uniformly randomly according to the given ranges.

Based on the coordinates, we apply our algorithm of location-based service selection. The parameters ϵ is set to be 1.5 while $MinPts$ to be 2 for DBSCAN algorithm. The clustering result is shown by Fig. 6, where devices in different rooms are represented as triangles with different colors.

```
<process name="doorwayCooperationService"
        targetNamespace="http://packtpub.com/bpel/doorway"
        xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
        xmlns:lock="http://IoTSN.org/lock"
        xmlns:light="http://IoTSN.org/light"
        xmlns:heating="http://IoTSN.org/heating" >

    <partnerLinks>
        <partnerLink name="DoorUnlock"
                partnerLinkType="lock:DoorLockLT"
                myRole="DoorLockManagementService"/>

        <partnerLink name="LightOn"
                partnerLinkType="light:LightLT"
            myRole="LightManagementService"
                partnerRole="LightManagementRequestor"/>

        <partnerLink name="HeatingOn"
                partnerLinkType="heat:HeatingLT"
          myRole="HeatingManagementService"
                partnerRole="HeatingManagementRequestor"/>
    </partnerLinks>

    <variables>
    <variable name="DoorUnlockRequest"
                messageType="lock:DoorUnlockRequestMessage"/>
        <variable name="DoorUnlockResponse"
                messageType="lock:DoorUnlockResponseMessage"/>
        <variable name="LightOnRequest"
                messageType="light:LightOnRequestMessage"/>
        <variable name="LightOnResponse"
                messageType="light:LightOnResponseMessage"/>
        <variable name="HeatingOnRequest"
                messageType="heat:HeatingOnRequestMessage"/>
        <variable name="HeatingOnResponse"
                messageType="heat:HeatingOnResponseMessage"/>
    </variables>

    <flow>
        <sequence>
            <receive partnerLink="DoorUnlock"
                portType="lock:LockManagementPT"
                operation="doorUnlock"
                variable="DoorUnlockRequestMessage"/>
             <invoke partnerLink="LightOn"
                  portType="light:LightManagementPT"
                  operation="lightOn"
                  inputVariable="LightOnRequestMessage"
            outputVariable="LightOnResponseMessage"/>

             <invoke partnerLink="HeatingOn"
                  portType="heat:HeatingManagementPT"
                  operation="heatingOn"
                  inputVariable="HeatingOnRequestMessage"
            outputVariable="HeatingOnResponseMessage"/>

             <reply partnerLink="DoorUnlock"
                 portType="lock:LockManagementPT"
                 operation="doorUnlock"
                 variable="DoorUnlockResponseMessage"/>
        </sequence>
    </flow>
</process>
```

Figure 4. An example of WSDL configuration file for service composition in a smart home senario.

Type of Room	Range in Coordinate
Office Study	(0,0) to (3,4)
Bedroom	(3,0) to (7,4)
Living Room	(3,4) to (9,10)
Foyer	(1,10) to (3,13)
Bathroom	(5,10) to (9,13)
Dining Room	(3,13) to (9,17)
Kitchen	(5,17) to (9,20)

Figure 5. The range of coordinates for each room in the house.

2) Type-Based Service Selection:

When preparing for dinner, various types of household cookers are selected. Therefore, we deploy type-based search algorithm to find proper services. In this scenario, we choose five types of devices in our experiment, including light switch, freezer, refrigerator, microwave and toaster. As is

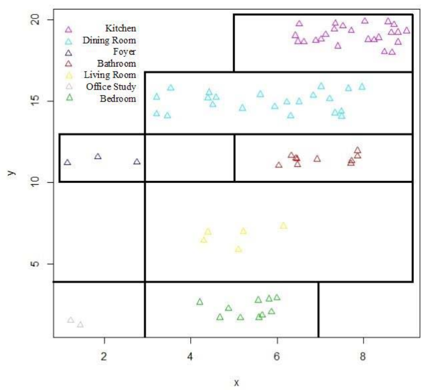

Figure 6. The results of location-based service searching.

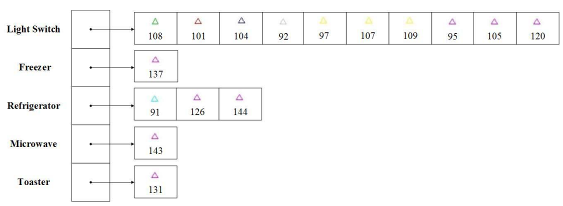

Figure 7. The results of type-based service searching.

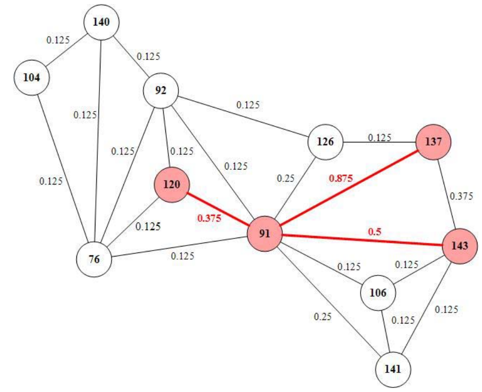

Figure 8. The results of correlation-based service searching.

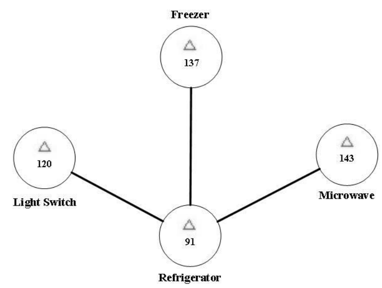

Figure 9. The result of service composition.

shown in Fig. 7, the final results are represented by a hash table with linked lists to store all possible devices of the same type. The locations are represented with different colors while the ID of each node is also recorded in the figure.

3) Correlation-Based Service Selection:

In this scenario, different service nodes like refrigerator, oven, microwave and freezer need to work together to help hostess prepare dinner. Based on the invoking records in the dataset, we start our correlation-based service selection algorithm with a light switch service in the kitchen as a trigger, as our objective, to search for the services with correlation value bigger than 0.3. The results are shown by Fig. 8, where the undirect graph describes the subnetwork in correlation dimension. The spanning tree generated by our approach is marked in red, which shows the effectiveness of our algorithm.

4) Service Composition:

In order to make the IoT services selected from the three dimensions to assist hostess prepare dinner, the service broker composites these available services according to the user requirements. At first, we select those in the intersection of these three IoT service node sets, which contains light switch, freezer, refrigerator, and microwave. All these nodes locate in the kitchen. Then these services are composited, which is shown in Fig. 9.

In this figure we can see that the selected service nodes with ID number and location information, which is represented by purple triangles indicating they are in kitchen, are connected with edges representing their cooperation. Under this condition, a service usage procedure is likely to happen: as hostess turns on the light in the kitchen to prepare for dinner, this action triggers refrigerator back to work from sleep, invoking other IoT services to make freezer and microwave adjust their own working mode in advance such that the hostess could prepare for a meal quickly and efficiently.

C. Performance Evaluation

In order to show efficiency of our search algorithms, we conduct simulation experiments varying the amount of IoT devices from 100 to 1200. Algorithms for all three dimensions are evaluated, and their performance (i.e., running time) is shown in Fig 10.

The results indicate that with the growth of search space, the increasing speed of running time is much slower than exponential alternative. Besides, since hash table is used in type-based searching, the time complexity is $O(1)$. The results also show that our searching algorithms can be applied to service selection with high efficiency.

VIII. Conclusion

In this paper, we propose a new approach for IoT device management and service composition. IoT devices are encapsulated by RESTful web services, and a 3-dimensional

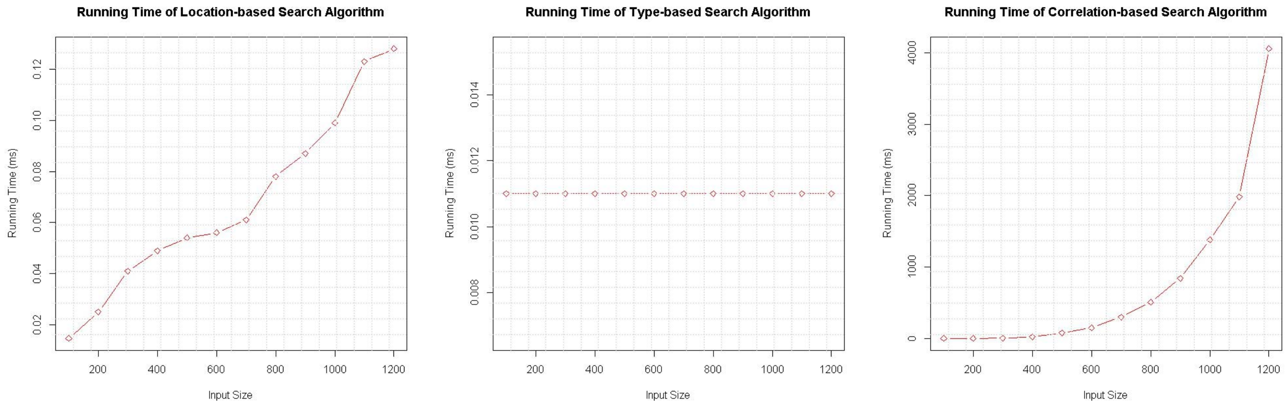

Figure 10. Performance of service selection algorithms.

social network model is applied to describe the relationships between devices/services. Data structures as well as algorithms are designed for each dimension of the social network, and the algorithms run in parallel for service search and selection. Furthermore, we propose a service composition scheme in order to make the IoT devices collaborate with each other and implement complex functions. We hope our work would offer reference value for the understanding of relationships between IoT devices and provide an efficient approach for IoT device management and service composition.

ACKNOWLEDGMENT

This work is supported by the National Grand Fundamental Research 973 Program of China (No. 2012CB315802), National High-tech R&D Program of China (863 Program, No. 2013AA102301), National Natural Science Foundation of China (No. 61132001), Program for New Century Excellent Talents in University (No. NCET-11-0592), and the Fundamental Research Funds for the Central Universities (No. 2015RC22).

REFERENCES

[1] J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014.

[2] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, and V. Trifa, "SOA-based integration of the Internet of Things in enterprise services," in *IEEE International Conference on Web Services (ICWS 2009)*, Jul. 2009, pp. 968–975.

[3] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*. Springer Berlin Heidelberg, 2007.

[4] L. Richardson and S. Ruby, *RESTful web services*. " O'Reilly Media, Inc.", 2008.

[5] L. G. Janessa Rivera. (2014) Gartner says a thirty-fold increase in internet-connected physical devices by 2020 will significantly alter how the supply chain operates. [Online]. Available: http://www.gartner.com/newsroom/id/2688717

[6] M. v. S. Andrew S. Tanenbaum, *Distributed Systems Principles and Paradigms*. Pearson Prentice Hall, 2007.

[7] L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a social structure to the Internet of Things," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1193–1195, Nov. 2011.

[8] L. Liu, M. Gaedke, and A. Koeppel, "M2M interface: a web services-based framework for federated enterprise management," in *IEEE International Conference on Web Services (ICWS 2005)*, Jul. 2005, pp. 767–774.

[9] J.-Y. Son, J.-H. Park, K.-D. Moon, and Y.-H. Lee, "Resource-aware smart home management system by constructing resource relation graph," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1112–1119, Aug. 2011.

[10] H. Asl, A. Iera, L. Atzori, and G. Morabito, "How often social objects meet each other? analysis of the properties of a social network of iot devices based on real data," in *IEEE Global Communications Conference (GLOBECOM 2013)*, Dec. 2013, pp. 2804–2809.

[11] W. Chen, I. Paik, and P. Hung, "Constructing a global social service network for better quality of web service discovery," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1–14, 2014.

[12] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," in *Proceedings of the 22nd International Conference on Software Engineering (ICSE 2010)*, 2000, pp. 407–416.

[13] J. Kleinberg, "Small-world phenomena and the dynamics of information," *Advances in neural information processing systems*, vol. 1, pp. 431–438, 2002.

[14] D. J. Watts, P. S. Dodds, and M. E. Newman, "Identity and search in social networks," *Science*, vol. 296, no. 5571, pp. 1302–1305, 2002.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996)*, 1996, pp. 226–231.

[16] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, vol. 3001, pp. 158–175.