2014 IEEE 38th Annual International Computers, Software and Applications Conference Workshops

The Application of Internet of Things in Social Network

Yu Zhang and Jiangtao Wen
Department of Computer Science and Technology
Tsinghua University
Beijing, China
yuzhangiot@gmail.com, jtwen@tsinghua.edu.cn

Fan Mo
School of Journalism and Communication
Lanzhou University of Finance and Economics
Lanzhou, China
mof1985@126.com

Abstract*—the combination of Internet of things and the social network make it possible for us to interact with environment and other people; this leading to the social Internet of things concept. In order to expand the idea of human to human connectivity to include everything in the real world, a suitable social model acts as a key factor to expand basic communication functionalities to include services discovery, push messages, subscribe channels and execute orders. Existing solutions do not mention about the relationships between entities or limit the relationships within the physical entities range. In this paper we propose a social model of Internet of things to organize everything in the physical world including both entities and human. And a smart environment prototype including sensors, hardware middleware, cloud server, social network platform, and mobile clients along with service discovery functions and reminding functions.

***Keywords*—Internet of things; social network; object oriented; IFTTT**

I. Introduction

Internet of things (IoT) is now becoming a new revolution after Internet in IT domain. It is the combination of Internet technology and machinery industry. The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc. – which through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals[1]. There are many intersections between IoT and other technologies including SCM, sensor, communication, cloud compute and storage, data visualization and data mining, and it is widely used in logistics and warehousing, Health and medical, intelligence and social environment [2].

Owning to the rapid development of IoT, many companies and developers devote to seek out methods that make the IoT more universal. But for years, IoT is staying in the conceptual level. On the contrary, recently, the development of social network is amazing. The total number of registered users of Facebook increased to 1 million in only one year since it was founded [3]. By the end of May 2012, the total number has increased to 900 million [4]. The Sina Weibo is another example. The total number of registered users of Sina Weibo reached to 500 million from 2009 to 2012 in China. What's more, it had 46. 2 million active daily users who send over 1000 million tweet through Sina Weibo platform each day [5]. With social network, people can share their status and moods to their friends everywhere and any times through just a little click. Social network users use messages, pictures, videos and websites to share contents like travel, food and fun things. However, there are only computers, mobile phones and tablets can use social networks now. So if smart devices can involve in social network, and they also can send tweets to social network as human beings do, then Internet of things can be well connected with social network.

Recently, people made an effect in finding the ways to combine IoT equipment with the social network. For instance, Twitter developed a whimsical tweet-enabled cuckoo clock called #Flock in 2003, whose shape is similar to a clock. Its main function is to remind users of new tweets or to send simple tweets through animating small wooden puppets [14]. Reference [6] worked out an application of the Internet of things in social network that includes a smart device that can push its status messages in twitter. Reference [7] put forward a human relationship architecture that is based on social network. He also put forward a Ubiquitous IoT architecture. The author of [8] studied the combination of social network and technical network, and discussed about the implication of socio-technical networks in the context of the Internet of things. Reference [13] analyzed the conception of the human social network and used the method of analogism to create the object social network. He also put forward methods to classify objects by manufacturing time, manufacturer, owner and location in social-IoT.

Last but not least, the work in [8] considered that the status and motion of an object can be sensed through RFID as well as sensors in a specific context. Take for example a cup, we can recognize it as a coffee cup through its RFID device. We also know it had been lifted slowly through a 3-axis sensor, and through the value of the temperature sensor on the cup we can get the message like “liquid in the cup is too hot to drink”. What's more, we can send that message on twitter through middleware, but the author didn't mention the specific method about the integration of technical network and social network. Regarding the organization method of objects in social-IoT and how to build an architecture of social-IoT, but few researches work addressed this challenge. Some work [8] investigated on the potential of combining social and technical networks but they didn’t mention about the relationships between entities.

978-1-4799-3578-9/14 $31.00 © 2014 IEEE
DOI 10.1109/COMPSACW.2014.41

Others [13] limit the relationships in social-IoT between entities in order to establish a social network of things.

II. Object Oriented Method in Social-IoT

A. Basic Application

The object of the combination of social network and internet of things is to make the things smart which can reduce the difficulty of using the internet of things by people. But the real world is so complicated that we cannot organize objects like people as in social network. Therefore, instead of traditional method, we use object oriented method to organize everything in the social-IoT. One advantage of object oriented method is using the way human beings think to abstract and cognize the real world, abstracting real substances into objects which are the capsule of both attributes and operations[16]. From a building to a potted of flower in the building are both object, and their status are attributes, as well as their actions are operations. For example, there is a potted of flower with several sensors (soil moisture sensor, pH sensor and light sensor) to monitor its health and a water supply device which is controlled by an electric relay. Then attributes of this object include the soil moisture, soil pH value and intensity of illumination, operation is watering. If there is a same attribute in different objects, then they can be abstracted into a class which is the abstraction of objects. For instance, bracket plant, rose, chrysanthemum share the same class “flower” which has basic attributes and operations of flower.

Real world entities cannot connect the social network without the help of internet of things (e.g. sensors, WSN, sensor cloud). Besides, these entities have to be organized by object oriented methods so that they could be recognized and communicate with each other on the social network. There are four features in object oriented methods. They are encapsulation, abstraction, inheritance and polymorphism.

Encapsulation is especially fit for the objects in social-IoT, because users don’t need to know the specific implementation of a function. Instead, they only need to know what these objects can do for them. For example, there is an object called “PM2.5 sensor in Beijing” on the Facebook. We can get the current value of PM2.5 by send a simple message like “What is the PM2.5 value now?” to that object on the Facebook.

Object's uniqueness are not affected by what sensors are equipped on them, even if these sensors are of the same type and manufactured in the same period. There need a set of standards to manage objects in social-IoT. Every object belongs to a class which contains many features and attributes. For instance, an office front door is the instance of class door which contains functions like check the status of the door, unlock/lock the door and take photos.

Inheritance enables you to extend the functionality of an existing class. Because we cannot define all types of entities in the real world, this feature enable users to define their own class which is inherited from parent class in the standard class library. Taking the class door above as an example, it contains several features like “The status of the door (open or closed)”,”If there is someone knocking the door”, “Camera to take photos” . So both front door and back door will inherit these features. If the front door doesn't equip a camera sensor, it also inherit the operation of “take photos”, the only different is that you can't execute it. Whereas if we setup a device which can speak to the people in the front door, what we need to do is just define a new class which is inherited from class door, and this operation will be add as an item in its operation list.

Just as a person assumes different roles in different scenarios, one operation may act differently in different context. Therefore, user need to redefine a function of a social-IoT class with different content. For instance, an operation (i.e. Send status to social network) for the office node may a send message (i.e. air temperature is XX, air moisture is XX, PM2. 5 is XX and it's time to open/close the window) to the social network, but the message for the flower node may like: “the soil moisture is XX, time to water”. In addition, polymorphism can also reflect on the different reactions when receiving the same command. Take operation “switch off lights” as an example. When it as the operation of the lamp node, the implementation may tune off one lamp; while it as the operation of the office node, this operation will iterate through the list of all the nodes labeled light under the office node, then read status to judge if the light is on, then invokes switch off operation on these node which is not off to complete the operation. All the progresses are invisible to users who just need to know there is an operation and how to control it.

B. Organization

1) Define a reasonable class library

In order to make entities be recognized and communicate with each other like human beings on the social network, we have to define the basic structure and reasonable class library for them.

People need to know what is a social-IoT object and how could it serve for them. Therefore, each social-IoT object is consisted of attributes and operations.

Attributions and operations are basic properties of an entity as a social network object. Reference [13] introduces 7 factors in social-IoT. They are ID management, owner control (OC), ship management (RM), service discovery (SD), service composition (CS) and trustworthiness management(TM). The basic idea is to create a “social network” whose protagonists are not people but substances. In this paper, we think it is unnecessary to copy everything in human beings social network. Human are always the protagonist. So the social-IoT should serves as a convenience to human beings other than a pure imitation. For example, there is no obvious benefit for us if two sensors talked about the basketball match last night.

Consequently, an object in social-IoT is expected to contain following basic attributes and operations:

a) ID

An object's ID is used as an identity in a specific context. Though each object is unique, just like names, they are similar to people that their name may be the same with each other. In which case, we can distinguish them in the context. For instance, a lamp in the office is definitely not the lamp at home. In this paper, we think it is not possible nor necessary to assign a unique ID to each object.

b) Location

Both people and things depend on their own geographical location to carry out activities. In social-IoT, this feature is especially useful. Imagining a user in a complete unfamiliar place, objects' location attributes is necessary if he want to know what services in IoT network are available.

c) Owner

There's one or more owners for every object in social-IoT, but owner are not always a physical person. For example, objects in the office are owned by a company, objects in public facilities are owned by a state-run agency. This feature also can establish a connection between social-IoT and human social network. For instance, a user who is nurturing bracket plant can search same plants in his community to find out the owner information (only when they are published) of these plants and communicate with them through social network, or he can add them in his society of friends (SoF) to set up a community of amateur.

d) Authority

Different user groups have different security levels: whether the object is visible, whether the object is editable, whether the service is available and so on.

e) Service Type

One of the greatest advantage of social-IoT is to make discovery of service easier for users. Service types are labels about what services the object can offer. That is to say, users can find out which service these object nodes can provide to do for him through these labels. For example, when a man just arrives in a strange parking lot, he can get parking sensor data through service discovery function and finds out which parking lot is free as soon as he views these message. What's more, this feature enable these entitles to automatically discover services like human beings to achieve some complicate jobs.

f) trustworthiness

Trustworthiness is a very important feature that people or entities will use it as a reference to sort the rank of usable service on the social-IoT. Every success service can increase the credit. On the contrary, some impropriate services or mismatch of service type and actual effect will decrease the credit.

2) Manage instances

Similar to human beings, there are many relationships between substances so that they can be related to each other on the social-IoT. Reference [13] introduces 5 human relation models: parental object relationship, Co-location object relationship, Co-work object relationship, Social object relationship and Owner object relationship. Besides, they created an object social network model and assumed there are 4 relationships (communal sharing, equality matching, authority ranking, market pricing) between objects. All relation models and relationships discuss above can be well organized by node method.

a) Parental object relationship, Co-location object relationship and Co-work object relationship

Relationships in this group can be managed by tree node method.

Parental object relationship is a typical one that we can manage it with tree node method. There are two types of nodes in the social-IoT. They are parent nodes and leaf nodes. Group nodes can add new nodes on it, like the office node above; leaf nodes are the end of a tree which cannot add any node (e.g. the flower node above).

Because each object in social-IoT has the location property, user can classify objects based on location information. And these location information can be organized in parental relationship way (i.e. first floor is the child node of an office building).

The essence of Co-work object relationship is parental relationship. Because the key word of this relationship is "service type", different device can achieve different jobs, but through co-work relationship, they could do more complex tasks. For instance, user can get only the value of water temperature and coffee machine status from single device, but the combination of these object can make coffee for users and inform them when coffee is ready. The diagram is shown in figure 1.

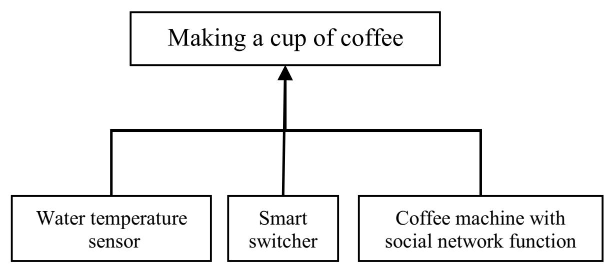


Fig. 1. Using tree node method to orgnize the co-work object relationship

b) Social object relationship

Human being social network has many sophisticated social object relationship models. Social-IoT is the combination of both social network and internet of things. Therefore, it is not necessary to import new models to manage the relationship between objects in social-IoT. Besides, this model is specially designed for humans in social network. And this it is not fit for substances in social-IoT.

Web node is the most frequently used method in the management of social relationship. As shown in figure 2, web node is an appropriate method for this kind of sophisticated relationship.

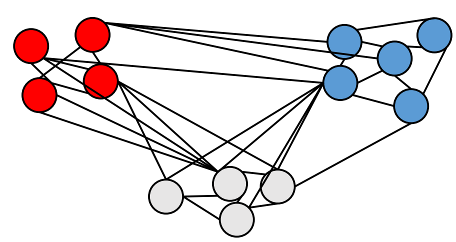


Fig. 2. Using web node method to orgnize the social object relationship

c) Owner object relationship.

The key word of this relationship is “owner”. Therefore, there need to use both tree node method and web node method to manage this relationship. Formal method is used to organize the relationship between owner and objects, and latter method is used to organize the relationships between owners.

III. THE SMART OFFICE BASED ON ANDROID PLATFORM

In order to verify the method above, we built up a smart office environment. In this test, we don't use sensors, service or owner as the key to organize objects. Instead, we use the most natural key “location” to organize objects. In this method, objects are organized by their location information, and other additional values (e.g. the value of sensors, who is the owner, service type) are just attributes of objects. Services are treated as their operations. For instance, both node IDs (i.e. manual input) and sensor values (i.e. automatic input) are attributes of this node. What's more, all objects under plants node inheritance the attributes (e.g. air temperature, soil moisture) from the class plants.

In figure 3, because we organize these object base on the location information, the office node is on the top of the whole architecture (i.e. All other objects are in the office). We setup a DHT11 temperature and humidity sensor to add attributes (i.e. Air temperature, humidity) into the attribute list of the office node. Besides, we define attributes as follows: ID --“B03 office”, location -- “B03 Tsinghua university, Peking, China”, owner -- “media lab”, service type -- “View temperature”, “View humidity”, ”send status on social network”, operation -- “Send status on social network”. The difference between the service type and the operation is that the former is a label for users to view and the latter is a function to operate. Under office node, besides, there are 4 class nodes: door node, electric appliance node, furniture node, plants node. For the experiment purpose, we only instanced two class nodes: one is a lamp node under electric node, the other is a bracket plant node under plants node. Lamp node include a light sensor and an electric relay to control the switch, while a soil moisture sensor are installed on the bracket plant node and it can send back the value of soil moisture per 15 seconds. Both of them communicate with service cloud via wifi. In addition, because both of them are inherited from office node, they own all the attributes and operations of the nodes (e.g. ID, owner, send status on WeChat).

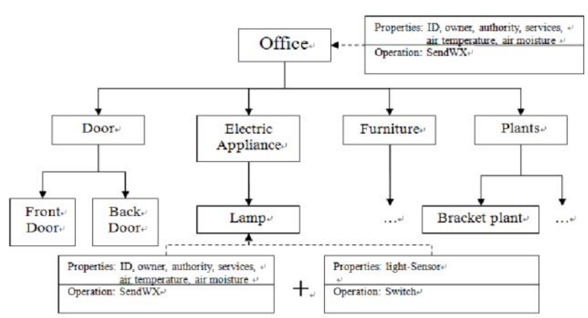


Fig. 3. Smart office structure

A. Hardware

Owing to the various types of sensors, we need a middleware to uniform output data, so we use arduino as the hardware middleware to connect soil moisture sensor, light sensor and electric relay and so on. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.

DHT11 sensor is needed to provide air temperature and air humidity data to the office node, we use Arduino pro mini as mother board which is shown in figure 4a, then connect the DHT11's data interface to digital interface of Arduino pro mini to upload environment data. In figure 4b the output signal type of soil moisture sensor is analog, so we connect its data interface to analog interface of Arduino pro mini. In order to control the lamp, we have to cut off the live wire of lamp and connect one end to NC contactor while another end to middle contactor of electric relay which is shown in figure 4c.

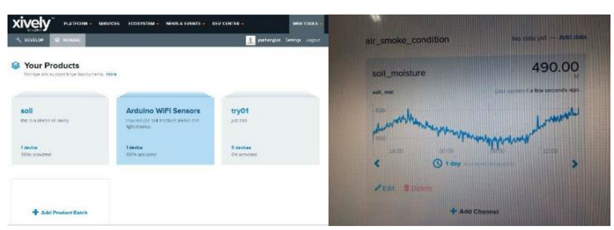
(a) Arduino pro mini (b) Soil moisture sensor (c) Electric relay

Fig. 4. Hardware connection

After stored the sensor data in hardware, we have to upload them to cloud server via Internet. Therefore, we need an arduino wifi shield to exchange data with Internet via wifi connection.

In this paper, we use Xively [18] as the cloud server. It is a real time network central which has a broad coverage of objects and the environment, using the standard REST protocol.

We use REST method which is based on http protocol to exchange data between hardware and cloud. Data structure is defined as “feed” which contains 3 factors: device ID, attribute name and number. Take for example the soil moisture sensor, the URL send to cloud server is: “http://api. xively.com/v2/feeds/device_ID/datastreams/soil_moisture”. Then the server will return TRUE if device_ID and channel soil_moisture exist. After that, structure “feed” was send to server which contain many factors besides current soil moisture value.



(a) Device on the cloud (b) soil moisture

Fig. 5. Cloud server

B. Cloud Server

Although the object oriented organize method make the social-IoT easier for users, its internal management mechanism is strictly based on one device corresponding to the device ID. Because our project has 3 Arduino equipment (One arduino Pro mini with temperature and humidity sensor, one of the arduino UNOs is used on a plant and the other is used to control the lamp). Therefore, we created three equipment in Xively (figure 5a). Curve in figure 5b is sampling results of soil moisture value per 15 second. In addition, each device is identified through the device ID and Xively Key.

C. Mobile Platform

In this test, we use Android equipment as the user terminal. The advantage of using Android OS is that its developers can leverage existing data and services provided by other applications while still giving the impression of a single, seamless application [10]. We developed our own APP platform to integrate WeChat [20] API and other applications, users only need to install one APP with everything integrated in it. Key technologies include: define data structure, exchange data with the cloud server, send status of objects on social network, service discovery and reminding function.

1) Define Data Structure

All nodes have to include basic factors as we discussed in 2.4 on page 4. Moreover, there are many attributes and operations in each node. Some of them are inherited from the parent node and others are of their own. In order to understand it better, we use the structure of office and bracket plant node as an example to illustrate the data structure.

As the top node of the whole system, office node needs only a few attributes. Structure code is shown in listing1:

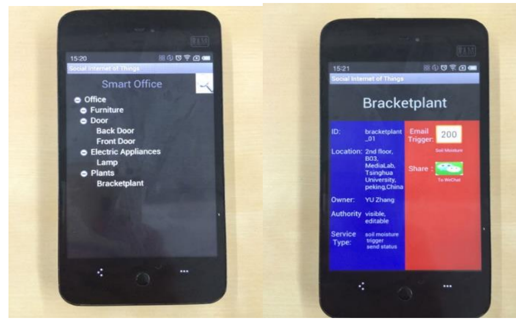
LIST I. OFFICE NODE STRUCTURE

```
structure SIOT_officebuilding_object
{
   String ID;
   public double air_temperature, air_moisture;
    protect String[] owner = new String[5];
    protect ArrayList service_type = new
ArrayList();
    bool isVisable, isEditable ;
    bool isLeafNode;
    sendWX();
}
```

As the leaf node, bracketplant node contains its own unique attributes as well as all properties and operations inherited from office node (Illustrated in listing2).

LIST II. BRACKETPLANT NODE STRUCTURE

```
public class SIOT_plants_bracketplant_object ::
SIOT_plants_object
   {
    public double soil_moisture;
    sendWX();
    trigger();
   }
```



(a) Tree list page **(b) Bracket plant page**

Fig. 6. Android APP

The tree list effect are shown in figure 6a. When an entity object is selected by the user, corresponding page is shown in figure 6b.

2) Exchange Data with the Cloud Server

In the first step, Android APP need to read sensor values from the cloud server before display them. We use HTTP protocol that follows REST principle to get data from the cloud server. In the next step, the cloud server have to authenticate users to fetch sensor values. Each user has a unique key ID which can be sent to cloud server through function "setApiKey" to create connection. Once the connection is successfully established, Android APP can send device ID, attribute name, time to cloud server to fetch corresponding data. In this test, we use Json format to transfer data for high efficiency (shown in listing3).

LIST III. RECEIVED JSON DATA


```
{
 "id":"soil_moisture",
 "current_value":"624. 00",
 "at":"2013-11-06T09:38:50. 747375Z",
 "max_value":"663. 0",
 "min_value":"0. 0",
 "tags":["soil_moi"],
 "unit":{"symbol":"M", "label":"Tsinghua"},
 "version":"1. 0. 0"
}
```


ID is attribute type, current value is the present soil moisture value.

3) Send Status of Objects on Social Network

WeChat platform is Chinese "twitter". Both of them are microblogging service which are a form of blogging that lets you write brief text updates (usually less than 200 characters) about your life on the go and send them to friends and interested observers via text messaging, instant messaging (IM), email or the web [12].

Substances on the social network "Wechat" can act like human beings, they can not only listen to the users' common, but also send their status.

In this test, we realize a function that can send status on social network "WeChat". Figure 7 shows the flow path of sending messages via WeChat API. Sharing operation can be triggered either by button click event or meeting certain condition event. For instance, when the value of soil moisture sensor is lower than a threshold, bracket plant will send status like: “I am thirsty, give me water!”

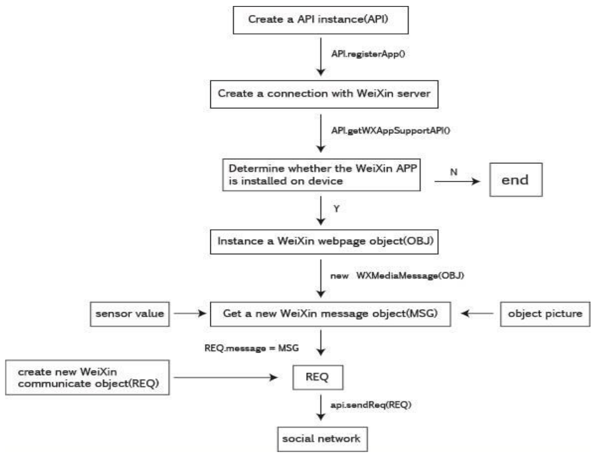
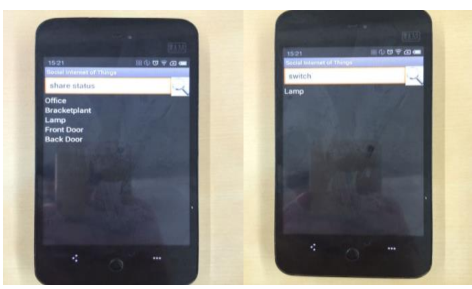


Fig. 7. Send status on WeChat

4) Service Discovery

A prerequisite for service discovery function is that each entity object in the social network must contain one or more service tags, because these tags are the keywords which can be used in retrieval services. What's more, for privacy reasons, we need to set access authority to these tags. For instance, front door and back door nodes are only visible to masters and the master authorized persons. In the contrary the government's public facilities are visible to everyone.



(a) Keyword is “share status” **(b) Keyword is “switch”**

Fig. 8. Service discovery

In figure 8 the keyword are “switch” and “share status”. Different lists correspond to different services. List items can be selected to show their properties and operations. For instance, lamp node can be turned on or off (showed in figure 8a) after you find its service via service search in figure 8b.

IV. Conclusion

In this paper we focused on the application of object oriented method in social-IoT, and we present a human being centered social-IoT construction method. Putting forward a natural way to manage entity object in social-IoT and create the architecture of network. Then we defined the relationships about node, class, entity object and owner. Moreover, we defined 6 basic attributes and 3 organize methods of entity objects which make it easier for objects and the real world entity correspond to each other as well as its service being conveniently found and operated. Then we introduced a smart office architecture which is based on this method, and this architecture fully comply with the object oriented requirements. In this architecture, not only users can view informations and search services, but also they can share status of an entity object and trigger events.

As part of future research, we will set up an outside entity object's organization which is more complicated and none-geographical related. We plan to research on a more universal model and then verify it in a few typical areas.

Acknowledgment

This work is supported by the National Science Fund for Distinguished Young Scholars of China (Grant No. 61125102), the State Key Program of National Natural Science of China (Grant No. 61133008) and State Support Program of National Science and Technology (Grant No. 2012BAK26B02)

References

[1] D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), The Internet of things, Springer, 2010. ISBN: 978-1-4419-1673-0.

[2] Luigi Atzori, Antonio Iera, Giacomo Morabito, The Internet of things: A survey, Computer Networks, 2010. ISBN：54 (2010) 2787–2805.

[3] http://zh. wikipedia.org/wiki/Facebook wikipedia

[4] http://baike. baidu.com/view/409608. htm baidu baike

[5] http://zh. wikipedia.org/wiki

[6] D. Guinard, M. Fischer, and V. Trifa, “Sharing using social networks in a composable web of things, ” in Proc.IEEE PERCOM, 2010.

[7] H. Ning and Z. Wang; , “Future Internet of things architecture: like mankind neural system or social organization framework?” IEEE Commun. Lett., vol. 15, no. 4, pp. 461–463, 2011.

[8] M. Kranz, L. Roalter, and F. Michahelles, “Things that Twitter: social networks and the Internet of things, ” in What can the Internet of things do for the Citizen (CIoT) Workshop at Pervasive, May 2010.

[9] http://arduino.cc/

[10] Chin E, Felt A P, Greenwood K, et al. Analyzing inter-application communication in Android[C]//Proceedings of the 9th international conference on Mobile systems, applications, and services. ACM, 2011: 239-252.

[11] Nurseitov N, Paulson M, Reynolds R, et al. Comparison of JSON and XML Data Interchange Formats: A Case Study[J]. CAINE, 2009, 2009: 157-162.

[12] J. Allen. Recognizing intentions from natural language utterances. Computational Models of Discourse, pages 107–166, 1983.

[13] Atzori L, Iera A, Morabito G, et al. The Social Internet of things (SIoT)–When social networks meet the Internet of things: Concept, architecture and network characterization[J]. Computer Networks, 2012.

[14] http://bergcloud. com/platform/casestudy/

[15] Coad P, Yourdon E. Object-oriented design[J]. 1991.

[16] http://baike.baidu. com/view/2814288. htm

[17] http://www.social-IoT.org/

[18] http://xively.com

[19] http://en.wikipedia.org/wiki/Webhook

[20] http://weixin.qq.com/