International Journal of Future Generation Communication and Networking
Vol. 9, No. 2 (2016), pp. 269-280
http://dx.doi.org/10.14257/ijfgcn.2016.9.2.27

WiFi Sensor Network Management Based On Runtime Model

Nan Ma [1], Yingxing Lin [1] and Xiang Zhou[2]

[1]*School of Economics and Management, Fuzhou University, Fujian, China. 350116*

[2]*Software Institute of Minjiang University, Fujian, China. 350108*

Abstract

The WiFi sensor network is a new technology for The Internet of Things, it main resolve the problem of informationize, it can collect thousands of sensor information by the sensors, and transform the information by WiFi sensor network. As for the function of WiFi sensor network, it must face to huge amounts of redundant data, in order to deal with the huge data, we have to code a lot of mapping code, it was a lot of time and space.In this article, we introduce a new method for WiFi sensor network management based on runtime model instead of coding. Our method have three steps: first, construct runtime model on WiFi sensor device; second, merge the distributed runtime model; last, convert the Composite Patterns Model to Application Scenario Model. The runtime composite model will help the WiFi sensor network to deal with huge amounts of redundant data. At the end of article, we apply our method on an application of enterprise management system, the results show that our method have high feasibility and effectively.

Keywords: *WiFi Sensor Network, Runtime Model, Composite patterns model, Application Scenario Model, management*

1. Introduction

With the rapid development of Internet of Things, a wide use of applications have been developed by the Internet of Things[1]. Internet of Things used information sensor devices to connect the devices which need to be monitored and managed, exploited an intelligent managing and monitoring network by message-switching technique[2]. Now, along with the development of WiFi, wireless sensor network(WSN)[3] is one of the hard-cores of Internet of Things, everyone can using WSN gather various data from environment and then transmit them onto Internet for processing, by using Internet, we can collect scattered data, unify management and control them, then complete the function of Internet of Things.

In fact, the Internet of Things is collecting the information in the objective world, then analyze and decide the collected data. From the implement of system point of view, application system need to use the sensor device to provide the management interface to get various information, and aim at the specific application scenarios to analyze and deal with this information. However, the present Internet of Things system development often used C/C++ bottom level development language to directly call the management interface of sensor devices, this method of coding has the advantage of adaptability, but with the high consumption of programming. The programmers need to be familiar with different sensor devices' management interface, so that implement the interactive of them, and build the application system based on them. Compared with the analysis and decision application system's management logic, these complex and trivial programming work is not the core of Internet of Things system, but in order to let the application system can normal and efficient running, the information obtained and data interaction bottom level programming development still need to spend programmers lots of time and energy.

ISSN: 2233-7857 IJFGCN

However, when WSNs gathered data from environment, they can only gather lots of structureless data in real-world for Internet of Things' calculation, these data need to be transfer and refer to opposite objective things in scenario[4]. For this reason, the construction of WiFi Loading Balance Management of Internet of Things is confronted with two challenge[5]:

(1) the diversity of device type, different devices have different ways to read and write data, and there must be different data format, this will lead to odds for information obtain and process.

(2) the variable of managing service, for the same device, while the scenario is changed, the device's duty will be changed, too. Meanwhile, in the same scenario, the manage service will be changed along with the time.

Thus, in order to build system application on WiFi Internet of Things, we will solve two problems: the problem domain and system have a chasm, but using hard coding to project problem domain to system domain have a vital programming complexity. Software Architecture using a group of manageable units to express the whole framework of the system, this can play the part of the relationship of demand and the achievement in the system[6].Researchers also using this to solve the project from demand to achievement, and answer the problem of the system's complexity.

In order to rapid customize and develop an Internet of Things' System according to the require of management, we propose a new method to manage Internet of Things based on WiFi and runtime model[7]. Specifically provides as follow, we give an example in an actual scenario of company management using our method, and during this example we can verify the feasible and validity of our method.

First, construct the runtime model for every single sensor device, perform the monitor for single sensor device using device management interface.

Second, above the performed runtime model, we customize, extract and merge the data from different sensor devices, this composite patterns model will make unified management to different sensor devices possible.

Last, build the relationship between objective objects and sensor devices in application scenario, project composite patterns model to application scenario model by model conversion. Then, the data gathering from all kinds of sensor device will be demonstrated for objective world's corresponding object properties.

2. Overview Methods

The overview of WiFi sensor network based on runtime model is described in Figure 1[8]. We can find in Figure 1, this method import runtime model software architecture in WiFi sensor network management. Then, project from WSN data to application scenario objective object attributes by model construction, model synchronization and model transformation. The method include 3 parts of work[9]:

(1) Construct WiFi sensor device runtime model by the existing management interface, propose a method to construct sensor device runtime model to shield the isomerism of sensor device manage interface.

(2) Construct a composite patterns model by distributed sensor device runtime model, and propose a merged method to build distributed runtime model, this strategy will shield the distributivity of the sensor device.

(3) By means of the model converting method, we construct a projection from composite patterns model to application scenario model. We propose a new method to convert models from composite patterns to application scenario.

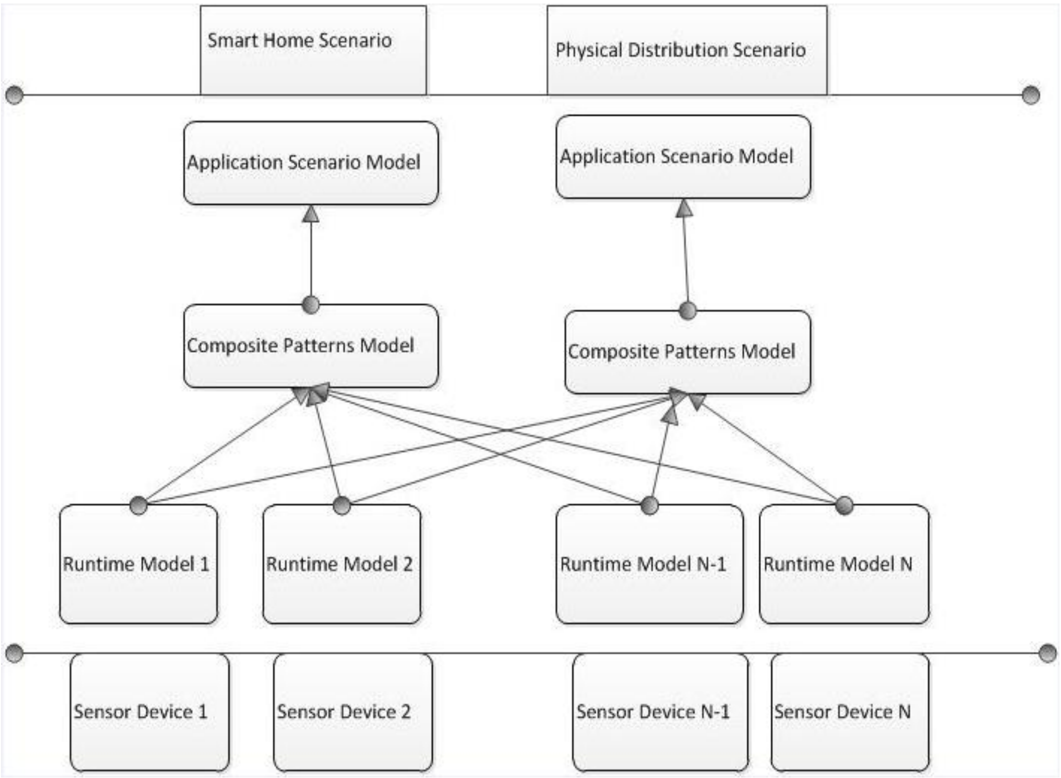


Figure 1. Overview of Management Approach of Wireless Sensor Networks based on Runtime Model

3. Construction Method of Runtime Model on Sensor Device

The core of Internet of Things is the WiFi sensor network, this core net is aimed at solving the information perception problem, but the diversity and the isomerism of sensor device make the data obtain difficult and complex. So, in this article, with the help of SM@RT tools[10], we construct a runtime model based on sensor device, and this model use the same method to gather and process data. SM@RT is one of the most significant tools to build the runtime model, its generator based on the input of meta-model and data access model, it can automatically generate infrastructure which will maintain the software architecture when in service, it also can reflect the state of formation to runtime model which make manage the formation possible.

Here is an example, Figure 2 shows a management procedure. The RFID is a frequency identification device which will be managed by our model. We named the RFID 158, and move it from room 1434 to room 1621, the synchronisatie-engine detect that the room 1621 added a new RFID tag, and this engine automatically add a new RFID tag in RFID Reader 1621, meanwhile, automatically delete the id 158 RFID tag from RFID Reader 1434, this means that the RFID 158 is deviated from room 1434. The example demonstrates how to construct runtime model based on sensor device, we construct it by open source method SM@RT tools.

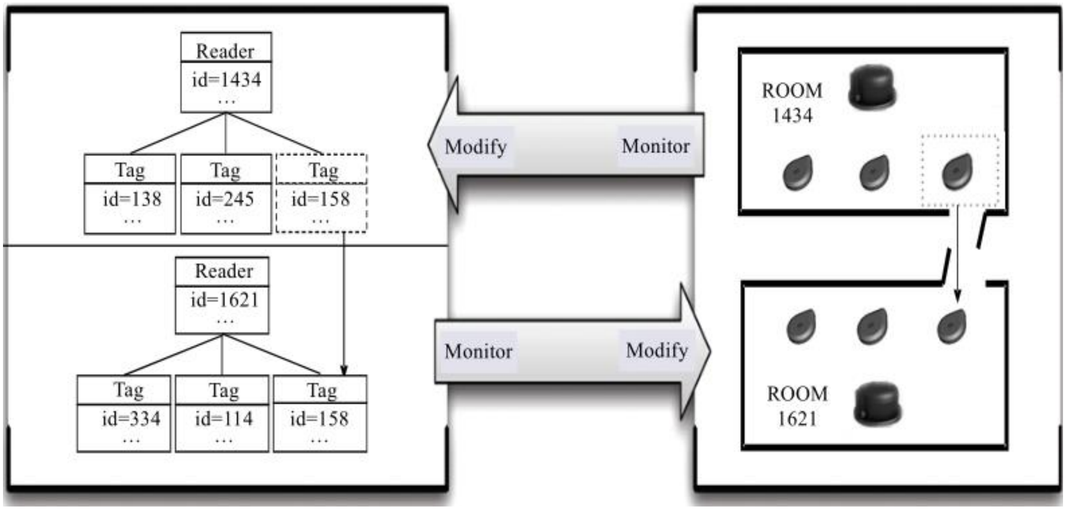


Figure 2. Synchronization between Runtime Models and Wireless Sensor Devices

4. A New Merge Method of Distributed Runtime Model

The WiFi sensor network is made up with various sensor devices, application system will unified process and analyze data from different sensor devices, so we must combine different sensor devices to a identical model to process data. In this chapter, we introduce a new merge method to combine different sensor devices by Distributed Runtime Model.

4.1. Model Customize

To customize a new model, we need to aim at the require of management, take corresponding fragment from the sensor devices runtime models, then merge these model fragments to a new model, the composite patterns model. Figure 3 give a management scenario of RFID 1621 of a meeting room's lighting equipments. The scenario of management needs sensor devices to provide the gathering location information and the luminance information. According to these information, the model will demand the on-off state of lighting equipment by the numbers of people and the luminance situation. Sensor devices offer the acquisition system of the location system and the environment system, respectively. Thus, we customize our runtime model based on the above two devices. In the example, we take the location information from RFID 1621 acquisition system whole model and gather the environment information on runtime model, then combine them as a new specific management require composite patterns model. As shown in Figure 3, the composite patterns model Reader element and the corresponding Tag element are easy to reflect the people's situation in meeting room, sensor element has a brightness attribute to reflect the brightness situation of meeting room. When the meeting room is occupied and the meeting room is lack of brightness, so the lighting equipments need to be opened. The customized model fragment needs to be able to acquire target device gathered information from the specific sensor device. The storage of sensor device runtime model and the model fragments formulated by administrators are formed as XML files, for an element of the model, there is a path set out from root node to the corresponding XML element. As shown in Figure 4, the figure (a) is a model fragment formulated by administrator, and the figure (b) is the corresponding runtime model, these elements have the similar organization structures. Users can customized their own elements, and these elements are started on the runtime model root node, the runtime model can using the customized path to find the corresponding elements by layers, this procedure perform a data correlation about the model fragments and the runtime model. For example, the RFID 1621 with the brightness property formulated by administrator of Tag, users can find the model fragments via the following path ⟨EnSystem⟩→⟨Tags⟩→⟨Tag id=1621⟩→brightness, in the Figure, we also give the XML file to show how the elements and the property organized in XML files. For other elements of the runtime model, users can use the same strategy to find what they need.

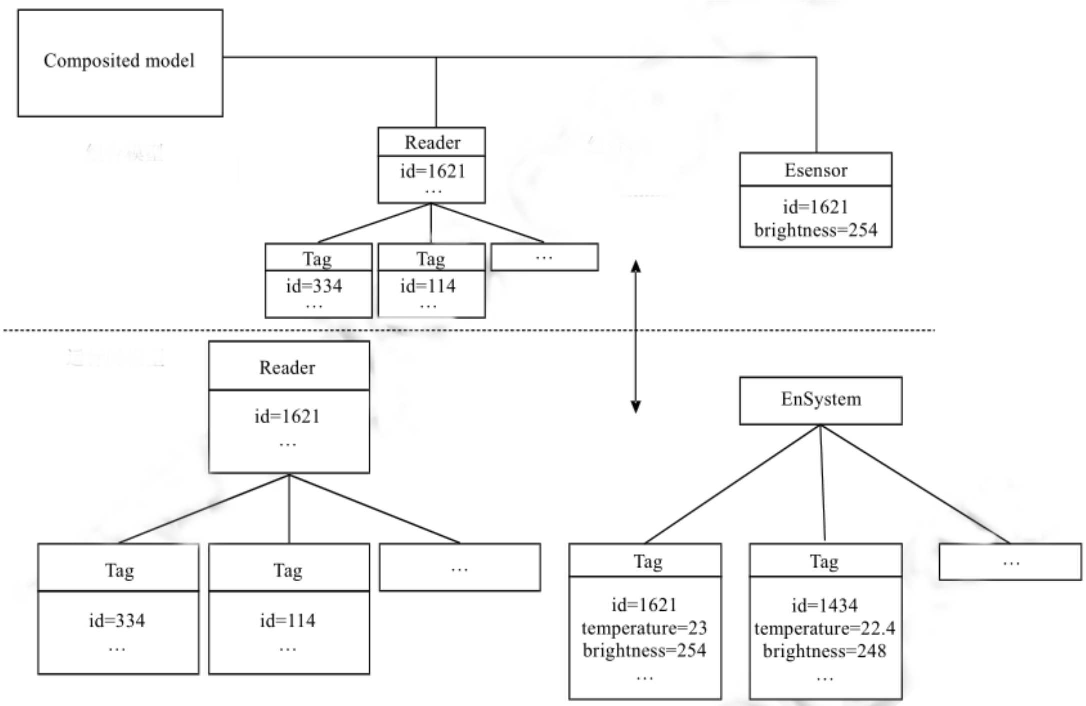


Figure 3. An Example of Composite Patterns Model Construction

```
<?xml version="1.0" encoding="ASCII"?>
<En_Sys:EnSystem xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://sei.pku.edu.cn/EnSystem/En_Sys">
    <Tags>
        <Tag id="1621" brightness="254" humidity="52" temperature="23" />
        <Tag id="1434" brightness="248" humidity="61" temperature="22.4" />
        ...
    </Tags>
</En_Sys:EnSystem>
<EnSystem>
    <Tags>
        <Tag id="1621">
            <property>brightness</property>
        </Tag>
    </Tags>
</EnSystem>
```

Figure 4. Fragment Recognition in Runtime Models of WiFi Sensor Devices

The model fragments in composite patterns model come from different sensor device runtime model, so they didn't have the grammatical conjunction. Thus, in order to solve the minor problem, we regard every model fragment root node as the composite patterns model root node, and these children nodes can be combined directly. Meanwhile, in order to solve the name conflicts for different model fragments, we need to provide a namespace, in the namespace, we record the conflict elements and replace them. As shown in Figure 5, the Reader module and the EnSystem are combined, but these two model fragments have the same Tag elements, so these Tags have the name conflicts. So, we change the EnSystem tag as Esensors and the Esensors Tag remain unchanged, under this strategy, the name conflict have been solved. At last, we can combine these two model fragments.


```xml
<merged:Merged>
    <Reader id="1621" mark="162.20.11.24_RE_Sys">
        <Tags>
            <Tag id="334" />
            <Tag id="114" />
            ...
        </Tags>
    </Reader>
    <EnSystem mark="220.105.31.245_En_Sys">
        <Esensors>
            <Esonsor id="1621" brightness="254" />
        </Esensors>
    </EnSystem>
</merged:Merged>
```

```xml
<EnSystem ip="220.105.31.245">
    <Tags>
        <Tag id="1621" brightness="254" />
    </Tags>
</EnSystem>
```

```xml
<namespace_description>
    <Reader>
        <Tags />
        <Tag />
    </Reader>
    <EnSystem>
        <Tags alias="Esensors" />
        <Tag alias="Esensor" />
    </EnSystem>
</namespace_description>
```

Figure 5. Solution to Naming Conflict in Composite Patterns Models

4.2. Data Synchronism

Based on the model customized, the data from composite patterns model and the system runtime model must be synchronized, after synchronized, the composite patterns model have the ability to manage the different systems. Composite patterns model was constituted by different system runtime model elements, the data consistency was accomplished by the corresponding model fragments. In order to solve the problem of data process, we deploy a duplication of model fragments when the system runtime model was running, and we use the pooling-driven strategy to compare and find the change of the model, model operations will be automatically generated, these operations are sent to composite patterns model to be executed. When the system runtime model changes have been detected, in order to preserve the data consistency about composite patterns model and the system runtime model, the model need to sense the change, and feedback the change to the composite patterns model. The difficulty of this procedure is how to find the model's changes and generate the corresponding operations. In this article, we use depth-first algorithm to achieve this:

(a) According to the description of the customized model fragments, when the system start to run, extract new model fragments.

(b) From the old model fragments' root node, compare the new and old model fragments for every node:

If the node is root mode, then compare this node and root node, if it is differ from the root node, repeat this step. If the node is not the root node, then execute the step (c)

(c) examine the new model fragments whether have this node or not, if it existed, then execute the step (d). If it is not, generate the operation "remove" according to the child node, and execute the step (f).

(d) examine the node in the new model fragments, judge whether the attribute have changed or not, if it is not, then execute the step (e). If it is, generate the operation "set" according to the child node, and execute the step (e).

(e) in the old and new model fragments, compare the node's child node, and judge whether it has changed or not. If it is not, then execute the step (f), if it is, generate the operation "add" according to the child node, and execute the step (e).

(f) find the next node in the old model fragments, if it has the next node, go to the step (b), if it not has the next node, the algorithm has gone to the end.

5. Convert Approach from Composite Patterns Model to Application Scenario Model

In different application scenario, sensor devices gather the data from the environment, these data are often used for describing the objective object, and it is must build the relationship between the objective object's attributes to the sensor devices in the application scenario[11]. After deep research in the relationship, we devise a mapping relationship description rules and the method for automatically generate the corresponding codes, the administrator can use this method to define the mapping relationship description rules to get the model convert code, always code by QVT language. The mapping relationship description rules are described by a XML file. The key-words was defined as:

(1) Helper: this key-word used for describing the mapping relationship between the elements, helper label have three attributes, such as key, value and type.value, these attributes have ability to express the elements in the composite patterns model. "Key" represents the elements in the application scenario, "type" represents the model mapping relationship type, when the type value called "basic", this represents that the helper is "one-to-one" or "more-to-one" mapping relationship, when the type value "multi-" this represents that the helper is "one-to-more" mapping relationship. When the "one-to-more" mapping relationship is generated, the helper appear by group, and the property "condition" describes the happen condition of the mapping relationship. As the elements have other elements or properties, the helper label often nest helper label, mapper label and query label. Among these labels, the mapper label and the query label are expressing the mapping relationship of properties.

(2) Mapper: this key-word used for the mapping relationship of properties. The mapper layer often have two properties------"key" and "value.value" express the properties of the composite patterns model, respectively. The key property expresses the corresponding property in the application scenario model. Especially, the elements of the property is defined by the outer layer helper label.

(3) Query: this key-word used for describing the mapping relationship of properties. Query label have four properties, are "key", "node", "value" and "condition". In these properties, the definition of property key and value is the same as the mapper label, and the elements of the properties in the application scenario model are defined by the outer layer helper label. However, the elements of properties in composite patterns model are defined by two properties called "node" and "condition", they express the type and the condition of the elements, respectively. Especially, the "query" label is used for the description of a mapping relationship "more-to-one" between the elements.

As shown in Figure 6, based on the above key-words, we can describe the mapping relationship for corresponding elements by model mapping relationship:

(1) The "one-to-one" mapping relationship in the model elements. The first example is a description of "one-to-one" mapping relationship in composite patterns model for the Power elements and the Light elements. In the description, we use helper label to express the mapping relationship from Power to Light, use mapper label to express the mapping relationship from the properties "id, switch" in Power to the properties "id, on, off" in Light.

(2) The "more-to-one" mapping relationship in the model elements. The second example is a description of "more-to-one" mapping relationship in composite patterns model for the Reader elements and the Room elements. In composite patterns model, the Reader elements and the Room elements express the same information of room. Thus, we use the helper label and mapper label to describe the two type elements and the mapping relationship of them. However, Room elements have more information than Reader elements, but this information consists in Esensor element. So, we use the query label to describe the mapping relationship from Esensor elements' property to Room elements' property. The property of "key" and "value" in query label expresses the application scenario model and the composite patterns model, respectively. "Node" property value "Esensor" express the element of the property have the type of "Esensor", and the condition property value "id = self.id" express the condition of the element which need to be meet.

(3)The "more-to-one" mapping relationship in the model elements. The third example is a description of "one-to-more" mapping relationship in composite patterns model for the Tag elements and the Person, Pet, or Car elements. We use a group of type value "multi-" helper label to express this mapping relationship. For example, when the Tag element's type value "person", the composite patterns model map the Tag elements to the "Person" elements.

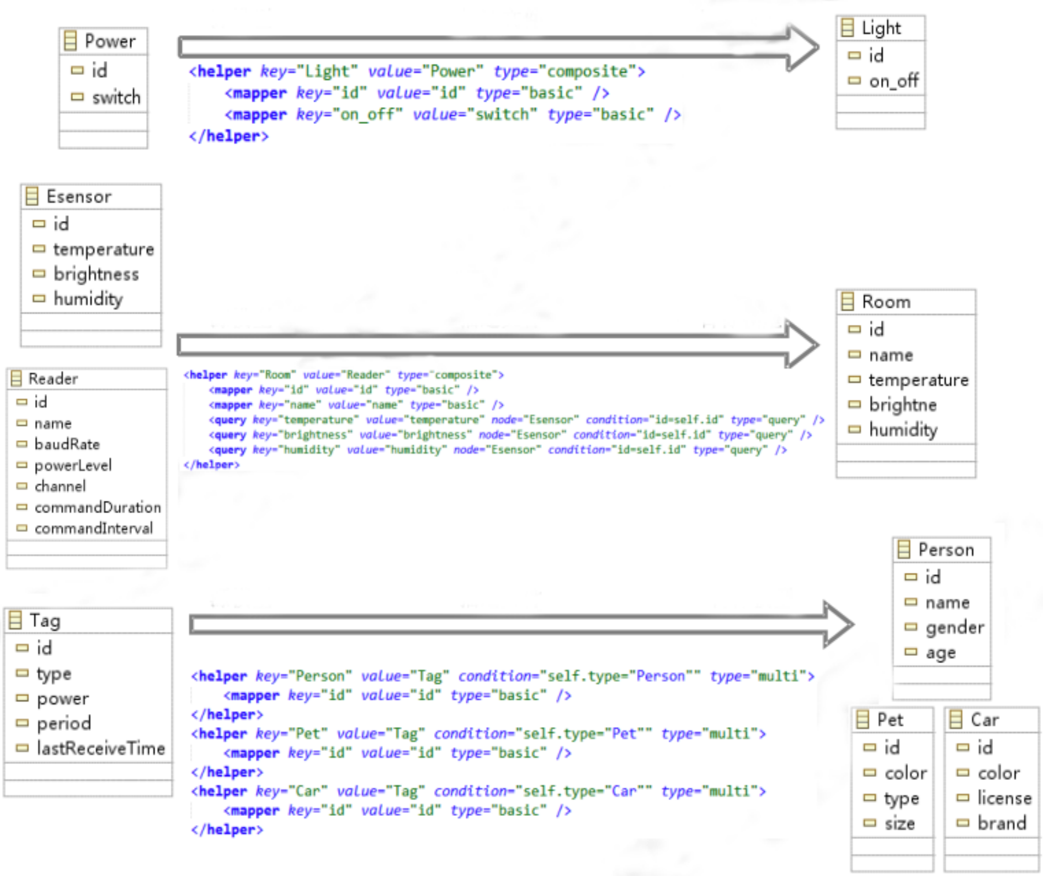


Figure 6. Descriptions of Basic Mapping Rules between Model Elements

6. An Example of Management on Runtime Model

Enterprise management is one of the application fields of Internet of Things, in various WiFi sensor device and complex management procedure, develop an application system for enterprise has a great difficulty[12]. In order to validate our method's feasible and validity, we give an example of management on runtime model of enterprise, aim at managing and monitoring an enterprise's staff, cars and devices. Based on the RFID devices and enterprise management system, we achieve an enterprise management report system, this system can raise the report when the enterprise have some abnormal events. At last, we compare our method to state-of-the-art methods, evaluate our method's feasibility and validity.

```
one-to-one

helper Power :: Power2Light : Light{
    return object Light{
        id:=self.id;
        on_off:=self.switch;
    }
}

one-to-more

helper Tag :: Tag2Person : Person{
    if (self.type = "Person"){
        return object Person{
            id:=self.id;
        }
    }
}
...

more-to-one

helper Reader :: Reader2Room : Room{
    return object Room{
        id:=self.id;
        name:=self.name;
        temperature:=EnSystem.objectOfType(Esensor)->select(id=self.id)->selectOne(true).temperature;
        brightness:=EnSystem.objectOfType(Esensor)->select(id=self.id)->selectOne(true).brightness;
        humidity:=EnSystem.objectOfType(Esensor)->select(id=self.id)->selectOne(true).humidity;
    }
}
```

Figure 7. Example of the Code Generation based on Mapping Relationships

6.1. Runtime Model on Company Management Information System

RFID sensor device can obtain people and goods' location information, this information include RFID reader-writer and RFID label. In the management system, we arrange RFID reader-writer in the critical area in enterprise region, and give the RFID label to every people and goods in enterprise. The RFID reader-writer can gather the label information in real time, using this information, we can judge people and goods' location. The Figure 8 described what elements and properties include in the RFID sensor device, as seen in the figure, Reader property describes the basic configuration information of RFID label. The relationship of Reader property and Tag property reflect the location information of RFID label. When the location of RFID label have changed, the corresponding Tag property and Reader property also have changed. Using the Runtime Model on management of Enterprise, the efficiency of management will be better, the runtime model is real time model, so we can manage the enterprise more efficient by the runtime model. Besides, the runtime model have good ability to manage the corresponding things in the WiFi sensor network.

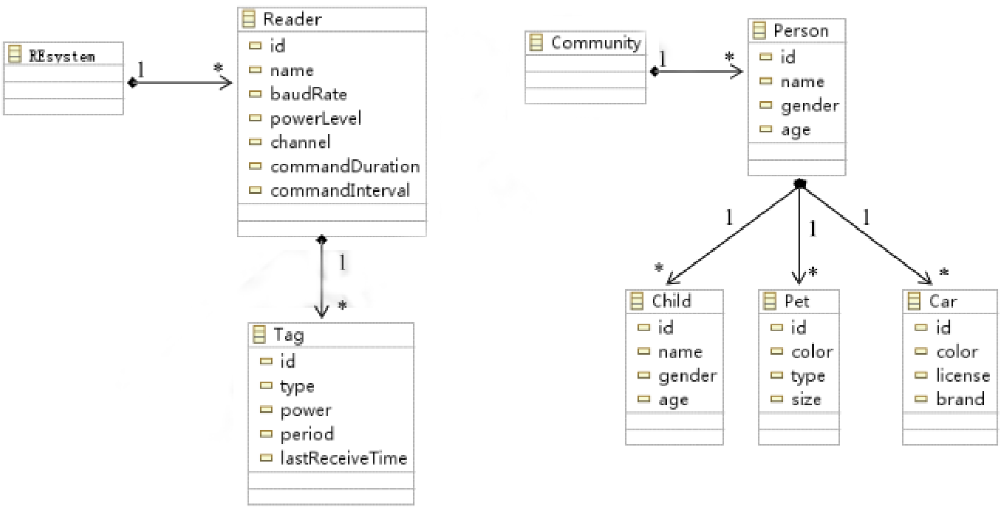


Figure 8. Architecture-Based Models of RFID System and Information System

Construct a RFID sensor device on runtime model, we can get the basic information and relationship of RFID reader-writer in model layer. Follow this principle, when we construct the runtime model of enterprise information management system, we can build the staff and goods' basic information and relationship in model layer.

6.2. Combined Model on Company Management Information System

With the form of model fragments, we extract object location information from RFID sensor device runtime model, and extract staffs and their goods information from enterprise information system, then combine the two models to construct a new model. As shown in Figure 9, the elements and properties in the composite patterns model are come from runtime model, and contain the two aspect of information that described above. Enterprise can make sure where are the staffs and their goods via composite patterns model.

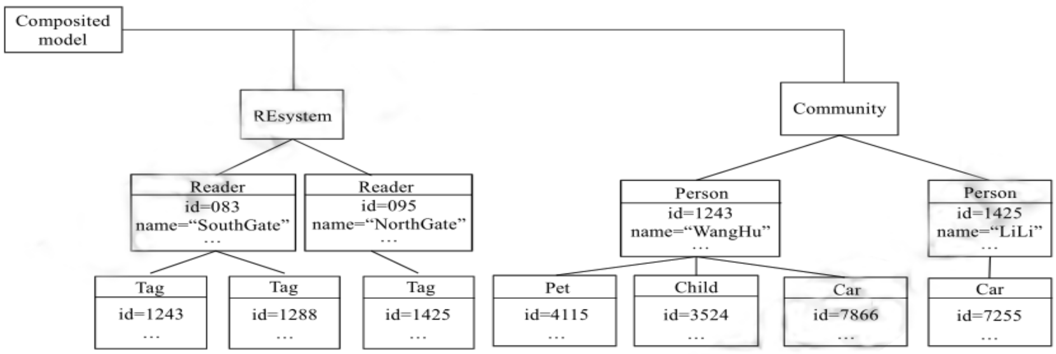
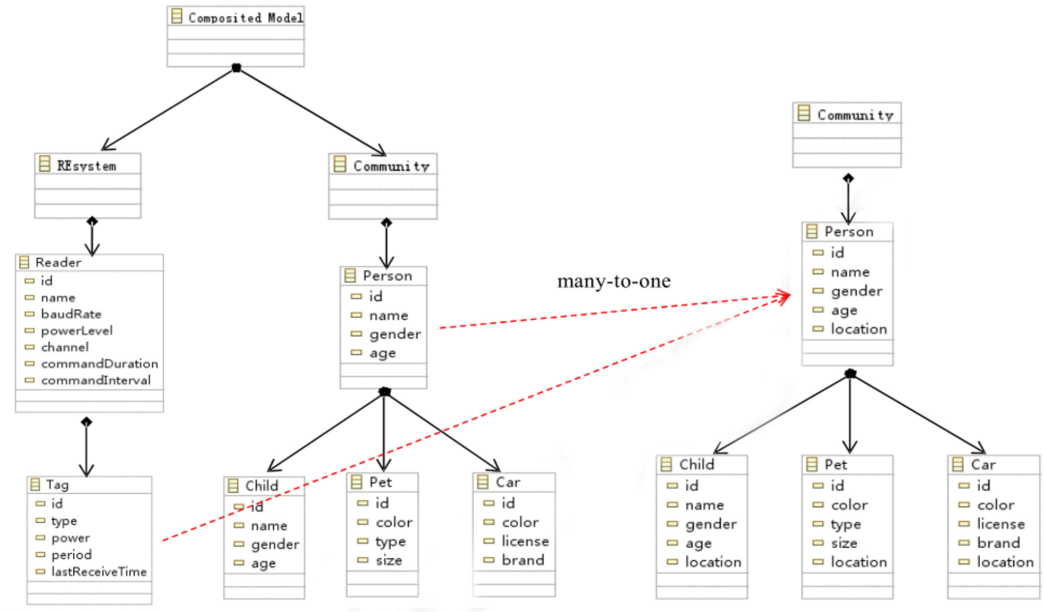


Figure 9. Composite Patterns Model of Alarm System

6.3. Convert Approach from Composite Patterns Model to Application Scenario Model

Although composite patterns model include the basic information and location information of person, device, good and car, but the administrator can't gather data directly from composite patterns model, this function need additional code to accomplish. Enterprise management system need location information and the mapping relationship of Person, good, device and car.

As shown in Figure 10, we construct an application of enterprise management system, the application scenario: Person element describe the identity and location information of staff, and include related device, good and car lists. Device, goods and car elements describe the identity and location of device, goods and car, respectively. These elements include a list of related person. Next, we build the relationship from composite patterns model to scenario model. The composite patterns model include the person, device, goods, car and tag elements, and convert to the person, device, goods, car in the application scenario model, this conversion is the key of model conversion. We give an example of person and tag elements in the composite patterns model that convert to person element in the application scenario model, introduce the mapping relationship in detail. Whether in the composite patterns model or the application scenario model, the Person elements express staffs. But the location information of staffs in the composite patterns model are expressed by the relationship of Reader element and Tag element, and the location information of staffs in the application scenario model are expressed by the location property of the Person element. So, this mapping procedure is a "more-to-one" mapping from Person elements and Tag elements in the composite patterns model to the Person elements in the application scenario model.



Mapping description

```
......
<helper key="Person" value="Person" type="composite">
      <mapper key="id" value="id" type="basic" />
      <mapper key="name" value="name" type="basic" />
      <mapper key="gender" value="gender" type="basic" />
      <mapper key="age" value="age" type="basic" />
      <query key="location" value="Reader.name" node="Tag" condition="id=self.id" type="query" />
</helper>
```

Figure 10. Descriptions of Mapping Rules of Model Transformation

Figure 10 shows the description of the mapping relationship. We can see that the Person element's location property in the application scenario model is determined by the Tag element and related Reader element's name property in the composite patterns model. According to the conversion from composite patterns model to application scenario model, any change of the composite patterns model can be reflected in the application scenario model automatically. Then, the location information and the corresponding relationship of Person, good, device and car can be shown in the system by objective properties of the objects. During the conversion from composite patterns model to application scenario model, the relationship is more easily to recognition, and the system can automatically generate the code to help administrator to manage the system. This will reduce the coding time for system, and the automatically way will be more accurate to manage the enterprise.

7. Tag

The data gathered by sensor device are real-time, huge amount, but without a good structure. If we map these non-structure data to Internet of Things' problem space domain, we have to write a lot of code to implement the mapping procedure, this work face to two challenges, the various devices and the complex management service. In order to develop the Internet of Things system based on the require of management, we import runtime model architecture in the procedure of managing WiFi sensor network, and propose a method to manage WiFi sensor network based on runtime model. This method constructs sensor network runtime model, and implement management of single sensor device in model layer, it can also manage different sensor devices by model in application scenario. By means of sensor devices conversion, it make object-scenario application of Internet of Things possible. The future work include two emphasis: one is used our method in actual production environment, performed management for different scenario by various Internet of Things devices. The other is research management strategy on the basis of our method, researches some senior management function based on model analysis, model inference. Finally, perform a perfect system with high error-tolerant rate, high reliability to simplify the development process of Internet of Things system.

References

[1] C. Xiang, P. Yang, X. Wu, "QoS-Based Service Selection with Lightweight Description for Large-Scale Service-Oriented Internet of Things", vol. 20, no. 4, **(2015)**, pp. 336-347.

[2] R. Gunasagaran, L. M. Kamarudin and A. Zakaria, "Internet of things: Sensor to sensor communication", SENSORS, IEEE. IEEE, **(2015)**, pp. 1-4.

[3] L. Jun and G. Wei-hua, "Node deployment strategy optimization for wireless sensor network with mobile base station", DOI:10.1007/sl1771-012-1024-3, vol. 19, no. 2, **(2012)**, pp. 453-458.

[4] Z. Pang, Q. Chen and W. Han, "Value-centric design of the internet-of-things solution for food supply chain: value creation, sensor portfolio and information fusion", Journal of Information Systems Frontiers, vol. 17, no. 2, **(2015)**, pp. 289-319.

[5] C. W. Tsai, S. W. Wang and K. C. Hu, "A modified multiple-search multi-start framework-enhanced micro genetic algorithm for WLAN load balancing", Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on. IEEE, **(2015)**, pp. 202-207.

[6] L. Atzori, A. Iera and G. Morabito, "The Internet of things: A survey", Computer Networks, [doi: 10.1016/j.comnet.2010.05.010], **(2010)**, vol. 54, no. 15, pp. 2787−2805.

[7] R. Neisse, G. Steri and I. N. Fovino, "SecKit: A model-based security toolkit for the internet of things", Computers & Security, vol. 54, **(2015)**, pp. 60-76.

[8] Zhao Xue-jian, Zhuang Yi and Wang Jin, "Local adaptive transmit power assignment strategy for wireless sensor networks", DOI:10.1007/s11771-012-1225-9, vol. 19, no. 7, **(2012)**, pp. 1909-1920.

[9] D. Carlson, M. Mogerle and M. Pagel, "Ambient flow: A visual approach for remixing the internet of things", Internet of Things (IOT), 2015 5th International Conference on the. IEEE, **(2015)**, pp. 114-121.

[10] Peking University. SM@RT: Supporting models at run-time, http://code.google.com/p/smatrt/, **(2009)**.

[11] Xue, Guangtao and Qi, "Sociality-Aware Access Point Selection in Enterprise Wireless LANs", IEEE Transactions on Parallel and Distributed Systems: A Publication of the IEEE Computer Society, vol. 24, no. 10, **(2013)**, pp. 2069-2078.

[12] S. Li, L. D. Xu and X. Wang, "Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things", IEEE transactions on industrial informatics, vol. 9, no. 4, **(2013)**, pp. 2177-2186.

[13] P. Zhang and J. Wang, "Management of Intelligent Campus Wireless Sensor Networks Based on Runtime Model", Journal of Computer and Communications, vol. 3, no. 07, **(2015)**, pp. 22.

Authors

Nan Ma, female, doctoral candidate at School of Economics and Management, Fuzhou University, Fujian, China. Research direction: Technological Innovation and Management.

Yingxing Lin, male, professor and doctoral supervisor at School of Economics and Management, Fuzhou University. Fujian, China. Research explores: Innovation and Entrepreneurial Management, and Strategic Management.

Xiang Zhou, male, engineer at Software Institute of Minjiang University, Fujian, China. Research explores: Computer Application.

