

# Middleware Technology Research and Interface Design Based on Internet of Things

Yingmei Wang

Jinan Engineering Vocational Technical College, Jinan, 250014, China  
[wym\\_75@163.com](mailto:wym_75@163.com)

## Abstract

*With the advent of Internet of Things (IoT) technology, communications between physical objects and humans become realistic, and smart devices can be monitored and controlled remotely. It promotes the development of smart home, smart city and even smart planet, which greatly improves the quality and efficiency of human life. As a kind of common services, middleware technology plays an importance role in IoT system. This paper firstly introduces the research background of IoT and middleware technology. Then three main middleware access scenes are analyzed in IoT system, which is determined by the sensing device functions. Lastly, this paper defines the middleware interface, and proposes the interface design method. The middleware interface is classified into three categories: device side interface, application side interface and middleware management interface, in which device side interface defines the data transmission method between the device layer and the platform layer; application side interface defines the middleware opening resources and services between the application layer and the platform layer; and middleware management interface is mainly for devices and middleware management functions. It indicates the message interaction process, and separates different message information, thus improving the data transmission efficiency and enhancing the services open ability of IoT system.*

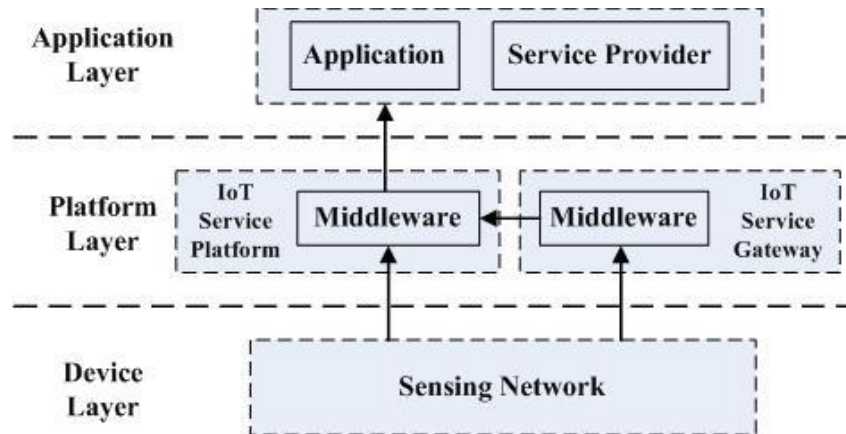
**Keywords:** *Internet of Things, middleware technology, interface design.*

## 1. Background of IoT

Internet of Things, short as IoT, refers to a sensing network formed by sensing devices with control and short-range communication abilities to realize interconnections between the physical objects and physical objects, physical objects and humans, humans and humans. The core feature of IoT is that the physical world state is reflected in data forms, thus statistics, monitoring and other complex functions can be executed based on it. IoT extends the effect diameter and ability of physical world information acquisition, and introduces the new information dimension of physical objects information rather than the traditional human information. At the same time, IoT changes the reverse control ability of physical device state by the controller, and provides the interaction mechanism between human information world and physical world. Thus the abstract digital ecosystem with both physical world and human world is established, and becomes the central nervous of the future information system [1].

IoT connects all the physical objects into Internet through information sensing devices with designated protocols to realize information transmission and interaction. It has features of comprehensive perception, reliable transmission and intelligent processing, which is an extension of the Internet. IoT integrates the sensing devices attached to the physical object into the information space to make them be accessed and controlled wherever and whenever link the Internet sites, and the information world and physical world much closer.

From the basic functions of IoT system, it can be divided into three layers: device layer, platform layer and application layer [2], as shown in Figure 1.



**Figure 1. The Architecture of Internet of Things**

- Device layer, also called as the sensing layer, mainly refers to the sensor network composed of sensors, controllers, transmitters and other underlying communication modules. On one hand, it senses the physical information and uploads it to the platform layer; on the other hand, it also accepts the control instructions from the platform layer, and executes the corresponding actions.
- Platform layer, the core of IoT system, mainly includes the IoT service gateway and IoT service platform. It accesses the underlying data by shielding the sensing devices heterogeneity and opens it to the application layer through the standard interface. Most functions in platform layer are realized by middleware modules.
- Application layer, directly providing services for the users, mainly includes IoT applications and service providers. The abundant applications and services are developed with the underlying resources and abilities.

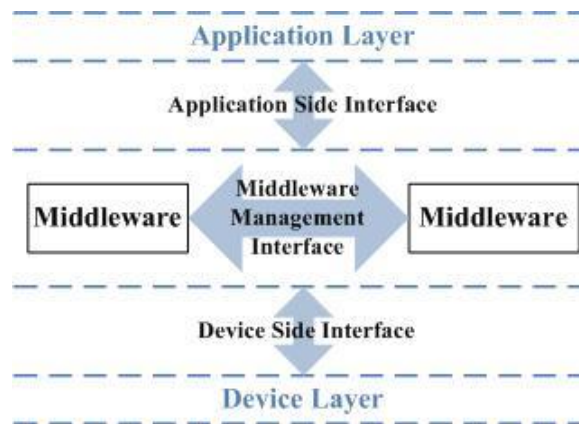
## 2. Introduction of Middleware Technology

Middleware is a kind of common services. It realizes most functions in the platform layer, and also realizes communications between platform layer, device layer and application layer. Middleware provides unified data and environment for services in the application layer to help users develop applications more flexibly and efficiently [3].

Middleware can be divided into different categories according to their purposes, such as data access middleware, remote procedure call middleware, message-oriented middleware and transaction processing monitors. Data access middleware refers to all the software connected with application programs and data sources. As with the general middleware, it allows application services access to the data resources deployed in other nodes through well-defined API (Application Programming Interface). Data access middleware is the best solution to the heterogeneous platform and heterogeneous data sources due to the following features [4]: (1) It can run on multiple hardware and OS platform; (2) It supports standard protocols and interfaces; (3) It has high efficiency and is easy to use.

Middleware has the standard program interfaces and protocols, and can be realized in multiple ways for the different hardware nodes. To improve the data transmission efficiency, this paper defines three kinds of middleware interfaces according to their locations and functions: device side interface, application side interface and middleware management interface, as shown in Figure 2. Device side interface defines the data transmission method between the device layer and the platform layer. Application side

interface defines the middleware opening resources and services between the application layer and the platform layer. Middleware management interface is mainly for devices and middleware management functions. Section 4 will presents the design methods of different interfaces.



**Figure 2. The Structure of Middleware**

### 2.1. Resource Description

Resource description in IoT system refers to the unified abstract of the information and ability resources in IoT system, such as data and service, for system management and resources opening. The heterogeneity of IoT is not suitable for collaborative work between resources and effective organization and management of resources. Defined the unified resource description framework can facilitate the information sharing and collaborative work between services for efficient and intelligent resources management [5].

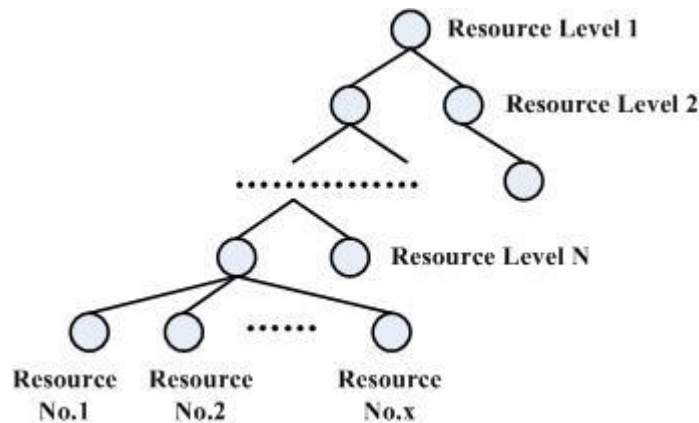
The present resource description methods can be divided into two categories:

- One is the resource description framework based on XML defined by standardization organizations [6]. The representative is WSDL (Web Service Definition Language), which defines different tags to describe the Web service resources, and the resources can register to the service center and be called by the upper applications. However, WSDL can only describe Web services, and it cannot describe other resources in IoT system, such as terminals and data. So it needs to be further expanded with resource models.
- The other method introduces the concept of ontology into the resource description framework. The representative is RDF (Resource Description Framework) proposed by W3C, which describes more network resources than the former method [7]. It makes the relationship between tags more logically, and covers IoT terminals, data and other resources with ontology to open more resources. However, this method is more complex, and more difficult to maintain.

To realize the effective resource description in IoT system, we should firstly list the factors determining the resources classification, such as organization forms, ability, data types, space and time constraints, and the resources can be classified into different categories based on these factors. Then the resource description attributes for each category should be induced, such as category attribute, physical space attribute, resource control attribute, etc. Lastly, resource description metadata set can be defined to realize the standardized description of resources in IoT system.

## 2.2. Identity Distribution

As the basic component of resource identification system, identifier distribution is one of the most important key values for resource management in IoT service environment. The distribution of resource identifier should follow the principles such as uniqueness, extensibility, independence and information reflection. Based on URI format and identification rules, such as IPv6 and ENUM, resource identifier distribution rule with tree-structure is instituted [8].



**Figure 3. Resource Identifier Schema**

As shown in Figure 3, resource identifier format in IoT system is: `http://<resource level 1>/<resource level 2>/.../<resource level n>/<resource number>...`, in which “resource levels” is determined by the IoT service system, service resources types and attributes, resource organization mechanism, resource access mechanism and resource management mechanism, and “resource number” is to identify different resources in the same resource level.

## 2.3. Data Adapter

The differences of sensing data in organization and transmission format are called as data heterogeneity [9], which is the main reason that restricts the data unified transmission between different modules.

In order to realize the unified data transmission in sensing network, the data heterogeneity should be shielded to reduce the difficulty in data processing. Data adapter helps to adapt data into unified format for transmission, and receivers only parses the specified data format. It eliminates the influences caused by data source heterogeneity, reduces the complexity of data storage, analysis and development, and enhances the scalability of sensor network [10].

The heterogeneous data adapter technologies mainly include data migration and conversion, data access middleware:

- Data migration and conversion refers to convert the original data into new data management system. Many relational database vendors provide data adapter functions, such as ODBC, JDBC and OLE DB [11]. This method is simple, and has been understood and accepted by the majority of users, however, it is complex in developing and cannot adapt to the data diversity problem in sensor network.
- Data access middleware is another solution for data adapter, which refers to all the software connected with application programs and data sources. It allows application services access to the data resources deployed in other nodes through well-defined API. Data access middleware is the best solution to the heterogeneous platform and heterogeneous data sources.

### 3. Analysis of middleware access scenes

IoT system can be realized in multiple ways, which mainly includes the following three access scenes according to the devices in the sensing layer [12].

#### 3.1. Access scene 1

As for devices without network communication capacity and secondary development ability in the sensing layer, they need to be firstly accessed to data adapter middleware in IoT service gateway, and then IoT service platform.

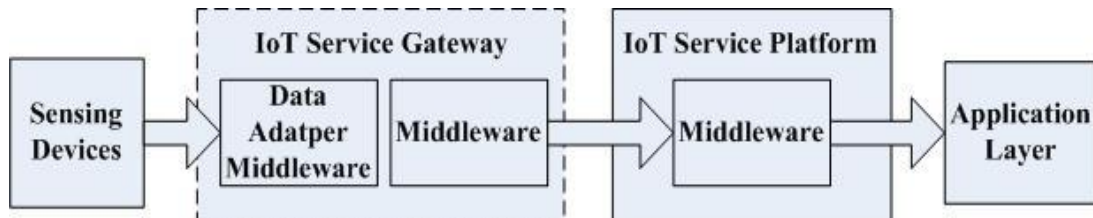


Figure 4. Resource Identifier Schema

As the devices have no network communication capacity, they need to be firstly accessed to IoT service gateway for data forwarding. Also, the devices cannot be secondary developed, and data formats are very different. So all the devices should firstly access to data adapter middleware to unify them into the same format, and IoT service gateway will transfer them to IoT service platform through the middleware management interfaces in the middleware. Then these data and resources of the device layer can be accessed by the application layer.

#### 3.2. Access Scene 2

As for devices with network communication capacity but no secondary development ability in the sensing layer, they need to be accessed to data adapter middleware in IoT service platform.

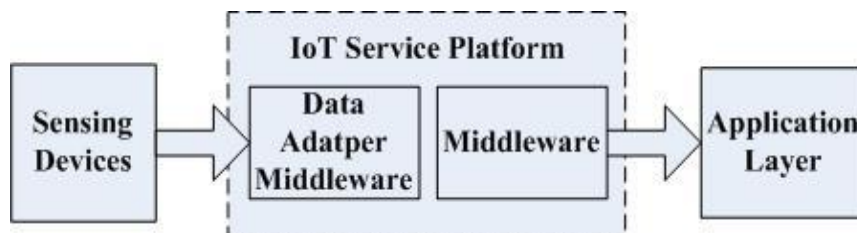
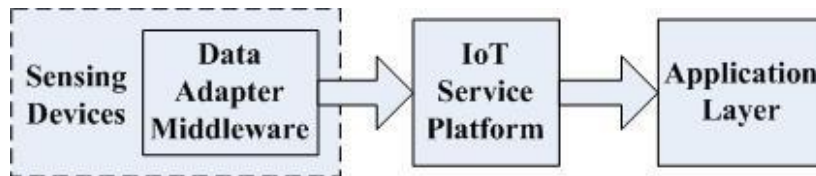


Figure 5. Resource Identifier Schema

Similar to access scene 1, the devices have no secondary development ability, and the data formats are very different. So they should be firstly accessed to the data adapter middleware before IoT service platform. Data adapter unifies these data into the same format for device management and resources open to the application layer.

#### 3.3. Access Scene 3

As for devices with both network communication capacity and secondary development ability in the sensing layer, they can be directly accessed to IoT service platform.



**Figure 6. Resource Identifier Schema**

Devices in the sensing layer can be secondary developed, which means data can be unified to be the same format in these devices. So they can be directly accessed into IoT service platform to be accessed by the application layer.

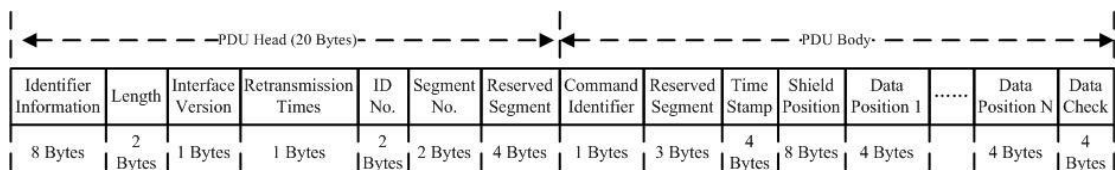
## 4. Middleware Interface Design

As shown in Section 2, middleware interfaces in different locations in IoT system realize different functions. This paper classifies them into three categories: device side interface, application side interface and middleware management interface.

### 4.1. Device Side Interface Design

Device side interface, also called as "D interface", refers to the interfaces between the device layer and platform layer. It mainly defines the methods and content of data transmission between these two layers. As for the different access scenes, device side interface provides two gateway access method and platform access method [13]:

- Gateway access method is for the devices without network communication capacity or secondary development ability, namely the access scene 1 and access scene 2 in Section 3. With this method, IoT service gateway or IoT service platform requires data from the sensors or local servers. Then the data adapter middleware encapsulates data into PDU (Protocol Data Unit), and sent to IoT middleware.
- Platform access method is for the devices with secondary development ability and local sensing data servers, namely the access scene 3 in Section 3. With this method, sensing data is encapsulated into PDU by data adapter middleware, and then sent to IoT service platform.



**Figure 7. PDU Message Format**

Figure 7 shows the format of PDU message, which is composed of PDU header and PDU body. The length of PDU header is 20 bytes, including identifier information, length, interface version, retransmission times, ID No., segment No. and reserved segment. The length of PDU body is uncertain, which changes with the transmitting resources, including command identifier, reserved segment, timestamp, shield position, data positions and CRC data check code [14].

"D interface" supports TCP, UDP, serial port and other transmission ways. Data should be firstly encapsulated into PDU format before accessing to IoT middleware, and then transmitted in appropriate way.

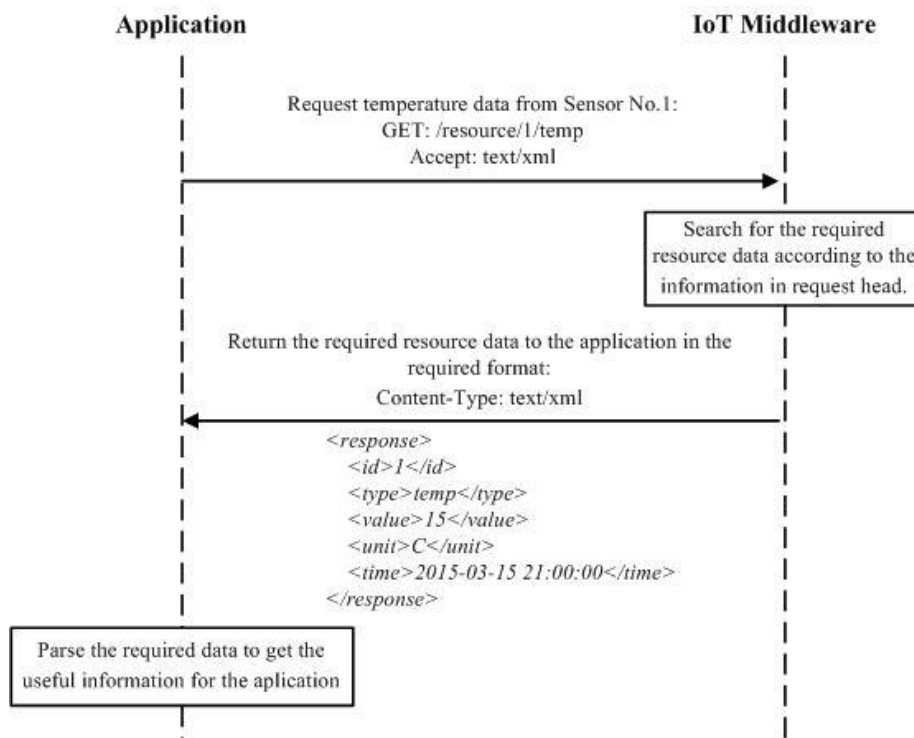
Device accessed by "D interface" is simple in both message processing and data flow. Also, it can identify the data error in transmission to ensure its reliability. "D interface" transmits data without considering its content, thus shielding the heterogeneity caused by

different data sources. It is convenient for upper layers processing unified data, and provides possibilities for abundant IoT applications with powerful functions.

#### 4.2. Application Side Interface Design

Application side interface, also called as "A interface", refers to the interfaces between the application layer and platform layer. It mainly defines ways for middleware opening resources and services. IoT middleware provides two kinds of resources through "A interface": one is the service data such as the sensing information; the other is the service ability such as the descending control information.

As "A interface" is designed based on HTTP protocol, each resource has a unique URI. The operations of "a interface" are basic HTTP methods, such as GET/POST/PUT/DELETE, thus the applications access to the heterogeneous resources without understanding the underlying structure [15]. Each HTTP request is composed of a request head and a request body, in which request head indicates the requester address, request time, request resource, etc; and request body indicates the specific resource description content in text, HTML, XML, JSON and other formats [16].



**Figure 8. Process of Requiring Temperature Information from a Sensor by "A Interface"**

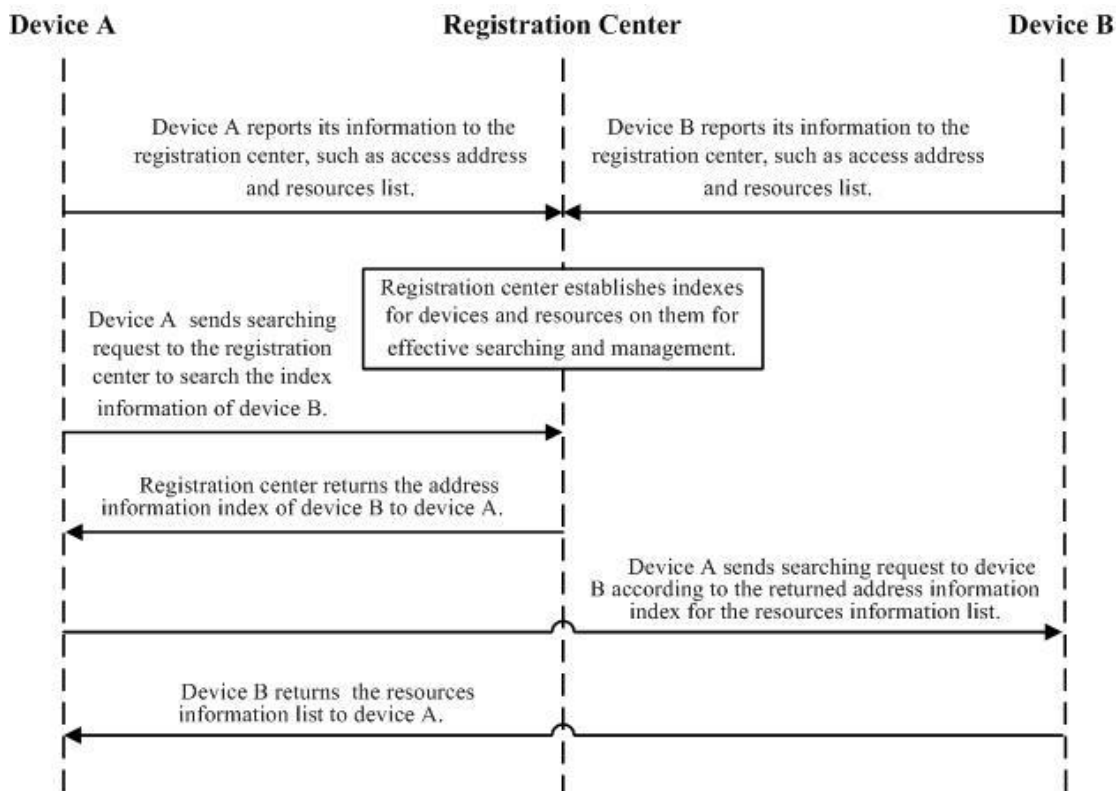
Figure 8 shows the process of requiring the temperature information from a sensor by "A interface". Application firstly sends a request to IoT middleware, and the middleware searches for the required resource after receiving the request. Then the middleware returns the required resource to the application. After receiving the required resource, the application parses it to get the useful information, and the information requiring process ends.

#### 4.3. Middleware Management Interface Design

Middleware management interface, also called as "M interface", is mainly for devices and middleware management functions in IoT system. It can be further divided into data

interface and management interface, in which data interface transfers the data information, and management interface transfers the control commands. Middleware management interfaces support the following functions [17]:

- Devices registration, cancellation, certification and identification distribution functions for IoT service gateway and IoT service platform.
- Device discovery, IoT service gateway discovery and IoT service platform discovery in IoT system.
- Data searching function in IoT service gateway and IoT service platform.
- Monitoring function in IoT service gateway and IoT service platform (for example, checks the status and data information of a sensor or a set of sensors).
- Data controlling function in IoT service gateway and IoT service platform (for example, turn off a sensor or a set of sensors).
- Data synchronization between IoT service gateways or IoT service platforms.



**Figure 9. Device Discovery Process by “M interface”**

Figure 9 shows the process of device A discovering device B in IoT system. All the devices should firstly report their information to the registration center, and the registration center establishes device indexes for searching and management. If device A wants to discover device B, it sends the searching request to the registration center, and the registration center returns the index information of device B to it. Then according to the returned index information, device A sends searching request to B for the resources information list. After receiving the request, device B sends the required information list to device A, and the discovery process ends.

## 5. Conclusions

The unprecedented rise of IoT technology provides opportunities for human to interact with objects in the physical worlds, making the research on smart systems and applications based on them are recently a hotspot [18]. As a kind of common software,



middleware provides unified data for IoT services to help users develop applications more flexibly and efficiently. Based on the introduction of IoT and middleware technology, this paper proposes the middleware interface design method. The main contribution of this paper is that it classifies the middleware interfaces into three categories according to their functions: device side interface, application side interface and middleware management interface. It defines the interface message and interaction process, thus separating the transmission methods of different information, improving the data transmission efficiency and enhancing the services open ability of IoT system [19].

Due to space limitations, the interface design in this paper is only introduced for the general scene. IoT system can provides various powerful applications based on the underlying resources and abilities, so the middleware technology and interface design method is worth continue discussion and further improvement for some specific applications in future studies.

## Acknowledgement

These should be brief and placed at the end of the text before the references.

## References

- [1] D. Miorandi, S. Sicari and F. D. Pellegrini, "Internet of things: Vision, Applications and Research Challenges", *Ad Hoc Networks*, vol. 10, no. 7, (2012), pp. 1497–1516.
- [2] L. Atzori and A. G. Morabito. "SIoT: Giving a Social Structure to the Internet of Things", *Ommnaon Lr*, vol. 15, no. 11, (2011), pp. 1193 - 1195.
- [3] N. Ibrahim, "Orthogonal Classification of Middleware Technologies", *Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, (2009).
- [4] H. Dong, "Middleware Technologies", *Journal of Anhui Vocational College of Electronics & Information Technology*, (2006).
- [5] Scientist E M R, "An Introduction to the Resource Description Framework", *Journal of Library Administration*, vol. 34, no. 3, (1998), pp. 245-255.
- [6] C. Ensel and A. Keller, "Managing application service dependencies with XML and the resource description framework", *Integrated Network Management Proceedings, IEEE/IFIP International Symposium, IEEE*, (2001).
- [7] B. McBride, "The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS", *International Handbooks on Information Systems*, (2004).
- [8] V. E. Moscaritolo and D. Cokenias, "Encrypted Universal Resource Identifier (URI) based Messaging", (2010); US.
- [9] J. P. Huelsenbeck and P. J. Waddell, "Is Character Weighting a Panacea for the Problem of Data Heterogeneity in Phylogenetic Analysis?", *Systematic Biology*, vol. 43, no. 2, (1994), pp. 288-291.
- [10] J. Autio, V. Vatulala and P. Ollikainen, "Communication System and Data Adapter", (2000).
- [11] S. Q. Wen, "Modern Database Programming Interfaces and Connection", *Computer Programming Skills & Maintenance*, (2010).
- [12] V. Mann and M. Parashar, "Middleware Support for Global Access to Integrated Computational Collaboratories", *hpdc*, (2001), pp. 35 - 46.
- [13] Y. Yan, Z. F. Xu and X. Zhu, "A Middleware of IoT-Based Smart Home Based on Service", *Sustainable Development of Urban and Rural Areas*, (2014).
- [14] L. R. Freeman III, "Protocol data unit prioritization in a data processing network", (2001); US.
- [15] P. Pruthi and A. K. Jena, "A Popescu, HTTP Interactions with TCP", *The 11th ITC Specialist Seminar*, (1998).
- [16] H. Chen, X. Jia and H. Li, "A Brief Introduction to IoT Gateway", *Communication Technology and Application (ICCTA), IET International Conference, IET*, (2011).
- [17] H. A. D. Limon, G. Coulson and G. S. Blair, "Adaptive Resource Management in Middleware: A Survey", *Distributed Systems Online, IEEE*, vol. 5, no. 7, (2004).
- [18] Q. Z. Hong and W. Y. Jun, "IoT Technology and Application", *Acta Electronica Sinica*, (2012).
- [19] J. S. Kang, Y. G. Kim and H. S. Park, "Remote Data Transmission Middleware for Telerobotics", *IT Convergence and Security (ICITCS), International Conference, IEEE*, (2013).

## Author



**Yingmei Wang**, she received the Master degree in Modern education technology from Shandong Normal University in 2005. She works in Jinan Engineering Vocational Technical College .She is currently researching on Network communication and simulation technology.