

The Design of RFID Middleware Architecture for Application in the Advanced Manufacturing Industry and the Internet of Things

Y. LIU

School of Software & Service Outsourcing, Qingdao Technical College, Qingdao, Shandong, China

G.W. LI

Qingdao HIFA Information Technology Co., LTD, Qingdao, Shandong, China

S. GAO

I.O.T. Applied Technology Academy, Qingdao Technical College, Qingdao, Shandong, China

ABSTRACT: In the field of the Internet of things, the RFID (Radio Frequency Identification) electronic tags has been increasingly widely used, but also face the shortcomings including diversity of data interface, the complexity of access and so on. In this paper, the authors introduce a design of architecture for general application program interface (API), which shields the diversity and complexity of RFID equipment. Also it realizes the integrated link management between applications and RFID read-write devices, which can greatly improve the management efficiency of information collection, analysis, processing and automatic operation. According to actual test, it realizes the design of seamless integration middleware, and can greatly promote the industry application of RFID.

KEYWORD: Internet of Things; Radio Frequency Identification; Middleware; Architecture design

1 INTRODUCTIONS

Nowadays, the IOT RFID and the reader-writers are widely utilized in various environments such as advanced manufacturing, logistics, environmental control, equipment building, and so on[1]. Information collection, data analysis and process controls are significantly better facilitated by this emerging technology. However, some problems in technology integration, compatibility and application have been exposed. For example, some issues are how to unify the interface of equipment and application systems from different manufacturers, and how to best timely treat and utilize large numbers of complex data collected by RFID. Another challenge is how best to merge RFID data capture systems with a company's legacy information platform. These problems have seriously hindered the promotion of Internet of things RFID in industrial automatic application and management. Therefore, an urgent need arose for industrial development to design and develop a kind of RFID middleware with a centrally controlled management function to simplify the development and implementation of enterprise RFID projects. This is paramount to actualizing further application of the RFID technology.

A key and common approach in the RFID industry is to propagate the enterprise-level middleware technology to RFID. As the media between RFID reader-writer and the application systems, the

middleware provides a set of universal APIs in realizing the connection from the application program to the RFID reader-writer. Middleware is a software layer that connects and manages application components running on a distribution host, which hides and abstracts many of the complex details of distributed programming from the application developers[2]. RFID middleware shielded the complexity and diversity of equipment related to RFID, which can provide strong support for various background businesses and promote a greater proliferation of RFID application in industries. In an RFID system, the middleware receives data as a stream of tag read events from multiple readers and performs altering operations according to application requests/queries[3]. It has been widely observed that multiple application systems frequently obtain data from RFID devices in various ways. The efficiency of data processing is restricted by the limitation of the network bandwidth. Therefore, an effective information transporting system is needed to overcome this problem. The reader-writer of news delivery systems generates a trigger event, which is rapidly transmitted to the message system, and then the message system determines how to pass the generated data to the corresponding application system. Under this mode, the reader does not need to be concerned with what kind of data is needed by each application system. Furthermore, network communication between these readers does not need to be maintained, and only needs to send a request to

the message system. First of all, the middleware must have the function to decide the direction of message transmission through the contents of event messages. Otherwise, the message system will pass all data to the application program. Consequently, the application program will undertake the task of information filtering. Secondly, RFID middleware should have readers within automatic filtering function that could feed back information processing rules to the readers and thus reduce the demand for network bandwidth. Thirdly, the message system should have an information storage function in order to guarantee that the users receive data properly. In the end, the middleware design must be full-featured, fully compliant with international standards, and able to support simultaneous communication of multiple applications with multiple types of RFID hardware[4], and also with interoperability, scalability and abstraction provision flexibility[5].

Currently, in some of the RFID application fields such as in logistics, RFID reader-writers are usually installed in locales such as warehouse entrances, vehicle entrances and production lines, etc. The IOT manufacturers often collect data from direct-drive RFID reader-writer devices in application programs. However, the number of RFID reading and writing devices and RFID labels will be greatly increased with the expansion of a company's production capacity. In this case, this application mode will obviously not be able to meet the needs of enterprises to expand the scale of production.

The RFID middleware is the basic component of the IOT technology and is the key for the large-scale application of RFID technology. Therefore, the RFID middleware has been classified as a fundamental technology in many RFID industry support policies of China. However, at present the RFID middleware products that could be used in large scale IOT integrated projects are very few in domestic markets. Therefore, it is particularly urgent for our country to develop RFID middleware products with independent intellectual property rights.

2 GOALS AND METHODS

The solution for RFID middleware architecture is to design a new type of software package, namely a new RFID middleware, which could bring about automated production and management control in the advanced manufacturing fields by using IOT technologies. It could improve the efficiency and safety of data collection and automatic recognition in the industrial application of IOT. The key technology lies in the fact that the system structure of Enterprise Middleware is introduced into RFID application to realize the management of RFID

reader-writer devices, management of radio resource, and data packets filtering. The architecture and coordination mechanism is designed to support multi reader-writer networks and achieve large scale industrialization application of Reader-writer device networks. The aim of this research is to solve a key application, i.e., the RFID Middleware could support not only the internationally popular ALE (Application Level Events) protocol and LLRP (Low Level Reader Protocol) protocol reader-writer, but also support major domestic reader-writers. This will make it possible for the domestic reader-writer to be applied in large-scale IOT networking.

When designing an RFID middleware solution, the following issues need to be considered[6].

(a) Multiple types of hardware and vendors support: the middleware must provide a common interface to access different kinds of hardware by one or more vendors.

(b) Real-time handling of incoming data from the RFID readers: the middleware should handle the huge amount of data captured by the connected readers in realtime without readmisses.

(c) Interfacing with multiple applications: the middleware should be capable of interacting with multiple applications simultaneously, by catering to all the requirements of the applications with minimal latency.

(d) Device neutral interface to the applications: the application developer should only use the generic set of interfaces provided by the middleware in dependently of the type of hardware connected to the system.

(e) Scalability: the middleware design must allow easy integration of new hardware and data processing features. This needs distributed design and an open system to adapt to change.

(f) International standards compliant: the middleware design should follow international standards such as EPCglobal so that other related RFID hardware and software can also comply to the standards to make networking easier.

According to the architecture defined by EPC global, a middleware management system for IOT applications should be constructed to effectuate the compatible application of current HF (High Frequency), UHF (Ultra High Frequency), microwave frequency and the RFID middleware in coding system. This middleware management system is a complete control hub for all electronic labels, reader-writers, middleware, apps and networks, which is created to solve a wide array of technical problems such as technical solutions for reader-writer configuration, data collection, data filtering, information distribution, information integration and information security in heterogeneous networks and WAN environment. It will make it possible to seamlessly connect the RFID applications to the SCM (Supply Chain

Management), ERP (Enterprise Resource Planning), MES (Mechanical Engineering System), WMS (Warehouse Management System) and other application systems of the user.

In order to improve the accuracy and efficiency of RFID data collection, the architecture needs to design an integrated system composed of two modules, namely the core middleware (ALE) and data processing application platform DPS. ALE is the core module, while DPS is the interface layer of the application middleware to forward and process data between upper applications and ALE.

3 DESIGN OF RFID MIDDLEWARE ARCHITECTURE AND FUNCTION

3.1 Design of RFID Middleware Architecture

RFID middleware is a functional integration of the program modules, interacting with external information through two interfaces, i.e., the reader-writer interface and the application program interface. The reader-writer interface provides an approach of RFID reader-writer connection. The application program interface provides the connection with other external applications, such as enterprise information management system, ERP, and so on. It can also realize the connection with a specific RFID application or other RFID middleware. Besides these two interfaces, the data interactions among program modules are enabled by their API functions.

The schematic diagram of the RFID middleware architecture design is shown in Figure 1. On the left side of this schematic diagram, originally dispersed reader-writers are connected to a “Reader-writers Pool”, which can be connected with Reader-writer Servers via public interface. The Reader-writer Server is not only responsible for the configurations and settings of “Reader-writers Pool”, but also responsible for passing on the collected information to the event processing system (ALE) to be formatted. ALE will be responsible for packaging the information from different Readers into standard data format, and then transferring it to the Gateway for processing. The Gateway will realize the transformation from the management service protocol to the data service protocol in order to achieve the network information collection from the reader-writer equipment and to realize the integration of a variety of applications and standard management of enterprises.

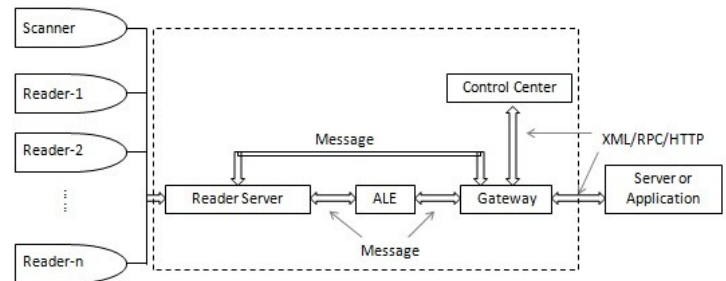


Figure 1. Schematic diagram of the RFID middleware architecture module.

3.2 Design of Function

The designs of four key modules function in Figure 1 are as follows:

3.2.1 Design of Control Center System (CCS) function

The Control Center System (CCS) is responsible for the configuration and management of Reader-writers Servers, ALE Servers and the reading devices. Protocols, which establish communication rules between devices are utilized to monitor and manage communication between tags and readers[7]. The system adopts B/S structure with web server management interface integrated. Users will be able to control and manage the middleware by logging into the CCS using web-browser. The key points of the design and functions to be achieved are as follows:

(1) Achieve the function of supporting the extension of devices by Control Center

The program uses the portal framework in conjunction with databases to save the data of extended devices into the database, which will be helpful for the portal to find the information of corresponding devices. The program will run a newly added device procedure once the user has selected a device. This method finds a solution for the problem of continuous revision of program codes and will be able to reduce the error rate. In this way, problems will be controlled within a certain range and the errors will not be spread any further.

(2) Achieve the function of self-discovery for reading equipment by Control Center

The self-discovery function of equipment is an important feature for the system automatization, which will facilitate the maintenance of systems. By utilizing the Notify Messages or Discovery Messages of reading equipment, the system will be able to monitor the network messages, establish a list of online devices, and support device category lists. By performing a real-time matching of monitored devices, the system will create an instance for the compatible devices and configure them with default parameters or historical settings of parameters in order to meet the needs of the actual environment.

(3) Achieve the statistical function of reading equipment by ControlCenter

The ControlCenter will process statistics of data packages sent from the reading equipment, make statistics of key data categories, and save the results to the database.

(4) Centralized management of configuration information

The Control center is responsible for the centralized management of configuration information and then saving the configuration information by using some kind of information format such as XML. This will make it easier for the system administrator to edit these configuration files.

3.2.2 Event processing system (ALE) functional design

The design principle of this RFID middleware is the eventdriven architecture[8]. Therefore, the main function of this module is to carry out comprehensive treatment of data collected from the Reader Server, and package the label data into a standardized format to provide services for the upper layer application systems. The main points of the designs and functions to be achieved are as follows:

(1) Grouping function of ALE module

According to the EPC global standard, the middleware needs to support the multiple grouping functions, which is also a requirement of real applications. In practical applications, especially in the multi-tag reading system, it is very important to group the label ID. Filtering algorithms are used by the system to fascilitate the grouping of information transferred from the lower levels.

(2) Solution to ALE module reentrant problem

The ALE module reentrant problem is actually a problem to support multiple service requests, and it is a very important issue. To solve this problem, the developer must design a multi-threaded program architecture and address the problem of data sharing and contention.

3.2.3 Design of Reader Server System (RSS) function

The module implements the centralized management and configuration on the physical reading devices. It is responsible for the setting of working status and the mode of the reading devices. C++ program design tools are recommended. The reconstructions of the standard structure of Reader Servers can be realized by using the polymorphism of C++ program. Abstraction of some common base class allows the Reader Device have a better expansion performance. The upper layer Reader Manager can only use the base class to access the general Reader Device interface. Actually, it can access the Reader Device instance of a specific device, which will facilitate the maintenance of code.

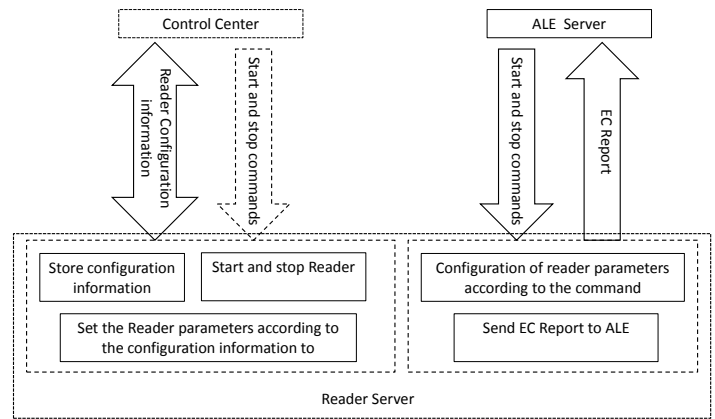


Figure 2. Collaborative work of Reader Server, ALE and Control Center.

Figure 2 is a schematic diagram of the operation information process of the three functional modules of Reader Server, ALE and Control Center working under the general coordination and control of the middleware.

3.2.4 Design of Gateway (GW) function.

This module realizes the transformation between the management services protocol and data service protocol. Using the Singleton single server mode, it can help reduce the waste of system resources, and improve the system performance. At the same time, in order to solve the problem of connection reuse, the connection pool mechanisms established in the Gateway operation, which helps pre-build the network connections to improve the system performance.

4 TECHNICAL ROUTE OF PLATFORM SELECTION, STANDARD AND TECHNICAL CHARACTERISTICS

4.1 Selection of Platform and Standard

C++, Java, C#, JMS, JNI, Soap and similar development tools are recommended. The core module can be built using C#. The benefit of this method is to use the bottom level characteristics of C# to optimize code execution efficiency and improve the efficiency of system performance. The upper application is mainly implemented based on J2EE or DOT NET platforms. As a result, the API interface section built by Java or C# can be connected to the upper level application very well. In addition, developers may consider designing the interfaces based on JMS and WebServices to enrich the diversity of interfaces and provide more flexible choices for users. The JNI technology is used in the calls and conversion between C# and Java to guarantee the stability and security of the call and conversion.

The system should follow an open architecture design route to allow the third party user to carry out the secondary development according to their own

actual situation and needs. They could seamlessly embed some special demand functions of their own. The system should not only meet the EPC Global standards, but also design the RFID middlewares with independent intellectual property rights under the domestic application environments. EPC Global is currently one of the most influential international standard organizations in the areas of RFID applications, and the standards of EPC Global are key instructions the design, development and even application of the middleware. Therefore, the EPC information from different enterprises must be exchanged based on the above standard, and it is apparently the trend of the times to study and follow the EPC Global standards.

4.2 Technical Characteristics of This RFID Middleware

- (1) The ideas of distributed pattern design should be adopted to improve the system scalability and flexibility.
- (2) Reader Server module uses the adapter design pattern to realize the support of various physical reading equipment, and provide a generic interface for the secondary development of systems.
- (3) The system should be compatible with the interface of domestic RFID devices in addition to supporting EPC ALE protocol standards.
- (4) Design the network management functions for reading devices to monitor the realtime status of these devices.
- (5) Enable the smoothing function characteristics of EPC reader protocol in order to improve the reading rate of those reading devices with weak performance ability.
- (6) Capacity to integrate with other technologies; Effective architecture to handle large amounts of data; Security access to data; Interoperability with various devices and high reliability to critical missions[9].
- (7) Dealing with false readings for the delivery of clean data to the applications in accordance with the tag reading environment[10].
- (8) Should provide an adaptive interface for the RFID reader-writer, with the function of data filtering and transmitting, and be able to manage the RFID reader-writer, meet the data request from multiple application platforms and support the existing enterprise application system.

4.3 System Experiment Result

In the process of experiment, the system is consisted with different numbers of RFID equipment samples, which connect with each other by RFID middleware

technology and the traditional technology respectively.

The experiment result shows that as the increase of sample size, it can significantly improve the efficiency of equipment interconnection by using RFID middleware technology. When the sample size is over 100, the RFID middleware technology has obvious advantages.

The experimental result is shown in Figure 3.

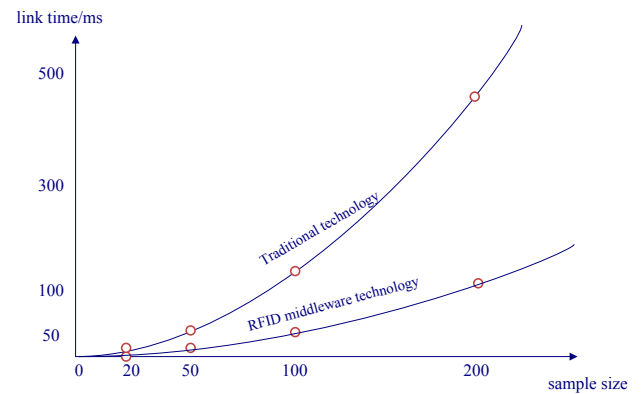


Figure 3. Experimental result.

5 CONCLUSIONS

In practice, it has been proved that the design using RFID middleware technology can greatly reduce the cost of RFID read-write device, and rapidly enlarge the RFID industry scale. At the same time, this design can promote the formulation of RFID technology standards with independent intellectual property, which has great significance.

This research based on RFID technology can be widely used in Internet industry, and can play an important role in the improvement of industrial transformation and upgrading, construction of industrial information, city intelligence, intelligent operation and management in the future.

RFID technology contains the huge market efficiency, which is expected to billions of dollars annually in size. Now, this system has achieved sales of 8 million Yuan.

ACKNOWLEDGEMENT

FUNDING PROJECT: Science and technology project of Qingdao Development Zone in 2013 "Design of RFID middleware architecture for the application in the advanced manufacturing industry and Internet of Things" (No. 2013-1-29).

REFERENCES

- [1] Zhu, Z. Tan, J. Ren, H. Ni, W. Guan, Q. 2012. RFID application in manufacturing: A case study on a pilot

- RFID project in household appliance production. *International Journal of Computer Integrated Manufacturing* 25(1): 3-10.
- [2] SeokChae, H. Park, J.G. Cui, J.F. Lee, J.S. 2010. An adaptive load balancing management technique for RFID middleware systems. *Software Practice and Experience* 40: 485-506.
- [3] AshadKabir, M. Han, J. Hong, B. 2015. Reader level filtering for efficient query processing in RFID middleware. *Journal of Network and Computer Applications* 48: 58-70.
- [4] Abad, I. Cerrada, C. Cerrada, J.A. Heradio, R. Valero, E. 2012. Managing RFID sensors networks with a general purpose RFID middleware. *Sensors* 12: 7719-7737.
- [5] Chaqfeh, M.A. & Nader, C. 2012. Challenges in Middleware Solutions for the Internet of Things. In: *Collaboration Technologies and Systems (CTS), 2012 International Conference on. IEEE, pp. 21-26.*
- [6] Tian, W. Xue, R. Dong, X. Wang, H. 2013. An approach to design and implement RFID middleware system over cloud computing. *International Journal of Distributed Sensor Networks* 13: 1-13.
- [7] Fescioglu-Unver, N. Choi, S.H. Sheen, D. Kumara, S. 2014. FID in production and service systems: Technology, applications and issues. *Information Systems Frontiers* 08: 1-12.
- [8] Hu, Y. Su, W. Huang, Y. Chen, I. Kuo, S. 2013. Dependable architecture of RFID middleware on networked RFID systems. In: *Green Computing and Communications (GreenCom), IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing. IEEE, 2013, 1881-1884.*
- [9] Torres, R.C. & Martins, L.E.G. 2014. NFR Catalogues for RFID Middleware. *Journal of Computer Science & Technology* 14(2): 102-108.
- [10] Shin, D. Oh, D. Ryu, S. Park, S. 2014. A smoothing data cleaning based on adaptive window sliding for intelligent RFID middleware systems. *Research on Intelligent Information* 20(3): 1-18.