Research and Design of RFID Middleware on Hierarchy

Dou Yewei[1]　Yin Ye[1]　Wang Jianwei[2]

([1] *College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai* 200234, *China*
2 *Shanghai Zhen Zhuo Electrical Technology Ltd. Co., Shanghai* 201100, *China*)

Abstract It briefly describes the radio-frequency identification technology (RFID) middleware and its current research, based on the study and analysis of the limitations and inadequacies of the existing food tracing system; it designs a layered RFID middleware architecture, and conducts an in-depth study. Layered middleware shields the differences of different devices, centralizes to manage distributed reader devices, according to a protocol rule to filter large amounts of the redundant data; and extract semantic information using publish-subscribe model for the more efficient, real-time information sharing mechanism.

Key words food traceability; reader; radio-frequency identification technology; middleware; filtering

OCIS codes 230.2090; 230.0250

射频识别技术中间件分层结构的研究和设计

窦业伟[1]　殷　业[1]　王建伟[2]

([1] 上海师范大学信息与机电工程学院，上海 200234；[2] 上海真灼电子技术有限公司，上海 201100)

摘要 简单介绍了射频识别技术(RFID)中间件及其研究现状，在研究和分析了现存食品追溯系统的局限和不足的基础上，提出设计了分层的 RFID 中间件体系结构，将中间件模块分成三层，并进行了深入的分析，此分层的中间件模型屏蔽了不同设备的差异性，集中管理分布的读写器设备，并按照一定的协议规则对大量的冗余数据进行过滤筛除，并从中提取语义信息；提高了各个企业、不同厂家产品的兼容性，更有利于应用于食品追溯系统中；利用发布-订阅模型实现更加有效、实时的信息共享机制。

关键词 食品追溯；读写器；射频识别技术；中间件；过滤

中图分类号 TP391　　**文献标识码** A　　**doi**：10.3788/LOP49.052303

1 Introduction

As the increasing of food supply chains and, unsafe aspects run through the entire process from production to processing, packaging, storage, transport and sale of the food. Food security crises occur frequently, affect people's physical health, and social stability. Food security has aroused widespread concern in the world. Setting up the food tracing system covering production, processing and distribution in all aspects for food safety is particularly important.

Radio-frequency identification technology (RFID) is abbreviation of radio-frequency identification technology, which developed rapidly in recent years internationally, and it is a non-contact automatic identification technology. Compared to the traditional automatic identification technology such as bar code,

收稿日期：2011-10-14；**收到修改稿日期**：2011-12-14；**网络出版日期**：2012-03-22

基金项目：国家自然科学基金(60971004)和用于科技馆的实时定位系统关键技术研究及应用示范项目(09511503400)资助课题。

作者简介：窦业伟(1987－)，女，硕士研究生，主要从事射频识别、中间件和食品追溯方面的研究。E-mail：douyeweishmily@163.com

导师简介：殷　业(1961－)，男，博士，副教授，主要从事语音和射频识别方面的研究。E-mail：yinye@188.com

magnetic stripe (Card), IC card, RFID has the advantage of recognition distance, the small volume, the storage of a large amount, strong environmental suitability, and so on. Its application in food traceability is increasingly widespread, and gradually becomes the important means for optimizing food production, ensuring food safety. For example, the United States government has declared that national cattle for food must be equipped with electronic tags[1]. After using RFID system to track the animals, you can get meat traceability information in a food supply chain. Various parts of China have also established monitoring system of pork based on RFID. Through RFID ear tag recording on the pig′s ear, it can record feed, medical records, drug, and quarantine information of pigs[2,3].

However, the food traceability system based on RFID involves the production, processing, circulation and marketing enterprises, and the enterprises use a variety of labels and readers. The data structures provided are also different, traced information are different too. And for most enterprises, from front-end data collecting to back-end business systems attaching, they mostly use custom software development. Once the front-end tags, types of readers increase, or the back-end business systems changes, you will need to rewrite the program. Development efficiency is low and high maintenance costs. In this context, RFID middleware came into being.

2 Introduction of RFID middleware

RFID middleware is between the front-end reader hardware modules (from the tag to the reader) to the back-end database and application software. It provides many software modules, such as reader management, data filtering and collection, event management, security management and network management. It has many kinds of implementations that can meet a variety of interfaces and protocol specification[4]. RFID middleware shields the diversity and complexity of RFID devices, provides powerful support for the background system, so as to drive broader, richer RFID application. In other words, RFID middleware provides a two-way transparent interface. On one side it shields the differences of the RFID hardware device as label, reader, and operating system, database, provides reliable protection of high security, high performance, high scalability, manageability and so on. On the other side, it provides unified data extraction interface for the upper business logic implementation, and itself is to achieve a common RFID data processing.

Although many countries are committed to develop RFID middleware software, there are deficiencies and limitations more or less. For example, thin middleware function, simple filtering rules[5], and the definition of complex event processing capabilities are missing. The divided definition of connection the interface is not quite clear. Middleware is not open; platform-dependent is too strong. It is difficult to use the other platform to integrate with existing legacy systems. Event integration layer is centralized, it is hard to improve on the performance and system redundancy.

RFID Middleware can shield the differences between various tags and reader. Through filtering it can process the original tag information, and can also offer large amounts of data sharing in real time to other partners in trace chains. Now all the major manufacturers of RFID middleware products are based on their research and development of core technologies, greater dependence, less expansion. Therefore, through the analysis of establishing the RFID middleware model, this letter designs the model to be used in food traceability[6]. It can reduce the investment and maintenance costs of enterprise in RFID application; ensures effective, real-time sharing of information on the traceability chain between the various links; enhance the traceability of the system.

3 Design for RFID middleware system

Middleware system architecture designed in this is divided into three layers, followed by the reader control layer, data processing layer, and business information sharing layers from bottom to top. As Fig. 1 shows, the reader control layer is at the bottom. It is responsible for controlling different models of readers distributed in different places. Then upward is the data processing layer. It is responsible for the processing original event and data from reader control layer. The top layer is the business information sharing layer,

which receives event information from the data processing layer, then transmit and publish them to upper-layer application system.

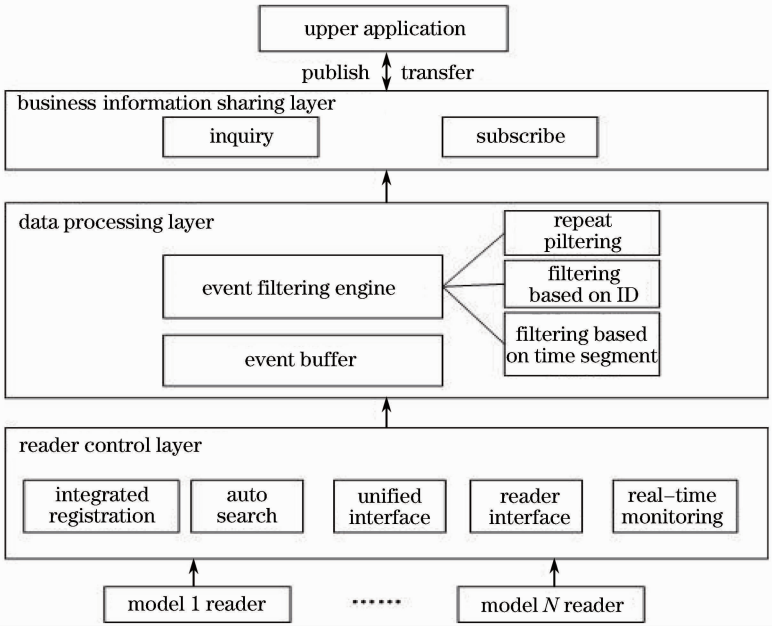

Fig. 1 RFID middleware system architecture

3.1 Reader control layer

Reader control layer is the foundation of entire middleware architecture, used to shield the differences between manufacturers, types of readers, to provide an accurate real-time original data to data processing layer. It mainly has the following features:

1) Integrated registration: you can manually enroll different reader distributed in various scenes to middleware, including deploying reader's property parameter, application and location, name and logo, reading/writing operations to the reader. We should also configure the connection parameters for the reader, for example, the interface types, the number of antennas, IP address, port number, user name, password and other information[7]. When the registration is successful, the middleware can achieve to control over the reader, ensuring reliable real-time information exchange between the reader and the reader control layer.

2) Automatic search: users can manually register the reader to the middleware, and middleware can also be used for the automatic search feature. In the specified interface search area, the middleware scans if there is one new reader device coming in. When it finds a new reader device, it would read some connection parameters, such as the IP address, port number, and manually configure the properties parameter. Then the new reader can be registered to the middleware[8].

3) Unified interface: in large scope of RFID deploying applications, due to the differences of the scenes, we have distinct needs of readers, such as a role of distance, bands, performance and other indicators. So it is likely to use different types of readers that come from various manufacturers. Each kind of reader has its own means of communication and communication protocol formats, creating hardware diversity. In order to shield the hardware differences, reader driver needs to be managed together. This module is based on an internal communication format of middleware, to allow unified, extensible interface. When you access a new reader, it only needs to copy the reader driver which achieves a unified interface to specified directory. And after simple configurations, you can achieve reader connection.

4) Reader interface: reader interface is the core of reader control layer. It uses different models of reader drivers that have been loaded, to truly achieve data transmission. Reader interface includes two parts: reader adapter and reader interface.

Different types of readers collected data formats are different. Reader interface transforms the data in different formats into a unified, systematic format. Reader adapter plays a crucial role in the process

(Fig.2). For different types of readers, middleware has a corresponding adapter, which can dynamically increase or delete in middleware. Reader adapter can communicate directly with the reader, collecting information on the label, and then the adapter can format the data as needed to transmit to reader interface.

Finally the data out of the reader interface are formatted original data, so that the underlying different types of readers are completely transparent to data processing layer.

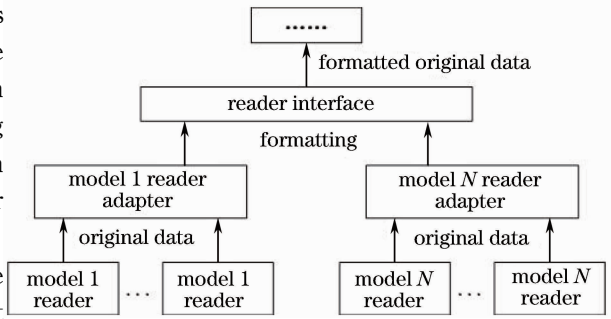

Fig.2 Reader interface diagram

5) Real-time monitoring: for registered reader devices, middleware can not only control their start, stop, test interface, but also monitor the readers' status of running, errors of antenna connection, RF/ interface and other issues. By real-time monitoring to reader devices, it can realize the remote operation of the equipment on the one hand. On the other hand it could find problems arising in the course of the equipment in time in order to respond immediately.

3.2 Data Processing layer

Original data has a large scale, much redundant information and may contain potential virus, less semantic information included, and other issues. To solve these problems, data processing layer deals with the raw data for buffering, filtering, aggregation. It provides appropriate sizes of semantic information to business information sharing layer. Data processing layer consists of two parts: the original event buffer and the original event filtering engine.

3.2.1 The original event buffer

A reader can collect hundreds of labels of data per second, so when multiple readers work at the same time, they can read thousands of labels per second, the data flow generates very quickly. How to deal with these huge amounts of data in real time is the problem that RFID middleware needs to deal with. Currently memory database becomes the best solution to solve this problem. In order to process large quantities of events data more efficiently, reduce the frequent operation to the back-end database in data processing, and lessen the back and forth transmission in the network because of storage and inquiry, middleware draws the data to the memory database before submitting them to the disk storage. Such treatment has hundreds or even thousands of times of higher efficiency than traditional operation.

3.2.2 The original event filtering engine

The work principle of RFID reader is: reader scans its own antenna radiation zones periodically, reads event of labels and sends the event to middleware. The minimum period of reader identifying electronic label and forwarding to middleware is different with the difference of readers' performance, but usually at 100 ms or so. If in the food inventory management system, food carrying electronic label through a portal need 10 s, then there will be 100 events to be sent to the middleware per second. If there are foods that carry hundreds of electronic tags through a portal, there will be 10000 events sent to the middleware per second at the same time[9]. If the events of readers are delivered to the middleware directly without any purification, there will be a great burden on middleware. And other studies are also shown: RFID labels may be infected with a virus that virus can be transmitted and affect background database, and can easily spread to other RFID labels. Therefore, we must provide event filtering for RFID middleware, filtering virus and redundant information in order to reduce the amount of events.

a) Repeat filtering: repeat filtering is the most widespread, commonly used method of filtering, which has the functions of streamline redundant data, smoothing oscillator reading.

There are two kinds of reasons to cause repeated data: in the short term, one reader repeatedly reads the same label; more than one neighboring readers repeatedly read the same label. There is a missing rate for readers, which has the relationship with location of reader antenna, the distance of readers and goods, texture of goods and other items. Usually in order to ensure the read rate, we may place more than one

reader adjacent in the same place. So in multiple reader antenna coverage area, a label is likely to be read by multiple readers repeatedly, creating duplicate data . Repeated filtering idea is to compare to the timestamp of many events for the same reader card number, if the timestamp of the adjacent records is small enough. You think it is repeated reading data, and we excludes the latter. And so on, out of all of the repeated data[10].

b) Reader identifier-based filtering: in many cases user may only want to get events from a specific reader collection. This filtering ensures that events outside of this collection are filtered out. A condition for this type of filtering used is that of course reader has set its own unique identifier like a label. Otherwise, this method cannot be used. There are also the same filtering principles based on specific locations, specific enterprise and so on.

c) Based on time segment filtering: this method of filtering is based on the timestamp of the event, where we choose a fixed point of time, the event before or after is removed or retained.

3.3 Business information sharing layer

After processing by the data processing layer, original event is converted to acceptable information, and then it is provided to upper applications layers or partners with multiple connection methods. It is usually centralized connection: HTTP, JMS, RMI, Web Service or centralized MQ (Message Queue). In food safety traceability system, it involves many businesses of the production, circulation, processing, marketing. The way of centralized delivery of information increases the burden of a server on the one hand, and makes a property bottleneck; on the other hand, if an enterprise wants to get more partners' information service from traceability chain, you have to maintain a connection with each service, the scalability is not enough[11].

To solve these issues, we use the publish-subscribe model (abbreviated as pub-sub) in the business information sharing layer.

Pub-sub model is a loosely coupled communication for distributed computing environments. In pub-sub system, publishers and subscribers are associated by subject, they do not have to know where the other party is, and online at the same time is unnecessary, achieving communication multi-dimensions coupling of both time and space and data traffic. Information publisher sents the message with "event" form to event agent; information subscriber issues a condition of subscription to event agent, expressing interesting in particular kinds of events in the system. And event agent guarantees to deliver all published events to all interested subscribers timely and reliably. Pub-sub system enables the producers and consumers are fully coupled in three aspects of time, space, and control flow, which can properly meet the needs of large scope, highly dynamic distributed systems, such as RFID middleware.

As Fig. 3 shows, we design a distributed event delivering agent service (DEDAS). The port of generating message simply publishes event information to the DEDAS, then DEDAS sends copies of the event to users that are interested in this event. Any enterprises on the traceability chain can publish RFID event information to DEDAS, and can also subscribe interested event information to the DEDAS, keeping on real-time sharing of information effectively in the chain among enterprises.

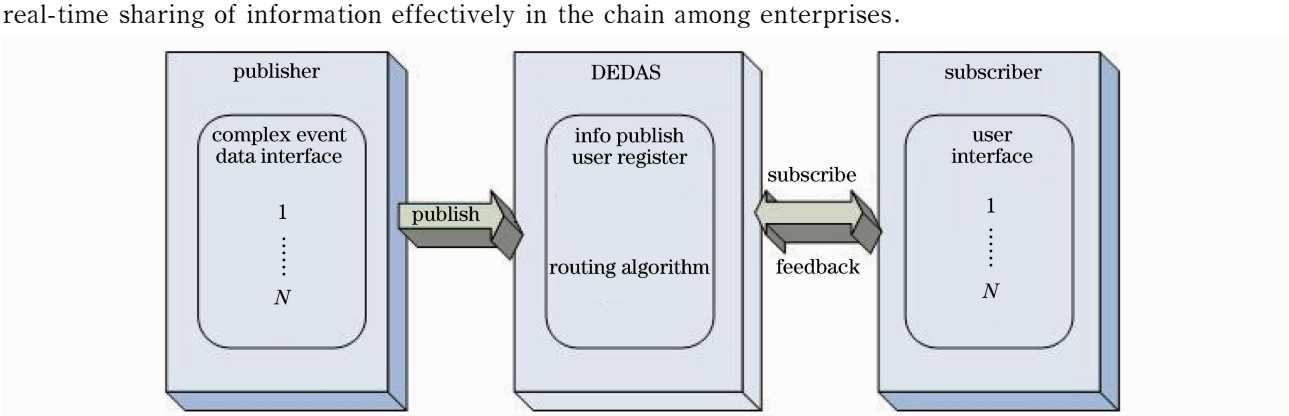

Fig. 3 DEDAS work diagram

4 Conclusion

Because of many benefits of RFID technology[12], it has increasingly broad application in food traceability. But due to the incompatible and diversity of tag and reader, and the differences between the data structure of various enterprises on the traceability chain, and the limitations of traditional information sharing, there is a certain degree of restraint to the development of RFID technology in the further of the food traceability system. Appearance of RFID middleware is a very good solution to solve these problems. It can shield the diversity of different hardware and data structure and provides real-time sharing of event information. It becomes the most favorable weapon for the development of RFID to overcome barriers. Based on this architecture to construct food traceability system, we could reduce RFID application investment and maintenance costs for food enterprises, and ensures effective, real-time sharing of information on the traceability chain between the various links.

References

1 Sun Jian, Chan Qi. Development situation of RFID middleware in China and in the world[J]. *Logistics Technol. & Appl.*, 2007, **12**(2): 98～101

2 T. Kim, H. Kim. Authorization Policy for middleware in RFID system consumer electronics [C]. 2006 IEEE Tenth International Symposium on Consumer Electronics, 2006. 1～4

3 Lorraine S. Lee, Kirk D. Fiedler, Jeffery S. Smith. Radio frequency identification implementation in the service sector [J]. *Internat. J. Prod. Econ.*, 2008, **112**(2): 587～600

4 Park, Sung-Mee Song, Jeong-Hwan. Load Balaneing Method Using Connection Pool in RFID Middleware [C]. 5 th ACIS International Conferenee, Aug, 2007. 132～137

5 T. Dimitriou. A Lightweight RFID Protocol to Protect Against Traceability and Cloning Attacks [C]. Seeurity and Privacy for Emerging Areas in Communications Networks, Athens Greece, 2005. 59～66

6 Lin Wang, Bryan A Norman, Jayant Rajgopal. Placement of multiple RFID reader antennas to maximize portal reed accuracy [J]. *Internat. J. Radio Frequency Identification Technol. and Appl.*, 2007, **1**(3): 260～277

7 Xingxin Gao, Zhe Xiang, Hao Wang. An Approach To Security And Privacy Of RFID System For Supply Chain[C]. IEEE International Conference on E—Commerce Technology for Dynamic E-Business (CEC-East′04), 2004. 164～168

8 J. Ha, S. Moon, JMG Nieto. Low-Cost and Strong-Security RFID Authentieation Protoeol[M]. Lecture Notes In Computer Seience. Heidelberg: Springer Berlin, 2007. 795～807

9 A. Bechini, M. Cimino, F. Marcelloni *et al.*. Patterns and technologies for enabling supply chain traceability through collaborative e-business[J]. *Info. & Software Technol.*, 2008, **50**(4): 342～359

10 K. Framling, T. Ala-Risku, M. Karkkainen *et al.*. Design paterns for managing product Life cycle information[J]. *Commun. ACM*, 2007, **50**(6): 75～79

11 J. Kim, K. Jeon, D. Jeong *et al.*. An ontology-based RFID system model for supporting semantic consistency in ubiquitous environment[C]. In Processing of Computational Intelligence and Security, 2006, **1**: 510～513

12 Fan Wenbing, Cao Xiaoguang, Chen Yan. RFID middleware standardization and implementation[J]. *Microcontrollers & Embedded System Appl.*, 2008, (1): 16～18