TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Transactions of the Institute of Measurement and Control
1–11
© The Author(s) 2017
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0142331216684550
journals.sagepub.com/home/tim
SAGE

Implementation of radio frequency identification middleware with database

Bocheng Chen[1], Andy Mak[2], Fengqun Lin[1], Bo Yuan[1], Wenhuang Liu[1] and Hongyang Wang[2]

Abstract

This paper introduces the principle of a radio frequency identification (RFID) middleware, including its characteristics, problems and implementation in relation to Electronic Product Code (EPC). The targeted users are small and medium-sized domestic enterprises (SMEs), who need flexible and convenient solutions. This paper provides a framework and construction solutions of this simple RFID middleware, and discusses EPC concepts, solutions and advantages. It has been found that the best solution is to construct a simple and convenient platform and continuously add new common reader adapters based on the actual needs and extend special applications, which will become the basic functions of the RFID middleware, so that SMEs can conveniently use and extend these functions according to their needs. This study shows that the simple RFID middleware is suitable for SMEs to use and the open source can be easily applied to make a platform based on the RFID middleware.

Keywords

Middleware, RFID, system design, tags, RFID application

Introduction

The basic principle of radio frequency identification (RFID) is as follows. When an RFID tag has entered the magnetic sensing field, a reader emits an RF signal and the tag receives the signal and then sends the production information stored in a chip by virtue of the induced current energy, or initiatively sends the signal of a particular frequency. After the information is read and decoded, the reader sends the information to a central information system for data processing (Fan et al., 2008). With a normal RFID device, however, there are the following difficulties: (1) Adapting different types of RFID readers and RFID protocols; (2) Utilising distributed systems and adapting networks of different structure; (3) Meeting the needs of the application environment at different levels; and (4) Adapting the needs of different network hardware and operating system platforms (Chen, 2006; Michael et al., 2010).

The possible approach to solving the above problems is to use an RFID middleware, which plays an intermediary role between RFID tags and application programs (Tajima, 2007). It can be connected to an RFID reader and used to read RFID tag data using a common application programming interface (API) in the application program. Therefore, even if the number of database software or the application program storing the RFID tag information increases, or they are replaced by other software, or the number of types of RFID readers increases, the application end can process these data without modification. In this way, it is no longer necessary to maintain the complex many-to-many connections (Pan, 2007). Because of the important role of RFID middleware in an RFID system, researchers have paid great attention to study RFID middleware for various practical RFID applications (Cardiel et al., 2012; Kabir et al., 2015; Tian et al., 2013; Tounsi et al., 2016).

As an RFID works in a multiple reader and protocol environment, it is important to use a complex full-featured middleware, which meets the international standards and can manage a comprehensive range of readers. In addition, the software must be adaptable to a variety of environments. At present, popular RFID middleware products are constructed based on EPC global standards. For practical applications, only a small number of RFID readers from well-known manufacturers are commonly used (Finkenzeller et al., 2010). It is noted that:

- While the EPC standards are very important, many of its complex applications are not essential, and it is more convenient to use simple types in system integration;

[1]Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[2]Shenzhen Xunliu Company, Guangdong, Shenzhen, China

Corresponding author:
Bocheng Chen, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China.
Email: chenbch@mail.tsinghua.edu.cn

- From the development point of view, the users like to connect and communicate with different systems in an early stage of developing software, but the software will eventually become redundant;
- From the practical need point of view, SMEs like simplified concepts or solutions to reduce the cost and reliance on external services (Fu and Jiang, 2012).

Because the application of RFID middleware is complex, it cannot be used after a reader is connected. Under the actual management conditions of SMEs, it is important to provide simplified solutions. From the system application point of view, an RFID middleware needs to address the following issues (Chen, 2006):

- Compatibility with interfaces of different readers (connecting ports of RFID reader and computer). There are many RFID system interface modes, such as RS232, RS485, USB and Ethernet (RJ45);
- Identification of the structures of different tag memories for effective read and write operations. Each reader has its own API, which may be written in different languages, with different functionality, performance and other aspects;
- Compatibility with interfaces of different application programs, which need to use RFID tag data in the application system. It is necessary to be compatible with RFID devices from different manufacturers and for different applications so that data can be read from the RFID devices;
- Ensure data acquisition, filtration and transmission. Any error or information disclosure from acquisition to transmission may cause problems for users (Fabian et al., 2012).

An RFID middleware should be designed and implemented in a simple way to solve the above issues. For a specific use, common application program interfaces should be defined and corresponding international standards should be complied to provide a unified interface for various application systems and reduce differences between various types of hardware (such as readers).

Major software companies have proposed the overall solutions of RFID software, including middleware. They have also suggested unified management for RFID-related software and hardware based on middleware. Although these products are powerful for large enterprises for facility management, the requirements of RFID reading and writing equipment in enterprise management cannot be met by simply applying these management products. Because these products often manage certain RFID-related hardware and software facilities, most of them do not give a suitable solution for simple applications, while SMEs often need flexible and scalable management solutions that can be adapted to the middleware equipment layer for common RFID reading and writing. Therefore, we propose a simple RFID middleware solution, which uses less than 20% of EPC system functions and has a low-cost compact system relative to the Application Level Event (ALE) structure of EPC (EPCglobal Inc., 2005), for the convenience to use.

Design and implementation

The development software used in the solution is an open-source, as the middleware system construction is, with easy and fast installation, and simple and convenient management. A user needs to establish the connection between reader and the MySQL database and the connection between the MySQL database and information system. The subsequent system maintenance and upgrading are simple. If a new type of reader is used in the system, it can be incorporated into the RFID system by writing a corresponding adapter program.

Characteristics of simple RFID middleware

The solution differs from the EPC system middleware in the following aspects:

- Interface. Either the standard or compatible EPC middleware provides application programs for API, XML, SOAP, socket connection, TCP connection and other data transmission modes so that different users can select appropriate connections according to their own circumstances, and to achieve the connection between reader and application program by using various interfaces provided by MySQL.
- Supported formats. While the EPC middleware only supports GID-96-format tag ID, the new solution can support several popular formats, such as ISO18000-6B, ISO18000-6C (EPC C1G2), and can identify active tags. In other words, this new middleware can expand its functions to achieve the functions of EPC middleware and extend its application range for compatibility with more tags.
- EPCIS-related. The EPC middleware supports EPCIS, and is usable owing to registration. In the new solution registration is not required and hence EPCIS cannot be used. Instead, database queries are mainly conducted by the program to match and obtain production information.
- Filtering. The EPC middleware can subscribe for information in the EPC network; that is, subscribe different types of data according to different needs. The EPC network can send a report after conducting filtering. For example, the returned report may indicate the number of mobile phones rather than the ID number of the tag, reading time and other information corresponding to the number of mobile phone products that can be subscribed. The solution does not have this function because it does not require registration. However, the data can be filtered easily because the read data is very detailed and filtering can be achieved by a database look-up table.
- Applicable extension. With clear structure and convenient transformation, SMEs can program the RFID middleware, requiring less maintenance and convenient for extension.

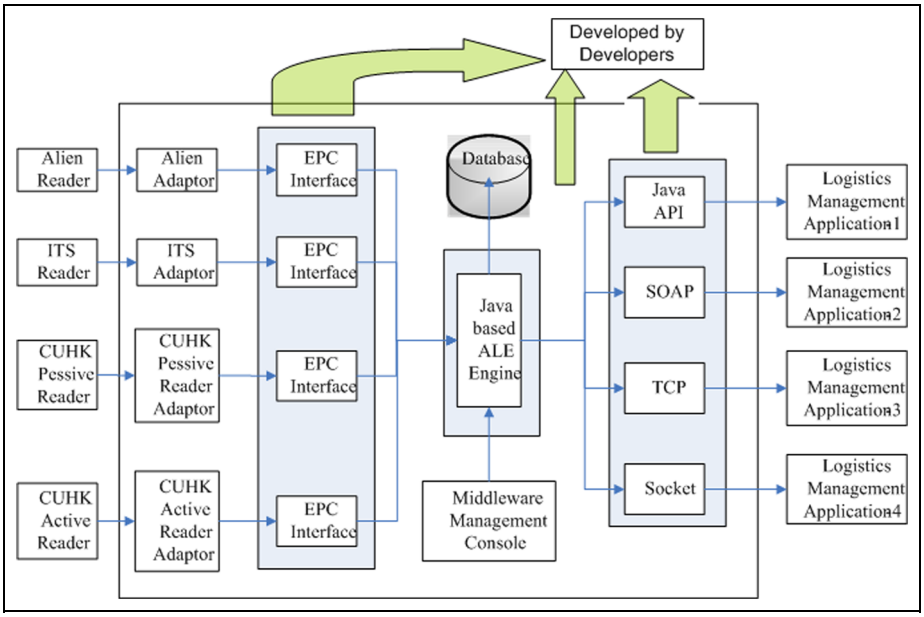

Figure 1. RFID middleware system developed by Hong Kong Chinese University (Mak et al., 2007).

System structure design

The integration point of this solution is the MySQL database because it can provide interfaces of a variety of programming languages. Almost all applied languages support it, including PHP, VB, VC and Java. Therefore, RFID devices from different companies can be connected to the database through the corresponding interface. For example, the Alien Company mainly cooperates with the Sun Company to develop systems. The Alien reader uses driving, reading and stored programs written in Java. Motorola mainly cooperates with Microsoft. Its reader uses the driving program written in C. These companies provide API in their programming languages. With this new middleware, however, the adapter program is written in the same language for use of these APIs. Similarly, the adapter program can connect with the information system database using the interface function of the corresponding language provided by MySQL to input the read tag data from the reader into the applicable-layer database. In addition, it can join with various application programs of the information system (Lin et al., 2010).

Figure 1 shows an RFID middleware system structure developed by Hong Kong Chinese University (Mak et al., 2007), which is constructed based on EPC. With this system, it is difficult to implement system integration and upgrading. The upgrade of each development platform may require changes of the whole system. Figure 2 shows the new solution (Mak et al., 2012). It can be seen that the system construction and upgrading entail the upgrading of database programming and MySQL database software.

In Figure 2, the connection of RFID middleware with the upper application and lower reader is in the vertical direction. The middleware can integrate the application programs for other users; that is, connect to the middleware of various users in the horizontal direction. The middleware is essentially the software connecting two or more application software packages together. It connects business application programs among enterprises, and enables them to transmit data to each other. For example, when a middleware connects with an EDI port, the transaction information regarding goods can be directly sent to other companies transacting with the company (Chalasani and Boppana, 2007). Thus, the middleware application is convenient, with no need to change the original system of the enterprise, and the data transfer process is simplified. Figures 1 and 2 also show the differences between the MySQL database integration and the RFID middleware of EPC solution.

In the new solution, the MySQL database exists as a temporary database because it is an open-source and many people are familiar with its programming, can save read data in real time, and upload to the enterprise database after the data are processed using a given filtering algorithm. The database is used as a development tool, and enterprises can use different database software to construct the integration point according to the solution and their own circumstances. For example, the database used by the original information system may be used in an Oracle and SQL Server.

In the application, a temporary database is set up and the records of the database are compared with a permanent database. A large number of data from all adapters are stored in the temporary database for filtering, and so the management is efficient. Because a temporary database is a warehouse where data are continuously stored, the expansion speed for the amount of data is fast. Because most of the data are useless afterwards, an enterprise may empty the temporary database daily, weekly or monthly, depending on the specific application environment, to maintain high operation and management efficiency.

Data storage, transmission and filtering algorithm design

The memory capacities of readers are often different because they are produced by different manufacturers, and anti-

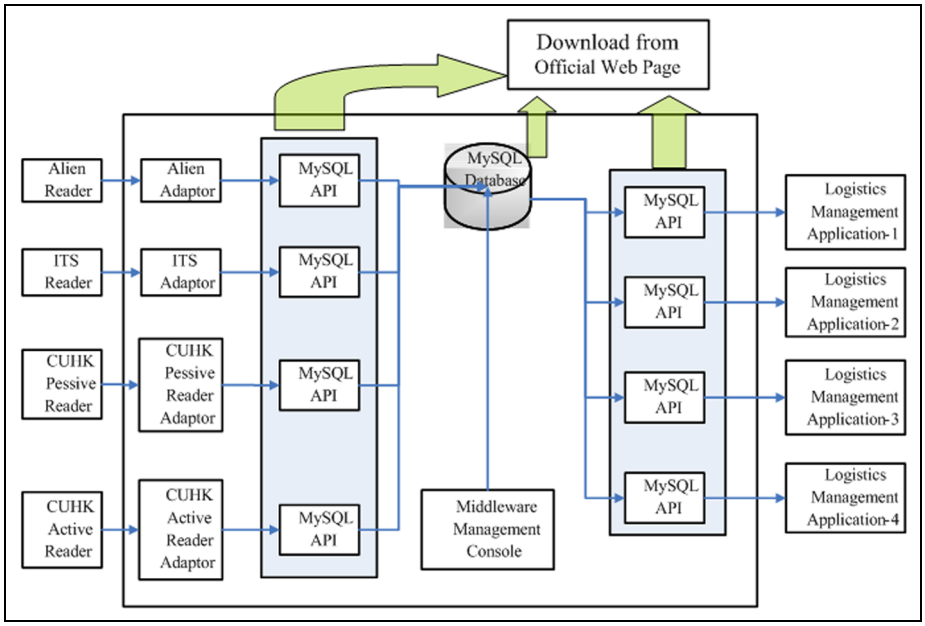

Figure 2. System structure of simple RFID middleware with database (Mak et al., 2012).

collision algorithms are also different (Chen et al., 2010; Ferrero et al., 2012). Some memories are very small, and can upload the data when a tag is read, and no filtering is required. The reader of the Motorola Company has a large memory space, and caching and filtering are first done in a reader. When the application program sends the command "read", the reader will upload the filtered data; that is, each ID number is read several times in the read cycle. Because the bandwidth in the air is limited, the number of times that each tag is read is relatively small if the number of tags is large, while the number of times each tag is read is relatively large if the number of tags is small. The number ranges from several times to several dozen times. The data structure of the reader uploaded each time in the system is considered as follows.

In addition to the read tag ID number, the mentioned number of tags being read is included to facilitate application management and data processing. Other data should also be provided based on the user's needs and the application, such as the antenna number of the tag, number of reader, type of tag and the time when the tag is finally seen (the time when the tag is read).

The antenna number and reader number shows the position of the tag. Tag type is used to describe the type of tag. "The time when the tag is read" describes the time attribute of the tag. If the tag is read in the read cycle, the attribute is read 0. If the tag is not read after 10 seconds, it is read 10. The time when the tag is finally read can be calculated compared with the present time. The advantages to set this attribute are: (1) the read tag is displayed; (2) assuming that "the time when the tag is read" is taken as the attribute, the tag record will disappear when this tag cannot be read. After this time point, it will not be known when it will disappear and if it ever exists. If this tag is again read over a period of time, it will not be known whether it is a new tag or a previously read tag, and thus it cannot be known when it is read. It is noted in the experiment that sometimes the reading may be intermittent when using the RFID reader features; that is, not all tags will be read in each reading owing to the location and distance, tag orientation, material, accidental event and other reasons. Only a field is needed to solve the problem when using "the time when the tag is finally read", while at least two time fields are needed to illustrate the event when using "the time when we read the tag" as the attribute.

To develop filtering algorithms for the simple RFID middleware, the users can first construct a main framework of system filtering with the most commonly used filtering algorithms and later increase special algorithms according to their needs. Commonly used filtering algorithms include multi-read tag, misread tag and data merging (Fan et al., 2012; Mahdin and Abawajy, 2011):

- Multi-read tag. A tag ID appears only once or twice, and then never appears again. There are a number of reasons for this situation. For example, a tag passes through the edge of the RF range of an antenna. The reader is just able to recognize this tag, but the tag does not belong to the warehouse. It only passes through the periphery. In this case, the ID number should be removed; that is, it should not be allowed to enter the database. An enterprise can set a threshold to determine whether or not to filter this multi-read tag depending on the application. For example, it can be set that a tag should be filtered out if it is read less than three times.
- Misread tag. A tag has some reading records, but the tag is not read at some point or within a relatively short period of time. In Figure 2, reading records of an RFID reader is shown, where six tags are within

the read range of the reader. The vacancy shows that not all tags can be detected in one scan. Two reasons leading to an error reading are also described in the diagram, specifically interference issues and air interface conflict (Floerkemeier and Lampe, 2005). In this case, these two tags always exist, and they do not leave the RF scope. Thus, a threshold can be set. For example, it is considered that the tag does not leave away from the database as long as the number that the tag is misread is smaller than five or it can be read within five seconds. In comparison, the tag is considered not within the range if it is not read after a relatively long period of time. To prevent misreading, the method of redundancy of simultaneous monitoring (such as optical-electronics technology) of other equipment is usually used in the processing line.

- Data merging. Because it is easy to read the tag, the data read should be transmitted after merging rather than each time after reading to reduce the redundancy of data transmission. Taking a library as an example, most books are on the shelf or flow in the library rather than being borrowed, and so most of the IDs are always read. In this case, a filtering parameter can be set and the data merged within a period of time. For example, the read data can be merged within 10 seconds, which may consist of 10 records merged into one record, and the time can be set to when the data is read for the last time.

Compatibility with active and passive tags

Different companies support different development languages. In addition, the tag types of different companies may differ, such as active and passive tags, with different management protocols, such as ISO, EPCglobal and UID. The data formats of different types of tags are also diverse, such as ISO tags with different specifications, including ISO18000-6B and ISO18000-6C, which have different specifications based on, for example, memory capacity such as 512 bits, 2 Kb, 96 bits of EPCglobal and 128 bits of UID. Different formatted data must be transformed into a unified format that can be identified by the management information system after being read. The MySQL database is used as a temporary database, where the read data in various formats are first added and then the untransformed data are regularly transformed into a readable format.

The commonly used tags comply with either the ISO18000-6B protocol (train, car, container identification) or the ISO18000-6C protocol (i.e. EPC logistics field, such as goods circulation of Wal-Mart), and they are all passive tags. The front ID parts of the two tags are basically the same. The main difference is that 6B has an additional user area compared with 6C, which is used to store the user information, such as the company address and phone number. The simple RFID middleware can identify two different tags, and it can input the read data into the MySQL database.

At present, there is no common standard for active tags. In general, each active tag company has its own rules to support the system operation and conduct encoding and decoding. Active tags are usually used in the circulation process of goods that need to be monitored (such as vegetables and meat) or valuable articles, and active tags are attached with other information in addition to an ID number. For example, temperature should be controlled during meat transport, and active tags can be used to record the temperature and change in temperature in real time during the whole transport process. Another example is that active tags can be installed on locks and used to monitor containers by the tag's memory recording whether the container has been opened, which can be expressed by 0 or 1. Although there is no standard for active tags, the new system can conduct identification and data management in a closed loop application environment (Ondemir et al., 2012; Kelepouris et al., 2007). Therefore, the simple RFID middleware can be easily made compatible with a variety of active and passive tags.

Extension of system functions

Concepts to conduct extension by direct use of database

The temporary database should be extended to a temporary database system to adapt to more types and tags, increase in flow speed, and other special requirements of RFID reader. Figure 3(a) shows a temporary database system including temporary databases 11, 12 and 13, which are connected together according to a tree-type topology. Temporary database 11 is at the bottom layer and is connected to the adapters of the corresponding readers after being classified based on the physical characteristics of the RFID reader. Temporary database 13 is on the top layer and is regularly sends updated data to the primary enterprise management information system database. Temporary database 11 regularly sends its updated data to temporary database 12 in the next layer, which are connected to each other and under trigger control. Temporary database 12 regularly sends its updated data to temporary database 13 in the next layer, which are connected to each other and under trigger control. This concept is not limited to three-layer tree-type topology. Two-layer or multi-layer tree-type topology may be flexibly selected according to the user needs. However, too many layers should be avoided in a practical application.

Figure 3(b) shows a temporary database system including temporary databases 11 and 13, which are connected according to the star-type topology. Temporary database 11 is on the bottom layer and is connected to the adapters of the corresponding readers after being classified based on the physical characteristics of RFID reader. Temporary database 13 is on the top layer and is used to regularly send updated data to the enterprise management information system primary database. Temporary database 11 regularly sends its update data to temporary database 13 in the next layer, which are connected to each other and under trigger control.

In a manufacturing assembly line, there would be many tags to be read (Lu et al., 2006). To prevent misreads or multi-reads, redundant monitoring can be used. For example, data from the assembly line material may be read with two types (Alien, Motorola) of readers (totally three) and antennas from

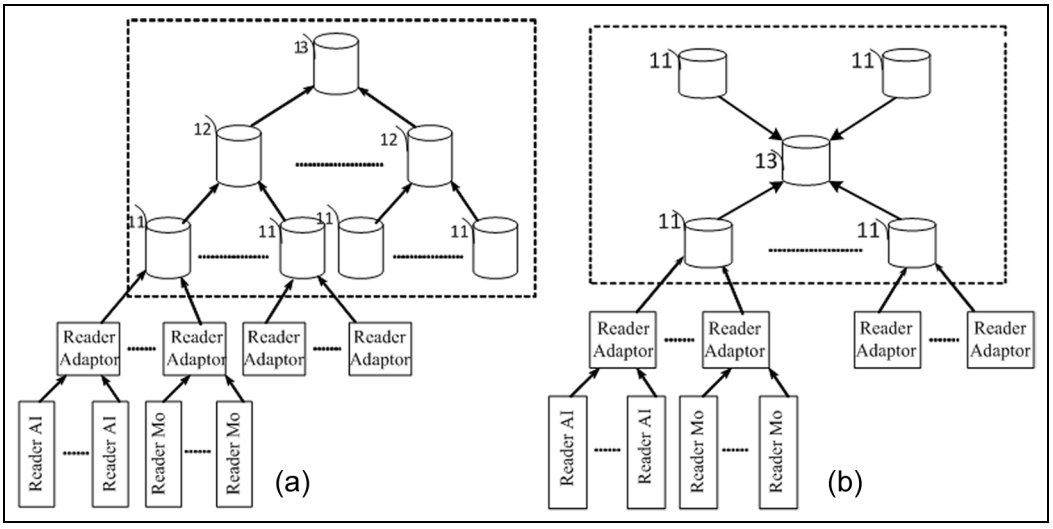

Figure 3. Expansion plan of topology.

different positions and at different angles. The data read by each reader are respectively stored in their own temporary database. The system can filter each temporary database at each sampling time point, and upload it to the primary temporary database

A judgment (two-out-of-three method) is increased in the filtering algorithm when combining the data. For a tag, if there are data in two temporary databases but no data in one temporary database, this tag exists. If there are data in one temporary database, but no data in two temporary databases, this tag does not exist. If the amount of data is small, these data are input into a temporary database for filtering and there are no problems in the reader number and antenna number. In a practical application, if a tag is not read by one of the readers for some reason (e.g. it is blocked), but it is read by another two readers, the tag is considered having been read, so that some misread problems can be solved. If a foreign tag is read near one of the reader antennas, but is not read by two other readers in the same group owing to distance, it will be filtered out in the process of being transmitted to the primary temporary database, so that part of multi-read problems can be solved. Filtering algorithm suggestions can be achieved easily by monitoring along the assembly line by implementing database extension. If other methods are used in the manufacturing assembly line for auxiliary monitoring (such as optical signal), the monitoring signal should be read in the database in the form of special attributes to participate in the filtering algorithm judgment. It is not difficult to achieve the operations with good reliability.

Concepts to conduct extension with logical reader and physical reader

In the EPCglobal middleware specification, only a reader carrying an IP address is counted as a reader. For a simple RFID reader, which operates on COM Port (Serial Port) only, a computer is needed to control it, and the IP address is actually carried in the computer itself. Usually, logical reader refers to the networked PC connecting with the RFID reader, while physical reader refers to the actual reader hardware. Where the RFID reader comes with a built-in LAN port, it has both functions. On the other hand, changes occur in the system design with the concept of logical reader and physical reader, because there are many kinds of connection protocols for logical reader and physical reader, including 1-to-1, 1-to-M, M-to-1 and M-to-M:

- One-to-one connection. As shown in Figure 4(a), one-to-one connection is the simplest way because a logical reader corresponds to a physical reader. The logical and physical readers are combined with a dotted line to show their roles as logical reader and physical reader at the same time. Current readers on the market are generally such readers.
- 1-to-M connection. With a single physical reader connecting to multiple logical readers, a reader can be shared, as shown in Figure 4(b), among several subscribers; for example, headquarters, regional headquarters and local office in a particular read zone.
- M-to-1 connection. With many physical readers connecting to a single logical reader, a subscriber can logically extend the coverage; for example, a common stock reader with four antenna ports. If a user wants to cover a particular read zone with eight or even 12 antennae to guarantee the reading rate, then the readers can be set up in an M-to-1 configuration, as shown in Figure 4(c).
- M-to-M connection, as shown in Figure 4(d), is basically a mix of scenarios (2) and (3) above to provide the highest level of flexibility.

Of course, with a more complete RFID middleware, customization will be more flexible using the M-to-M structure. However, it requires the user interface to be simple and reasonable to adapt to user application customization.

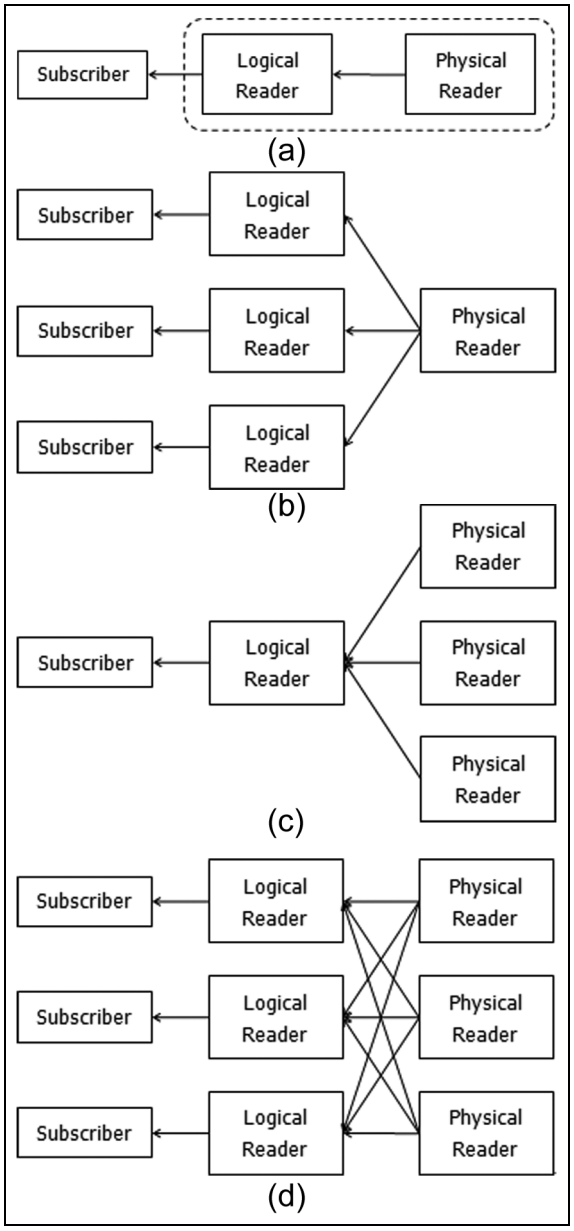

Figure 4. Different connection protocols.

Otherwise, problems may occur. Some examples can be given for the function extension of related software in the structure of the solution. The biggest feature is that a programmer familiar with the MySQL database and proficient in programming can quickly implement and effectively maintain all extensions. Having understood the principles, integration points and system structure, an SME can make a simple RFID middleware to meet their own requirements, and extend the system structure according to their further practical needs.

Extension of simple RFID middleware by hardware

The above concepts based on software extension can be applied to hardware extension, as the management of system hardware is more flexible and the network construction of ordinary PCs is convenient (Ismael et al., 2012). A light-weight RFID middleware based on database implementation and RFID application subsystems are set up on both middleware and management computers, and a networked RFID system is constructed through network connections, and tag data interaction can be achieved as follows.

The middleware computer calls the API functions of the corresponding RFID reader through its data extraction module, extracts the tag data detected by the RFID reader, and writes the read tag data into the temporary database system through calling the functions of its temporary database system. Based on the request of the management computer or the triggering of the middleware computer, the distributed data transfer management module transmits the tag data in the specified time period of the temporary database system to another temporary database of new read tags of the corresponding management computer via the internet for the primary database and information processing module of management computer for processing. The direct extension of hierarchical structure with hardware is shown in Figure 5(a) (Mak et al., 2012). Another form of direct extension of hierarchical structure with hardware is shown in Figure 5(b) (Mak et al., 2012), where the computer on the bottom layer directly sends data to the temporary database (it may come with a temporary database, and send the data out from the temporary database). Therefore, this system structure mode cannot manage numerous computers on the bottom layer in a practical application. The direct extension of star-type structure with hardware is shown in Figure 5(c) (Mak et al., 2012). Owing to the convenience in construction of the network with computer hardware, the reduction of cost and the existing complete network management functions of open-source software MySQL, based on the working mechanism, it is easy for a software engineer familiar with MySQL programming to implement these structures.

More complex application extension of simple RFID middleware

Based on the system mechanism and by the use of open-source software MySQL, the system can be easily extended. The structure of our network-based RFID middleware for an experimental site is shown in Figure 6 (Mak et al., 2012), where the small box stands for another data collection point, which is away from the temporary database system. Because the data collection point and the temporary database are not connected by the Ethernet, the data from the wireless router are read with a long-distance wireless network card for processing. With this structure, the simple RFID middleware has been successfully operated in a lab environment.

In a practical application, this simple RFID middleware based on the open-source software MySQL can be extended according to requirements. A software engineer familiar with MySQL programming can easily complete the operations because the number of changes required is small in relation to the software. The software and hardware expansion modes of the simple RFID middleware discussed above can be achieved

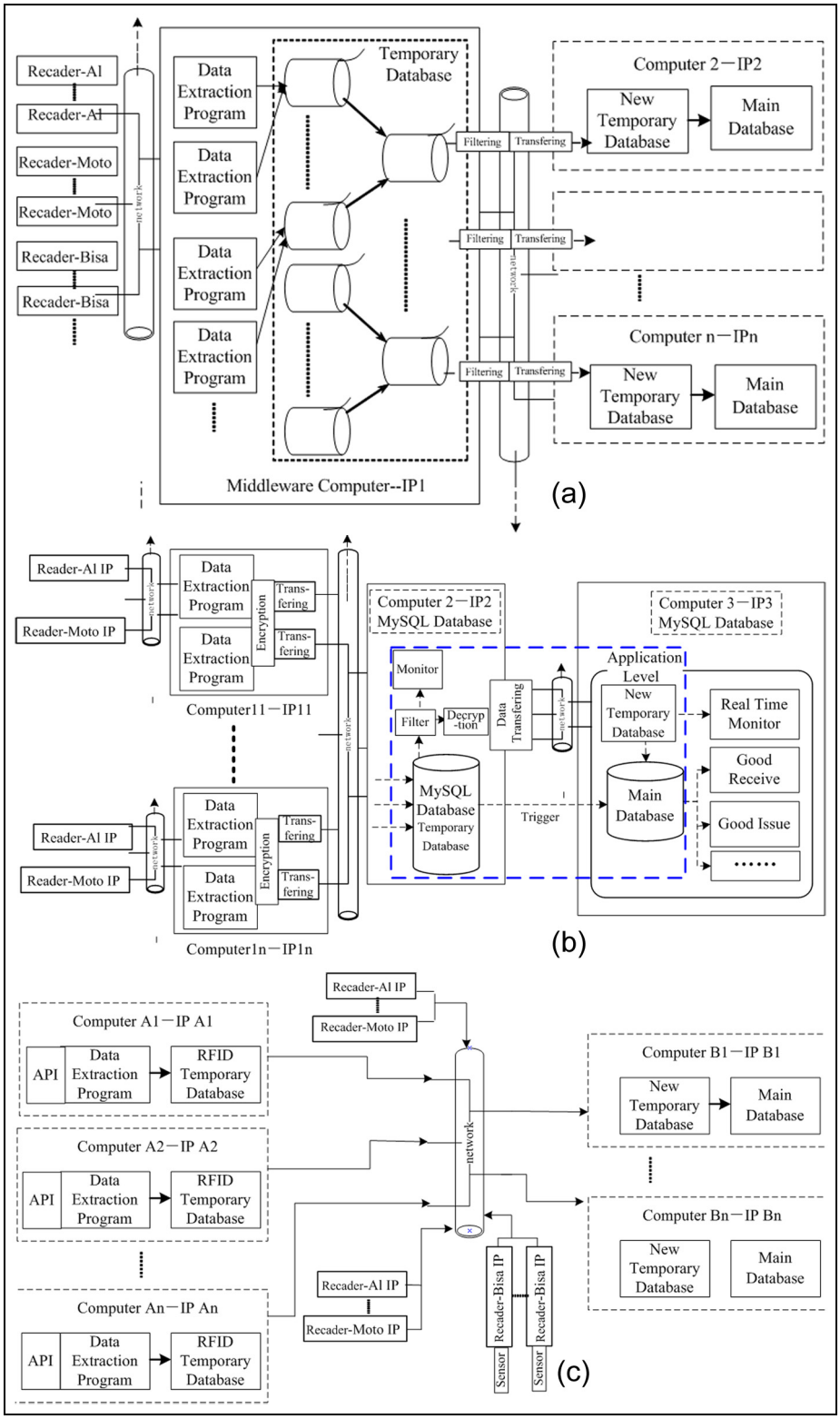

Figure 5. Direct extension with hardware.

based on the mechanism of this study, providing convenient and alternative solutions to the application of RFID for manufacturing assembly lines.

As an example, Table 1 gives some data recorded by such a system, where ALN is the Alien RFID Reader, Moto is the Motorola RFID Reader, EMU is a simulation signal, and

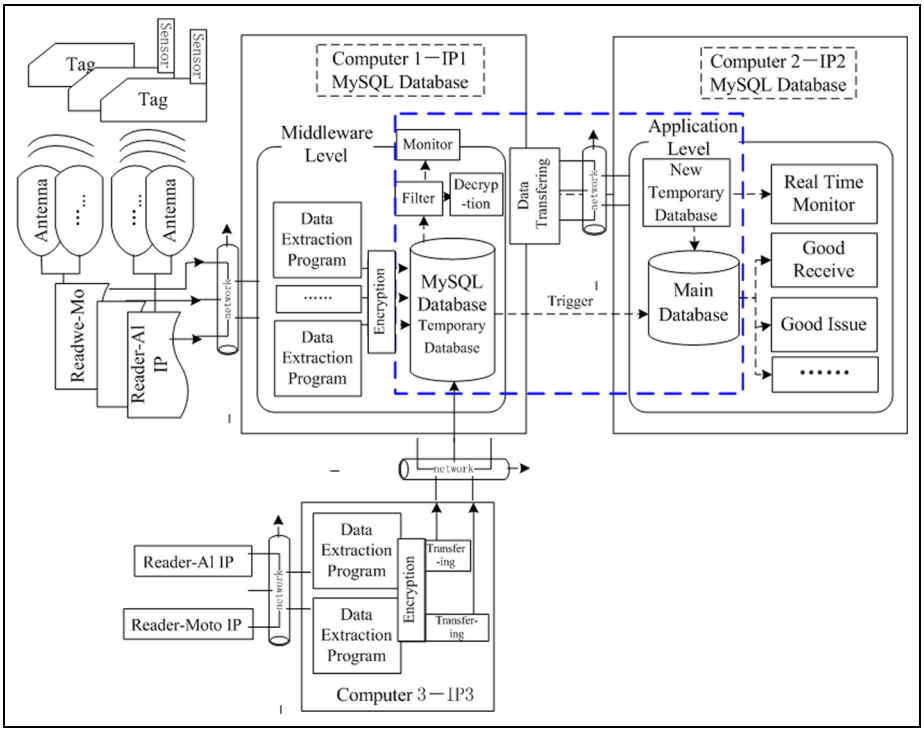

Figure 6. Complex application extension of simple RFID middleware.

Table 1. Data from readers in MySQL database (temperature: 25.9°C).

No.	Reader type	Antenna No.	rfid Id	Read time
252575	ALN	1	E200341 1B802011207172569	2009-05-20 16:49:30
252576	ALN	1	E200341 1B802011207172571	2009-05-20 16:49:30
252577	ALN	11	E200341 1B802011207172576	2009-05-20 16:49:30
252578	ALN	1	E200341 1B802011207172560	2009-05-20 16:49:30
252579	ALN	11	E200341 1B802011207172565	2009-05-20 16:49:30
252580	ALN	11	E200341 1B802011207172572	2009-05-20 16:49:30
252581	ALN	1	E200341 1B802011207172561	2009-05-20 16:49:30
252582	ALN	11	E200341 1B802011207172577	2009-05-20 16:49:30
252583	BISA	1	4300000000001368	2009-05-20 16:49:30
252584	EMU	0	35000006F0000DE00000014D	2009-05-20 16:49:30
252585	MOTO	1	E200341 1B802011190364623	2009-05-20 16:49:31
252586	MOTO	1	E200341 1B802011207172579	2009-05-20 16:49:31
252587	MOTO	1	0006000000000000000000DBA	2009-05-20 16:49:31
252588	BISA	1	4300000000001371	2009-05-20 16:49:31
252589	BISA	1	4300000000001362	2009-05-20 16:49:31
252590	EMU	0	35000006F0000DE00000014D	2009-05-20 16:49:31
252591	MOTO	1	E200341 1B802011190364623	2009-05-20 16:49:32
252592	MOTO	1	0006000000000000000000DBA	2009-05-20 16:49:32
252593	MOTO	1	E200341 1B802011207172579	2009-05-20 16:49:32
252594	BISA	1	4300000000001375	2009-05-20 16:49:32
252595	BISA	1	BBBB00000372	2009-05-20 16:49:32
252596	ALN	1	E200341 1B802011207172562	2009-05-20 16:49:32
252597	ALN	1	E200341 1B802011207172578	16:49:32

BISA is a kind of Active RFID system with a temperature sensor on its tag. This result demonstrates that the system works.

Conclusions and discussion

There are some important issues related to the domestic use of RFID middleware, including how to use RFIDs from a

few large international companies, and how to assist SMEs to understand the principle, and to implement the structure and simplify RFID middleware. This paper introduces the principle of RFID middleware, describes the characteristics of RFID middleware, analyses some problems with RFID middleware in implementation of EPC-based RFID middleware, and how SMEs can have flexible solutions. This paper provides the framework and construction solution of a simple RFID middleware and the solution to the EPC concepts. Some new common reader adapters can join a simple platform continuously based on actual application needs. Some special applications can be extended to ensure that they have the basic functions of the middleware. SMEs can conveniently use and extend these functions according to their needs. The common open-source approach can be easily applied to make the platform. The features of this simple RFID middleware can be summarized as follows:

- No EPC official registration. Members of the EPC system, including hardware and software companies, consulting firms, systems integrators, training companies, manufacturers, retailers, wholesalers, carriers and government departments, need to register with the EPC system and pay an annual membership fee. If SMEs or new enterprises only conduct local management, it may not be necessary for them to register.
- No need to integrate the EPC Information Service (EPCIS) or Object Name Service (ONS). EPCIS is an EPCglobal network service, and business partners can exchange EPC-related data via the internet using the service. Also, ONS is not required in the solution, which is a system designed by the Auto-ID Center and used to query unique EPCs and direct the computer to commodity-related information based on the code; that is, match the EPC and application commodity information. These functions can be completed in their own database by updating the read data to the information system database and matching the appropriate commodity information through ID numbers in the database.
- Low requirements for computer hardware. Middleware suppliers of overall solution products need to configure appropriate hardware devices because of the limit in computational power of normal PCs. For example, BEA/MS BizTalk is not available for download, and it is difficult to install them onto notebooks. With the proposed RFID middleware, however, a PC or a notebook can be used in the solution, manage and control the whole RFID system because of its simplicity. Therefore, the solution is not only suitable for SMEs, but also provides users with the opportunity to understand RFID.
- Easy to learn and can be constructed by the user. To use an RFID, a user can achieve the read and write operations for an RFID tag after the relevant language compiler software and MySQL database are installed, requiring a low technical skill. The solution can also be used as a design framework to provide a quick-to-use tool for users, so that they can be familiar with the RFID middleware.
- The software and hardware can be extended conveniently. In addition, the encryption may be constructed and perfected according to specific needs if necessary.

Declaration of conflicting interest

The authors declare that there is no conflict of interest.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the State Key Lab for Manufacturing Systems Engineering Foundation, Xi'an Jiaotong University (No. Sklms2011010), the Shenzhen Key Lab for Space Robot Technology and Tele-Science and Natural Science Foundation of Guangdong (No. 2015A030313881).

References

Cardiel IA, Gil RH, Somolinos CC and Somolinos JC (2012) A SCADA oriented middleware for RFID technology. *Expert Systems with Applications* 39(12): 11115–11124.

Chalasani S and Boppana RV (2007) Data architectures for RFID transactions. *IEEE Transactions on Industrial Informatics* 3(3): 246–257.

Chen LH (2006) Design and research on key technique of RFID Middleware. MSc Thesis, Department of Applied Computer Technology, Guangxi University, China.

Chen YH, Horng SJ, Run RS, et al. (2010) A novel anti-collision algorithm in RFID Systems for identifying passive tags. *IEEE Transactions on Industrial Informatics* 6(1): 105–121.

EPCglobal Inc. (2005) The Application Level Events (ALE) Specification, Version 1.0, RFC 5134.

Fabian B, Ermakova T and Müller C (2012) SHARDIS: A privacy-enhanced discovery service for RFID-based product information. *IEEE Trans. Industrial Informatics* 3(3): 707–718.

Fan H, Wu Q and Lin Y (2012), Behavior-based cleaning for unreliable RFID data sets. *Sensors* 12(8): 10196–10207.

Fan WB, Cao XG, Chen Y and Cao X (2008) Middleware standardization and implementation. *Micro-controllers & Embedded Systems* 2008(1): 16–18.

Ferrero R, Gandino F, Montrucchio B and Rebaudengo M (2012) A fair and high throughput reader- to-reader anticollision protocol in dense RFID networks. *IEEE Transactions on Industrial Informatics* 8(3): 697–706.

Finkenzeller K and Muller D (2010) *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. Chennai: Wiley.

Floerkemeier C and Lampe M (2005) RFID middleware design – Addressing application requirements and RFID constraints. In: *Proceedings of 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*. Grenoble, France, October, pp. 219–224.

Fu YB and Jiang PY (2012) RFID based e-quality tracking in service-oriented manufacturing execution system. *Chinese Journal of Mechanical Engineering* 25(5): 974–981.

Ismael A, Carlos C, Cerrada JA, et al. (2012) Managing RFID *sensors* networks with a general purpose RFID middleware. *Sensors* 12(6): 7719–7737.

Kabir MA, Han J and Hong B (2015) Reader level filtering for efficient query processing in RFID middleware. *Journal of Network and Computer Applications* 2015(48): 58–70.

Kelepouris T, Pramatari K and Doukidis G (2007) RFID-enabled traceability in the food supply chain. *Industrial Management & Data Systems* 107(2): 183–200.

Lin FQ, Chen BC, Yuan B, et al. (2010) Composition and implementation of RFID lightweight middleware. *Computer Engineering* 36(17): 77–80.

Lu BH, Bateman RJ and Cheng K (2006), RFID enabled manufacturing: Fundamentals, methodology and applications. *International Journal of Agile Systems and Management* 1(1): 73–92.

Mahdin H and Abawajy J (2011) An approach for removing redundant data from RFID data streams. *Sensors* 11(10): 9863–9877.

Mak A, Lam A and Qu DM (2007) CUHK RFID middleware-system design document. Available at: http://mobitec.ie.cuhk.edu.hk/rfid/middleware/docs.htm

Mak A, Chen BC, Liu WH and Wang HY (2012) Networked RFID systems and data interacting method based on a lightweight middleware. Chinese Patent 102053962B.

Michael K, Roussos G, Huang GQ, et al. (2010) Planetary-scale RFID services in an age of uberveillance. *Proceedings of the IEEE*, 98(9): 1663–1671.

Ondemir O, Ilgin MA and Gupta SM (2012) Optimal end-of-life management in closed-loop supply chains using RFID and sensors. *IEEE Transactions on Industrial Informatics* 8(3): 719–728.

Pan XJ (2007) The research of data transmission in RFID Middleware. *Science and Technology of West China* 2007(2): 43–45.

Tajima M (2007) Strategic value of RFID in supply chain management. *Journal of Purchasing and Supply Management* 13(4): 261–273.

Tian W, Xue R, Dong X and Wang HY (2013) An approach to design and implement RFID middleware system over cloud computing. *International Journal of Distributed Sensor Networks* 2013(10): 1–13.

Tounsi W, Cuppens-Boulahia N, Cuppens F and Pujolle G (2016) Access and privacy control enforcement in RFID middleware systems: Proposal and implementation on the fosstrak platform. *World Wide Web-Internet and Web Information Systems* 19(1): 41–68.