Research on RFID Integration Middleware for Enterprise Information System

Li Minbo
Software school, Fudan University, Shanghai, 201203, China
Email: limb@fudan.edu.cn

Li Hua
Software school, Fudan University, Shanghai, 201203, China
Email: 082053018@fudan.edu.cn

***Abstract*: Radio Frequency Identification (RFID) is a promising technology for automated non-line-of-sight object identification. Traditional research of RFID middleware does not concern RFID related integration with enterprise information systems. Considering the requirements of RFID middleware and business process integration with enterprise information systems, an integration framework for RFID middleware based on business process rule and data stream technologies are introduced. Main modules of the RFID integration middleware, such as devices monitoring, data management, XML business documents exchange, business process control and RFID event management, are designed and discussed in detail. The semantic definition of the complex RFID event and RFID event classification are presented. A real-time scheduling strategy of RFID event which is based on the buffer and priority queue, can process the concurrent RFID events. Application system based on RFID is realized to control the business process through RFID events.**

***Keywords:* RFID; RFID Integration Middleware; Data Stream; Business Process Rule**

I. Introduction

Enterprise applications such as Supply Chain Management (SCM) Systems, Enterprise Resource Plan (ERP) Systems, and Manufacturing Execution Systems (MES) help enterprises to achieve efficiency, reduce costs and increase productivity in their operations. Automatic Identification and Data Capture (AIDC) is a methodology of assigning an identity to a business object and automatically capturing the identity of the business object as it moves within and across an enterprise. Some enterprise applications can generate and aggregate information about raw materials, products, equipment through their identity by supporting AIDC to improve efficiency [1].

Radio Frequency Identification (RFID) is a non-contact technology of AIDC, which identifies objects attached with tags. RFID has been widely applied to various areas such as railway transport monitoring, highway fees and charges, tracking of agriculture product, food, medicine [2]' animal identification, anti-counterfeiting, logistics transportation, etc. Nowadays, low-cost RFID has attracted more and more interests from industry and academic institutes [3]. In supply chain management, RFID tags are used throughout the supply chain to track products, from supplier, deliver, to warehouse stock and retail. Compared with bar-code, which is another popular technology of AIDC, RFID has several advantages, including non-line-of sight reading, the ability of handling serial number, automatic real-time reading, more sensor networks or monitoring systems that flavor to it and so on [4].

ERP system may not be designed to handle the serial-number information from RFID data. It may cost a lot for enterprise applications with multiple identification technologies used but no explicit RFID interface to be rebuilt for RFID Integration. Such limitations of enterprise applications make it a challenging task to integrate RFID to enterprise applications. A graceful way to address this challenge is to introduce a layer between RFID readers and enterprise applications, which has come to be known as RFID middleware.

RFID middleware should meet requirements such as dissemination, filtering and aggregation of RFID data, reading from and writing to a tag, privacy [5]. Besides, some RFID middleware may be able to manage RFID events and to response to data subscriptions from external applications. In traditional RFID applications, there is little need for an RFID middleware because the RFID readers are not connected in a network and the RFID data is only used by a single application. However, it is not the same when RFID should be integrated with enterprise applications, such as supply chain management system. Under that situation, many readers which are distributed across enterprise and warehouse capture RFID data that will be disseminated to a variety of applications [6], and an RFID middleware is needed to manage RFID readers and data from these readers. At present, IBM, Microsoft, Oracle, Sun, SAP, BEA, Sybase and other companies had released their respective RFID middleware which are more platform-dependent and less scalability. Most of these RFID middleware solutions focused on RFID data

Supported by the National High-Tech. R&D Program, China under Grant 2006AA04A133, Email:limb@fudan.edu.cn

doi:10.4304/jsw.6.2.167-174

filtering and ignored the integration between RFID application system and real-time enterprise business handling.

This paper puts forward a RFID Integration middleware (called Rdspor) for enterprise information systems. Rdspor is driven by XML documents from enterprise information systems, uses a rule engine to control data process, and emphasizes on how to integrate RFID with enterprise information systems. Meanwhile, a data stream engine is used by Rdspor to filter and aggregate RFID data. The paper is organized as follows: The related work is described in section II. An overview of Rdspor is presented in section III. Section IV explains the details of the design and implementation. In the last section, we present our conclusions.

II. RELATED WORK

RFID middleware plays a key role to the application of RFID technology. So far, there are many achievements in the aspects of technology research and product development of RFID middleware. For products, as mentioned above, IBM, Oracle, Sun, BEA and other software companies have introduced RFID middleware or RFID solutions. These products or solutions which are developed mainly on existing web application servers or middleware products of the companies are tightly coupled and are heavyweights among the RFID middleware products, and only applies to large and complex RFID application systems because of their lower expandability and high costs. The MIT University, USA together with ETH had proposed an open-source middleware platform (called Accada) for RFID applications. They also conducted some thoroughly studies aimed at issues such as data distribution, data integration, data filtering, the external sensor-driven reader, events packaging, coding management and so on.

The major domestic RFID middleware products include RFID middleware SRM from Shanghai Jiaotong University, GDIX-RFID middleware from South China University of Technology, ezRFID middleware from Tsinghua Tongfang, etc. These products are designed for restructuring, and some of them are based on service oriented architecture (SOA). The research of RFID middleware technology mainly focused on topics such as hardware device integration, RFID event management, RFID information services, architecture of RFID middleware, etc. Zhang Jiehao analyzed the integration technology of RFID devices for middleware such as SNMP, EPC global Reader Protocol and Reader Management Protocol. On this basis, he had implemented RFID hardware device management module which based on the J2EE platform for the RFID middleware SRM of Shanghai Jiaotong University [7]; Yan Guoqing had implemented the multi-protocol RFID reader adaptor under TCP / IP communication protocol by configuring an XML file [8]; Zhao Li studied and designed the RFID middleware event management systems. First, he encoded the valid RFID events under Manchester Encoding, and then post-matches them with business rules after pattern recognition and content filtering. After that, according to the results of post-match, he called the pre-defined SOAP Service in the warehouse system. It is a successful interaction between RFID middleware event management systems and warehouse systems. However, it needs great reconstruction of the warehouse system [9]; Xu Qiang, borrowing the Java event model, had designed and implemented the real-time event management mechanism for RFID middleware [10]; Yang Xiaofeng did some research on reader network management, labels data smoothing, RFID data encryption and compression, RFID event filtering, aggregation, reporting and sharing [11]; Christian Floerkemerier used Java Message Service (JMS) as the way of information exchange between RFID middleware and enterprise information systems when designing RFID middleware [5].

According to the specification of architecture of RFID middleware which was proposed by EPC global, an RFID middleware should implement particular interfaces, such as Application Level Event (ALE) interface and Electronic Product Code Information Service (EPCIS) interface, to be compatible with the EPC Architecture and Protocol (shown in Fig.1).

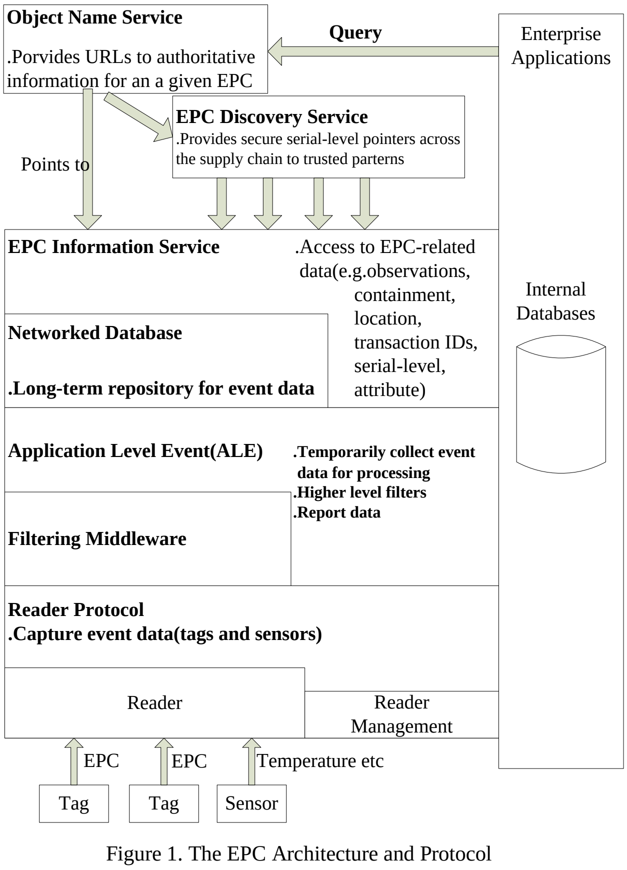

Object Name Service
.Porvides URLs to authoritative information for an a given EPC
Query
Enterprise Applications
EPC Discovery Service
.Provides secure serial-level pointers across the supply chain to trusted parterns
Points to
EPC Information Service
.Access to EPC-related data(e.g.observations, containment, location, transaction IDs, serial-level, attribute)
Internal Databases
Networked Database
.Long-term repository for event data
Application Level Event(ALE)
.Temporarily collect event data for processing
.Higher level filters
.Report data
Filtering Middleware
Reader Protocol
.Capture event data(tags and sensors)
Reader
Reader Management
EPC
EPC
Temperature etc
Tag
Tag
Sensor

Figure 1. The EPC Architecture and Protocol

ALE is an RFID data oriented processing and integration specification that defines how to accumulate, filter, and group EPC data and how to send result in the form of ALE report to the client which subscribes data to ALE, such as enterprise information systems. ALE interface allows applications to access to RFID data at the tag level in a read cycle or an event cycle; meanwhile the EPCIS interface allows applications query RFID events, in order to achieve logistics track and trace. ALE now serves as the standards for EPC data integration interface. However, the goal of integrating RFID with enterprise information systems is to achieve not only data integration but also business integration. In the management of enterprise information systems, there is always a spectrum of purposeful actions, ranging from planning to control. Generally, these purposeful actions are executed within the enterprise information system, but sometimes we need to execute some of them externally. For example, when we are using RFID to receive goods, but the received goods are invalid according to the arrival notice generated by a warehouse system, it may requires the RFID system (or RFID middleware) to determine what to do next, reject the goods directly or just send the invalid data back to the warehouse system and notify it to handle the mess. Some enterprise information systems may not be designed for RFID integration, and it is not easy to solve the issue mentioned above without RFID middleware. For that reason, we proposed a lightweight way of integrating RFID technology with enterprise information system, which focused on both data integration and business integration.

III. SYSTEM ARCHITECTURE

As shown in Fig.2, Rdspor (RFID integration middleware) is consisted of the following five modules:

a) Devices Monitoring and Control Module

①Manages and integrates different RFID devices in a registration way;

②Configure a registered RFID device dynamically according to the business process which is defined in the middleware context;

③Provides Rdspor with a unified interface to manipulate RFID data by sending reading or writing commands;

④Monitors registered devices and if an error or fault is detected, invokes a particular handler to deal with the error or fault.

b) Data Processing Module

①Processes the great number of received redundant data in the front-end of readers;

②Given the RFID data‘s characteristics such as real-time and continuing, combines data stream management technology with top business demands, which include redundant data clean-up, filter and aggregation of data;

③Encapsulates raw data from a reader adapter (will be introduced later);

④Generates fault-tolerant, filtered and aggregated data or events for the follow-up action.

c) Business Process Control Module

①Driven by business documents from enterprise information systems;

②Validates whether the data from readers matches the business documents, proofing the consistency of information flow, and then executes a predefined action, for example, automatically generates a business document which is planned by the enterprise information system and writes it back to the system;

③Captures EPC events and sends them to EPCIS server for data sharing and querying or logistic tracking and tracing.

d) Business Documents Exchange and Interpretation

①Exports or imports business documents from enterprise information systems;

②Sets up the encoding scheme for interpretation of heterogeneous documents from different enterprise information systems since business documents are usually rely on the IDs which are encoded quite differently in different systems;

③Interprets business documents in a set-up encoding scheme, persists the interpreted documents into databases and post them to business process control module.

e) Database Management Module

①Manages internal or external databases, such as database to persist business documents information and database for EPCIS;

②Accepts EPC events from business process control module, and defines the strategy how to manage these events and upload them to EPCIS server.

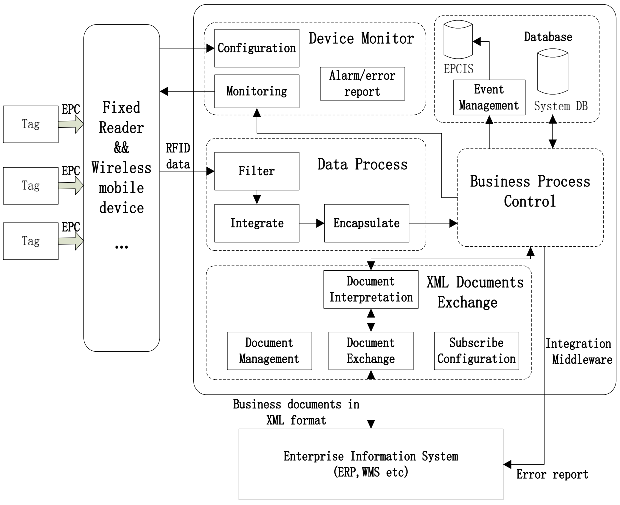


Figure 2. RFID integration middleware architecture

Rdspor is a business document driven RFID middleware and aims to achieve a more effective and loosely coupled way of integrating RFID to enterprise information systems. It accepts a business document and generates another planned document, and tries to float some business processes from enterprise information systems to RFID systems, such as product inbound or outbound in the inventory management of ERP system.

IV. DESIGN AND IMPLEMENTATION

A. *Device Management and Monitoring*

As shown in Fig.3, Device Monitor provides reader adapters to access to physical readers. When deploying a physical reader, Device Monitor will configure the reader’s communication mode and assign it a unique number as identification. A reader adapter may have several physical readers attached to it by maintaining a mapping from the adapter to unique number of physical readers. Meanwhile, a physical reader may be mapped to more than one reader adapter. The reader-adapter mechanism makes it possible for multiple physical readers that are not deployed together to cooperate with each other.

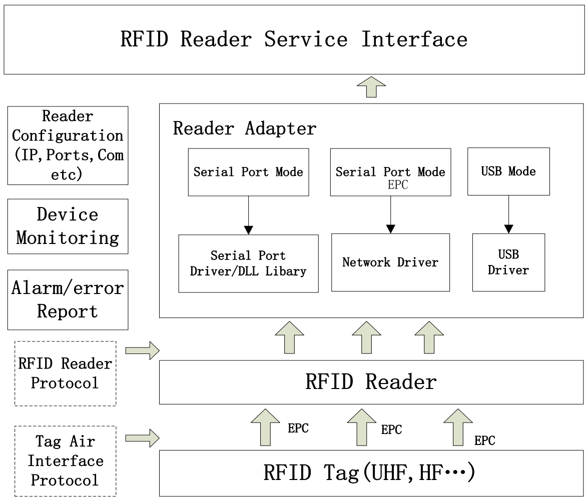


Figure 3. The architecture of Device Monitor

With a reader adapter, a programmer can read and write to a RFID tag without knowing physical readers and RFID protocols such as Tag Air Interface Protocol, which improves reusability. Fig.4 shows a part of the class diagram of reader adapter.

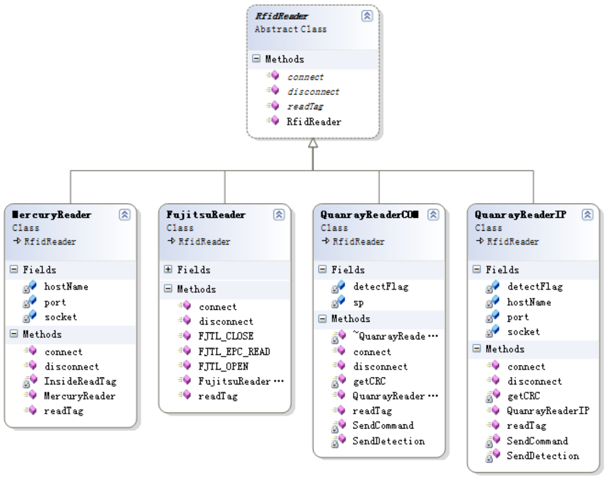


Figure 4. Partial class diagram of reader adapter

B. RFID Data Process

The real time RFID data is collected by readers at a high speed, which means that RFID data will arrive in great volume. Database operation involves frequent disk I/O operations, such as inserting or deleting data, and they have a significant effect on performance of the database management system. As a result, a traditional database may not work very well when dealing with RFID data. On that basis, we use data stream technologies instead of database methods to process RFID data. There are a lot of researches and products in the area of data stream technologies, and it is far beyond the paper to discuss these researches and products. We used an open source data stream engine called Esper to process RFID data in a data stream way.

Fig.5 shows the framework of RFID data process. The business process control module sends reading commands to device monitor, requiring the specific reader adapters to read RFID tags. After the device monitor has completed reading of physical readers attached to the specific reader adapters, it sends RFID data to data process module. In the form of data stream, RFID data will be represented by the data stream event model which has attributes such as EPC to represent the tag's EPC in the RFID data, reader adapter identification to identify the reader adapter that read the RFID data represented by the data stream event, timestamp to imply when this RFID data was generated and a task identification for business process control.

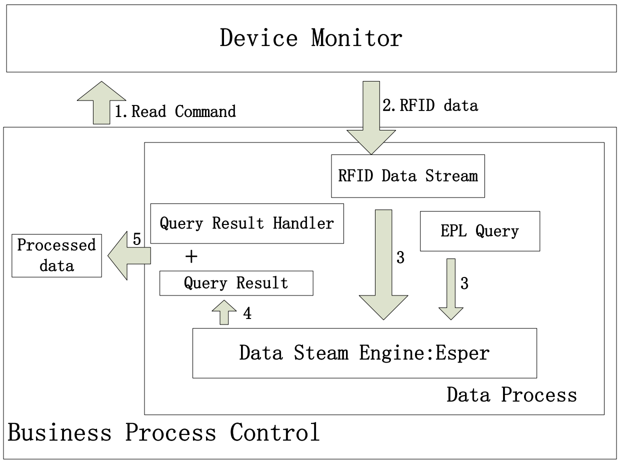


Figure 5. Data process framework

Esper needs to initialize the context before it begins to process RFID data stream. Firstly, it creates an service instance of class EPServiceProviderManager by invoking the class's static method GetProvider (providerURI, configuration), and then it creates queries represented by class EPStatement through calling a factory method named CreateEPL on the service instance.

```
Configuration configuration = new Configuration ();
configuration.addType (
"RfidDataEvent","Rdspor.Model.RfidDataEvent");
EPServiceProviderManager epService =
EPServiceProviderManager.GetProvider("DataFilterEngine",
configuration);
EPStatement statement = epService.CreateEPL(Create EPL);
```

Finally, Esper accepts RFID data stream events as input through the service instance's method sendEvent. In addition, Esper defines an interface named UpdateListener to handle the query result, and an instance of classes that implement UpdateListener should be added to a query.

```
statement.addListener (updateListener);
epService.sendEvent (object);
```

The sequences above can be demonstrated by Fig.6.

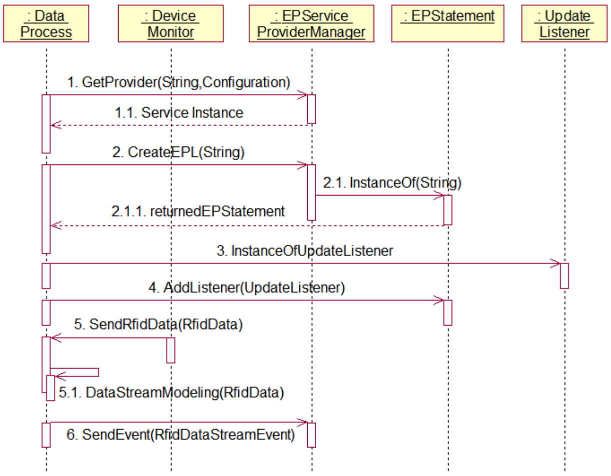

: Data Process
: Device Monitor
: EPService ProviderManager
: EPStatement
: Update Listener
1. GetProvider(String,Configuration)
1.1. Service Instance
2. CreateEPL(String)
2.1. InstanceOf(String)
2.1.1. returnedEPStatement
3. InstanceOfUpdateListener
4. AddListener(UpdateListener)
5. SendRfidData(RfidData)
5.1. DataStreamModeling(RfidData)
6. SendEvent(RfidDataStreamEvent)

Figure 6. Sequence diagram of Esper context initialization

C. *Business Documents Exchange*

Enterprise information systems can provide data integration interfaces, through which external applications can interact with these information systems, such as data exchange of business documents in XML file format. However, documents from different systems are not compatible with each other in document schema and encoding scheme.

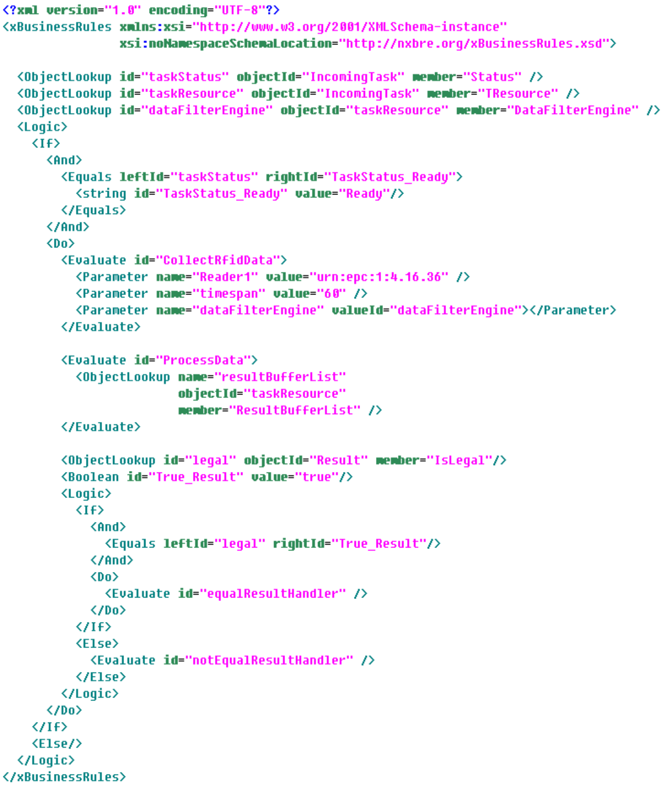
XPath is an effective way to extract information with XML file format, and it bases on the XML document's explicit schema. Since documents from different systems are defined in quite different schemas, a program which designed to process documents of an enterprise information system may not be able to handle documents from another system. Rdspor is a RFID integration middleware for many kinds of enterprise information systems, aims to deal with business documents of various enterprise information systems. It requires a lot of work to write a program for documents from each kind of system, and it also costs a lot to maintain these programs. In order to resolve this problem, we defined several document models, and each model has common attributes to represent a particular kind of documents such as StockIn in ERP system and warehouse system. It also maintained a mapping relationship of each model with every document it represents. A mapping relationship is configured in a XML formatted file, and tells us how an attribute of a model is mapped to the XML node in its represented document. With the relationship configuration file, a program language with reflect mechanism can initialize these model from the documents they represents in a unified way.

The next step of modeling a document is to interpret it. Since the document is rely on IDs in a specific encoding scheme, an attribute of a model mapped to an ID node in the document may be unreadable without the document's encoding scheme or the fundamental data, which is the meta-data in an enterprise information system, such as raw material, department, person and so on in an ERP system. To solve this problem, instead of directly reading data from the database of enterprise information systems, which may cause security issues, we create a local mirror of the fundamental data in enterprise information system, and maintain a one-way synchronization from the fundamental data to its mirror. Such synchronization can be achieved though deploying data-sync software such as OneBridge Mobile Data Suit.

D. *Business Process Control*

Business processes in enterprise information systems are configurable, and may be changed with business environment. The business process control module is designed for changeable business processes based on an open source rule engine called NxBRE. In the NxBRE rule file, we can define rules to define how to collect RFID from device monitor, including which reader adapters to use and how many times each reader adapter should read; how to filter and encapsulate the collected data; how to handle the result. A rule is a component that implements the interface ExecuteRuleInterface in NxBRE and configured in tag node Evaluate of the rule file.

Fig.7 shows an example of NxBRE rule file. This rule file contains four rules which are represented by xml node Evaluate, and some of them are configured with arguments in the context. Take the evaluate node CollectRfidData for example, there are three arguments configured within it. This rule means that NxBRE will collect RFID tag data using reader adapter Reader "urn:epc:1.4.16.36" and use a data filter named dataFilterEngine to complete the data process in a time span of sixty seconds. In addition, the IF-DO-ELSE logic is supported in the NxBRE engine context, which is useful for business control in some cases.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<xBusinessRules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:noNamespaceSchemaLocation="http://nxbre.org/xBusinessRules.xsd">

  <ObjectLookup id="taskStatus" objectId="IncomingTask" member="Status" />
  <ObjectLookup id="taskResource" objectId="IncomingTask" member="TResource" />
  <ObjectLookup id="dataFilterEngine" objectId="taskResource" member="DataFilterEngine" />
  <Logic>
    <If>
      <And>
        <Equals leftId="taskStatus" rightId="TaskStatus_Ready">
          <string id="TaskStatus_Ready" value="Ready"/>
        </Equals>
      </And>
      <Do>
        <Evaluate id="CollectRfidData">
          <Parameter name="Reader1" value="urn:epc:1:4.16.36" />
          <Parameter name="timespan" value="60" />
          <Parameter name="dataFilterEngine" valueId="dataFilterEngine"></Parameter>
        </Evaluate>

        <Evaluate id="ProcessData">
          <ObjectLookup name="resultBufferList"
                        objectId="taskResource"
                        member="ResultBufferList" />
        </Evaluate>

        <ObjectLookup id="legal" objectId="Result" member="IsLegal"/>
        <Boolean id="True_Result" value="true"/>
        <Logic>
          <If>
            <And>
              <Equals leftId="legal" rightId="True_Result"/>
            </And>
            <Do>
              <Evaluate id="equalResultHandler" />
            </Do>
          </If>
          <Else>
            <Evaluate id="notEqualResultHandler" />
          </Else>
        </Logic>
      </Do>
    </If>
    <Else/>
  </Logic>
</xBusinessRules>
```

Figure 7. An example of NxBRE rule file

The framework of business process control module is shown in Fig.8. When a business document from XML documents exchange module reaches, it will be added to the job queue, waiting for processing. The job dispatcher selects an appropriate job and rules in the queue under an explicit strategy, initializes NxBRE context. For example, a resource is allocated, and then start NxBRE engine in a new thread. Once the NxBRE engine begins to process, it will execute the rules defined in the rule file in sequence. When using the Inject Object Container of Spring.NET, the program can be easily to maintain.

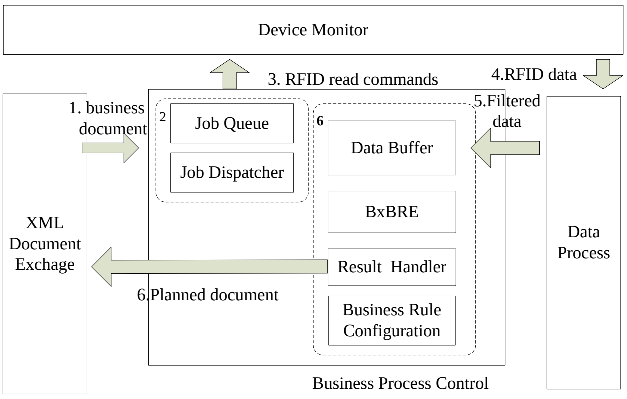


Figure 8. The framework of business process control module

E. RFID Event Semantic Definition

RFID is always used to record and track the product logistics. In order to set up the relationship between product tracking and corresponding logistics information, it need to define the mapping from physical entity identification to corresponding logical data in management system. The traceable unit which can be traced through all or part of its production and supply chain is the item of products, semi-finished products, boxes, trays, containers, etc. The traceable information contains identification definition and recording information. RFID tag is used to identify its uniqueness in the tracking and tracing process. With the recording of RFID identification and logistics information, we can realize the transition from the physical tracing to the information tracing.

The tuple (R, S, T, Step, Rela(S), Order(S)) is used to represent that RFID reader R identify RFID tag S at time T and record the logistics state of the traceable unit. Step of the tuple represents the business type of the traceable unit in supply chain, such as the producing, packing, storage, unpacking, distribution, query, etc. Rela(S) of the tuple is used to identify the relationship between S and the packaging or transport units, such as products associated with the package, packages associated with the tray, and packages associated with the container. Order(S) of the tuple is used to identify the bill number of the corresponding business documents, such as Goods Receipt Notification (GRN), sale bill, transport documents, etc.

In Rdspor, we refer to the EPCIS interface, and use XML schema to describe RFID events in logistics business. Fig.9 shows the five types of RFID complex business events and their elements.

According to the status of product in the supply chain, RFID events are divided into five types which are object event, aggregation event, quantity event, transaction event and query event. The semantic definition of RFID event is shown as the following:

1) Object event represents an event that happened to one or more entities denoted by EPCs.

2) Aggregation event represents an event that happened to one or more entities denoted by EPCs that are physically aggregated together.

3) Quantity Event represents an event concerned with a specific quantity of entities.

4) Transaction Event represents an event in which one or more entities denoted by EPC become associated or disassociated with one or more identified business transactions.

5) Query Event represents an event in which one or more entities enquired by users.

Each of the core event types has fields that represent four key dimensions of any EPCIS event. These four dimensions are: (a) the object(s) or other entities that are the subject of the event; (b) the date and time; (c) the location at which the event occurred; (d) the business context. These four dimensions may be conveniently remembered as “what, when, where, and why” respectively

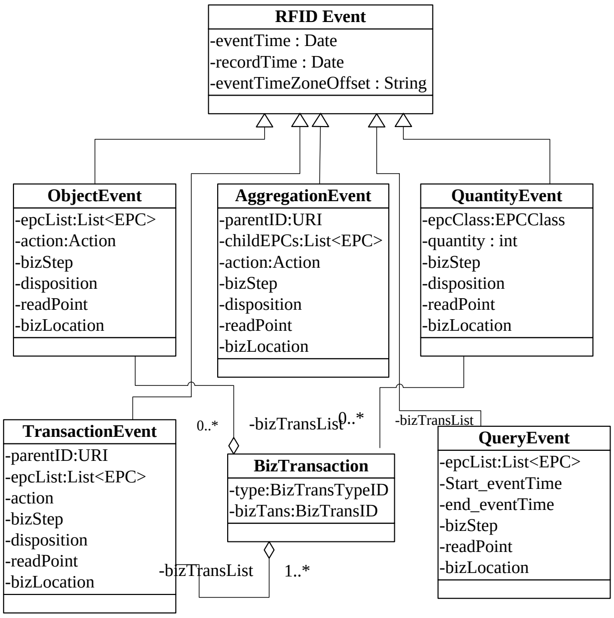


Figure 9. RFID Event types and elements

F. Process mechanism of RFID complex event

As shown in Fig.10, the data process module filters duplicated RFID data, and the business process control module encapsulates filtered RFID data into RFID events according to the business type and transfer them to the buffer pool. Event scheduling module generates the priority event queue from the pending events according to the business rules. The processing component of RFID complex event is responsible for matching the RFID complex events and business documents in order to complete the business operational steps or alarm. Meanwhile, RFID event based on XML format, which contains logistics information, will be uploaded into the EPCIS server. EPCIS server is used to provide the capture interface and query interface for product tracing in the supply chain.

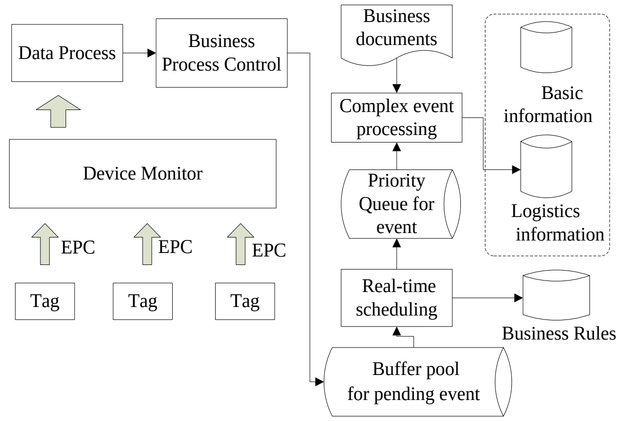


Figure 10. Dataflow process of RFID complex event

In enterprise application environment, there are many RFID readers to collect the logistics information of product. Several RFID events will create simultaneity by different RFID readers. RFID events should be stored in different priority buffer according to their importance, so buffer pool contains a lot of pending events. Real-time scheduling module of RFID event uses pending events to generate the priority queue based on business rules. In the priority queue, all events will be process according to the time. Real-time scheduling strategy includes three methods which are buffer priority processing, business rules scheduling and time sequence order to handling.

The modules of buffer priority processing are as follows:

①Buffer Initialization. We should set the capacity of each buffer, the upper limit threshold, the level of priority and the maximum blocking time;

② Buffer Size Monitor Module. When the buffer reached or exceeded the upper limit capacity, the module will change the priority of this buffer to highest dynamically and also will notify the data processing module;

③Buffer Blocking Time Monitoring Module. When the blocking time exceeds the threshold, the module will change the priority of this buffer to highest dynamically and also will notify the data processing module;

④Buffer Data Processing Module. The module process the buffer according to their priority. After that, the default priority value of each buffer has to be reset.

The buffer with higher priority will be processed first. The buffer priority is decided by the importance of the RFID reader which detected the RFID tag and sent it the buffer.

Business rules are used to process the RFID event according to the predefined business rules. Time sequence processing adopts the FIFS (First In, First Serve) mechanism.

G. Tracing Inquiry

The tracing information of unit S can be expressed as:

$$\mathrm{Inf}（S）=（\sum_{i=1}^{num1}\mathrm{Mat}_i(S),\ \sum_{j=1}^{num2}\mathrm{Attr}_j(S),\ R,\ S,\ T,\ \mathrm{Step},\ \mathrm{Rela}(S),\ \mathrm{Order}(S)) \qquad (1)$$

Equation (1) includes this unit’s raw materials, attributes, transaction time T in its business flow, location R, business process Step, relative objects Rela(S) and business bills Order(S).

According to the tracing method above, manufacturer is responsible to maintain raw materials and product attributes in its database. EPCIS event database is responsible to provide the logistics movement information about product manufacturing and circulating. RFID application system is provided the service to query RFID tags in both real-time querying pattern and publish/subscribe pattern. Users can use fixed RFID terminal, web site and NFC mobile with RFID reader to read the RFID tag on the product. Users can also use web service and SMS to visit the product tracing platform and achieve products’ logistic tracing information.

V. CONCLUSION

Because of some limitations of enterprise information systems and the characteristics of RFID, real –time integrating of RFID data with enterprise information systems is a challenge task. The integration of RFID is not only data integration but also business integration. Many RFID middleware are designed for RFID device management or RFID data collection, but not for business process integration. RFID middleware products of companies, such as Oracle, BEA etc, are powerful but not scalable since these products are dependent on these companies’ technology platform.

In this paper, some researches on RFID middleware are discussed. A RFID integration middleware framework named Rdspor is introduced. This middleware focuses on business process integration between RFID and enterprise information systems. The RFID integration middleware named Rdspor is consisted of five modules which are devices monitoring, data processing, business process control, database management and business documents exchange. The main mechanisms of these modules are discussed in detail.

The semantic definition of the complex RFID event and RFID event classification are introduced. A real-time scheduling strategy of RFID event which is based on the buffer and priority queue, can handle the complexity of concurrent RFID events. RFID application system controls the business process through RFID events. Users identify RFID tag to query product tracing information in different modes.

ACKNOWLEDGMENT

This paper is based on a project “Research of Information Integration & Management Technology Based on RFID” which is supported by the National High-Tech. R&D Program for CIMS, China, No. 2006AA04A118 and No. 2006AA04A113. The research work is cooperated and supported by Ufida Software Company. It also thanks three masters, Zhai xinyue, Jin zuxu and Chen chen, who research data management of RFID event and data stream.

REFERENCES

[1] Xiaoyong Su, Chi-Cheng Chu, B.S. Prabhu, Rajit Gadh. “Creating a RFID Data Integration Framework for Enterprise Information System”. International Journal of Internet Protocol Technology, 2009.
[2] WANG Jun-yu; MIN Hao, “R & D of RFID system for supply chain management”; Computer Engineering and Applications, 2007, 43(13): 22-25.
[3] Zhang Min, Li Wenfeng, et al. “A RFID-based Material Tracking Information System”, Proceedings of the IEEE International Conference on Automation and Logistics, Aug. 18-21, 2007, China.
[4] SANJAY SARMA. “Integrating RFID”. http://www.acmqueue.com
[5] Christian Floerkemeier, Matthias Lampe. “RFID middleware design-addressing application requirements and RFID constraints”. SOc-EUSAI conference, 2005.
[6] Christian F., Christof R., Matthias L., “RFID Application Development With the Accada Middleware Platform”, IEEE Systems Journal, Vol, 1, No.2, Dec. 2007, 82-94.
[7] Zhang Jiehao. “Research and development of RFID device integration for middleware”. Shanghai Jiaotong University master's thesis, 2007.
[8] Yan Guoqing. “Research on RFID adapter of multiple protocols”. University of Electronic Science and Technology of China master's thesis, 2006.
[9] Zhao Li. “Design and Implementation of Event Management System in RFID Middleware”. Huazhong University of Science and Techonlogy master's thesis, 2006.
[10] Xu Qiang. “Design and Implementation of real-time event management of RFID Middleware”. Huazhong University of Science and Techonlogy master's thesis, 2007.
[11] Yang Xiaofeng. “Research on the key Technology of RFID Middleware”. Jilin University master's thesis, 2009.

Li Min-bo is an Associate Professor in the Software Scholl of Fudan University, China. He is also employed part-time as a research fellow of UFIDA software institute since 2006. He received his Ph.D. from the University of Tsinghua University in January, 2001. He did a postdoctoral research work at National University of Singapore from 2001 to 2002. He worked at central research institute of Kingdee Software Company from 2003 to 2004. Currently, he is an associate professor of Software Scholl of Fudan University from 2006.His current research interests are RFID application research and management information system.

Li Hua received his B.S. degree in Software Engineering from Shanghai Jiaotong University, Shanghai, China, in 2008, and is currently a master candidate in the department of Software Scholl at Fudan University, China. His research interests data management of RFID event and data stream, integration technologies between RFID and enterprise information systems, and RFID middleware.