A Performance Evaluation Tool for EPCIS

TaiHyun Ahn, Gihong Kim, Bonghee Hong
Department of Computer Engineering
Pusan National University
Busan 609-735, Republic of Korea
{anta77, buglist, bhhong}@pusan.ac.kr

Joonho Kwon
The Institute of Logistics Information Technology
Pusan National University
Busan 609-735, Republic of Korea
jhkwon@pusan.ac.kr

***Abstract*— EPCglobal has provided a standard layered architecture built around the EPC (Electronic Product Code) for the global RFID data sharing. Among these layers, EPCIS (EPC Information Service) plays an important role for tracing and tracking individual items. However, there is no well-known solution for testing EPCIS. In this paper, we propose a performance evaluation tool for EPCIS. This tool provides fast data insertion method, rapid test data generation method, and performance evaluation method. Finally, we performed experiments to analyze our tools' test data generation speed. Through these functions, our suggested tool can generate a large number of EPCIS event data rapidly, almost 40 times faster than existing tools.**

Keywords - RFID; Middleware; Test Data Generation; EPCIS

I. INTRODUCTION

Nowadays, since the price of RFID (Radio Frequency Identification) tag drops and the size of RFID become smaller, RFID technology is becoming essential in many areas such as pharmaceutical and supply chain management [1]. In the field of supply chain management, RFID technology, replacement technology of barcode, enables us to track individual items. Therefore, an RFID system should store an enormous amount of data.

EPCglobal [2] has suggested a standard architecture for EPC (Electronic Product Code) to support the use of RFID. The standard architecture consists of several layers such as ALE (Application Level Event), Capturing Application, EPCIS (EPC Information Service), ONS (Object Name Service) and EPCDS (EPC Discovery Service) [3]. Each layer communicates with each other using a Web Service (XML Web Service).

ALE is filtering and grouping a number of raw tag data from several readers. A Capturing Application changes raw tag data to event data and sends it to the EPCIS. An EPCIS stores entire event data that contains of raw tag information and business information.

EPCIS needs high performance to store and process a large volume of data. However, there is no well-known solution for evaluating EPCIS. And deployment real system to the real application for a test is waste of time and man power. To solve this problem, in this paper, we propose a method to generate test data for evaluating the performance of EPCIS.

The remainder of this paper is organized as follows. Section 2 explains the related work of our paper and introduces previous work on RFID middleware testing. Section 3 discusses the motivation of our approach and Section 4 presents a detailed description of our performance evaluation tool for EPCIS. In Section 5, we analyze our experimental result. We conclude our work in Section 6.

II. RELATED WORK

A. EPCIS

EPCIS is one of the standard architectures proposed by the EPCglobal [3]. Its main function is storing and searching event data. Event data consists of raw tag data and business information. The EPCIS Standard V.1.0.1 [4] proposed the basic architecture.

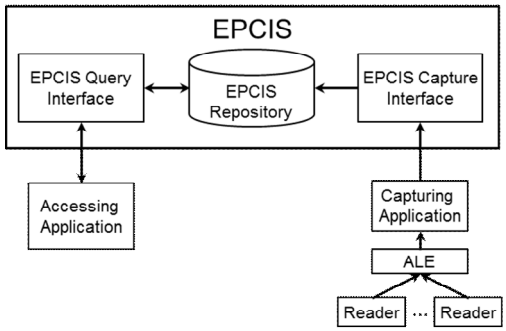

Figure 1. Simple EPCIS architecture

Figure 1 illustrates a simplified EPCIS architecture [5]. The Capturing Application receives raw EPC data from the Application Level Event and adds business information to make event data. After that, a Capturing Application sends event data to the EPCIS through the EPCIS Capture Interface. The EPC Repository changes form to store relational database such as Oracle and MySQL. When a user query from the Accessing Application is sent to the EPCIS Query Interface, the EPCIS Repository analyzes the query and changes it to the SQL and searches the data. After searching, the EPCIS Query Interface returns the result of the user query.

In the EPCIS standard data model, there are four event types: ObjectEvent, AggregationEvent, TransactionEvent and QuantityEvent [4]. Each event consists of various fields like event time, EPC list, action, bizlocation, and so on.

 ISBN: 978-1-61208-144-1

An ObjectEvent captures information about an event pertaining to one or more physical objects identified by EPCs. An AggregationEvent describes events that apply to objects that have been physically aggregated to one another. A QuantityEvent captures an event that takes place with respect to a specified quantity of an object class. A TransactionEvent describes the association or disassociation of physical objects to one or more business transactions.

B. Existing RFID Middleware Test Tool

Several papers [6, 7, 8] suggested test method for RFID middleware. These papers suggested that create virtual test data using virtual reader technique [6], performance test for ALE using several virtual readers and virtual accessing application [7], and ALE scalability verification to use several virtual readers and virtual path technique [8]. Figure 2 shows a simple idea of virtual reader and virtual tag. This method's main idea is to virtualize the real reader and to tag and make a real time test data for ALE.

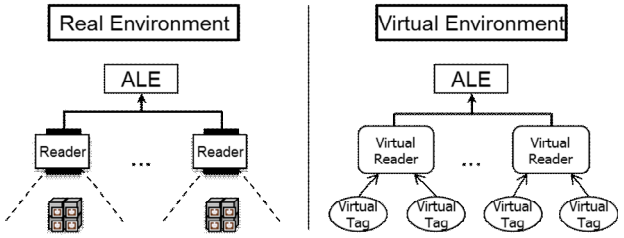

Figure 2. Example of Virtual reader

But these methods are focused on ALE. Therefore, these methods have problems for testing EPCIS. So in this paper we propose the performance evaluation tool for EPCIS.

III. PROBLEM DEFINITION

Since EPCIS handles a large amount of data, a high work load is expected in EPCIS. Figure 3 shows an example of the event occurs in a production factory. In producing place every item will be attaching a tag. Thus every item generates a one event. After producing, in packing part generates for group of items to one event. After packing, storing and transporting place also generate the same number of events.

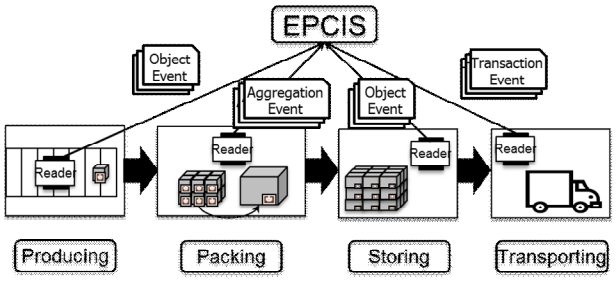

Figure 3. Example of real event generation

In this example, if a factory producing 200,000 items for a day and packed in 10 items, then the number of generated events in one month is shown as Table I.

TABLE I. EXAMPLE OF EVENT GENERATION

Business Process	Event Type	Event Number
Producing	Object Event	6,000,000
Packing	Aggregation Event	600,000
Storing	Object Event	600,000
Transporting	Transaction	600,000
Total		7,800,000

The first problem is the speed of data insertion. If test data is generated from ALE, that data must pass the Capturing Application. Since the Capturing Application adds business information to raw tag data to make EPCIS event data. But, as shown in Figure 4, each layer communicates through the web service using XML file in RFID middleware. But, marshaling operation, object conversion to XML file to be sent by the web service, and de-marshaling operation, XML file convert to object to be used in the program, are very slow operations [9]. Therefore, if a large volume of test data, such as Table I, generated from ALE will take long time to arrive at the EPCIS. And also, existing methods could generate noise data or redundant data cause of network communications.

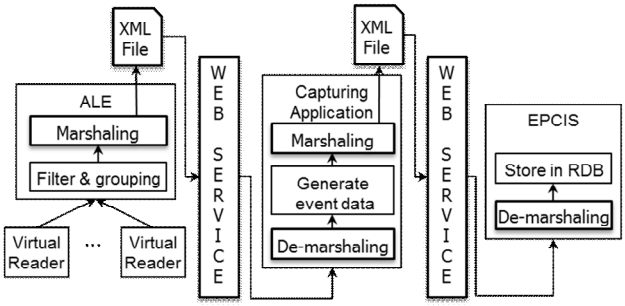

Figure 4. Example of web service communication

The second problem is the data generation latency. ALE processes only real time data. Therefore, the real environment's scenario, like Figure 3, has several seconds waiting time between product event and packing event or several hours waiting time between storing event and transporting event. Then, these testing tools have to wait same time for generating next event data since existing test tool only make a real time event for ALE. And, if the test data generating tool ignores the scenario and randomly generates event data, then meaningless data will be generated. Since that test data did not reflect the characteristics of the target environment.

To solve these problems this paper introduces a performance evaluation tool for EPCIS. This tool provides virtualization the passage of time for a high speed test data generation, high speed data insertion technique, and performance measurement function.

IV. DESIGN AND IMPLEMENTATION OF TESTING TOOL FOR EPCIS

This section proposes an architecture of the performance evaluation tool for EPCIS. Furthermore, we explain this tool's functions, parameters, and implementation.

 ISBN: 978-1-61208-144-1

A. Architecture

Figure 5 depicts the architecture of the performance evaluation tool for EPCIS. It consists of the configuration parser, event generator, tag generator, time controller, result generator, database connector, file connector, and EPCIS access controller module.

Configuration parser module analyzes user's input data such as tag information, event information, time information, and output information. Time controller module uses tag generator, which is instead of ALE's function in EPCglobal standard architecture, and event generator module, which is instead of capturing application, to generate business event data and tag data. Time controller module also gives time data to result generator module. Result generator module gathers the separated business, tag, and time data and generates event data for EPCIS performance evaluation.

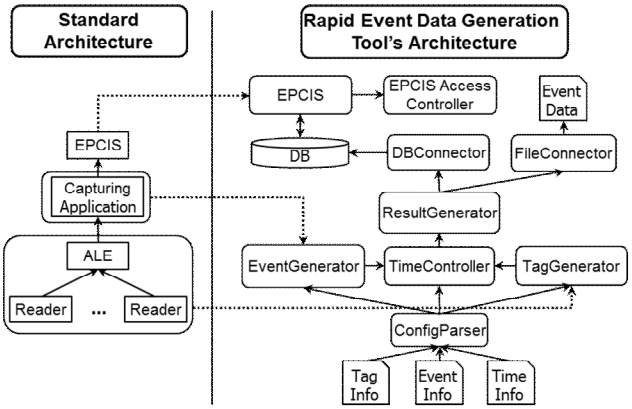

Figure 5. Architecture of the test data generation tool

Depends on users setting, generated EPCIS event data will insert database directly or store in the file using a database connector and file connector module. EPCIS accesses the controller module send user defined query to EPCIS. And if the EPCIS controller module got a reply from EPCIS then store query response time and number of stored event data to the database.

B. Function

In this section, we will explain each module's functions more detail.

1) Virtualizing passage of the time

Time controller module plays virtualizing passage of the real time function.

Existing RFID testing tools [6, 7, 8] are focused at ALE which uses only real-time data. Therefore, these testing tools are not considered scenario's waiting time of target environment. But EPCIS handles both real-time data and historical data. Therefore, test data generation tool for EPCIS has to control passage of the time to reflect test environment's scenario.

2) Tag, event data generating function

Tag generator and event generator module provide generating tag and business event data function. And result generator plays join of these two generated data function to make EPCIS event data.

EPCIS event data include both business information and raw tag data. According to this reason, unlike existing test tool, the performance evaluation tool for EPCIS has not only tag data generation function but also business event data generation function, instead of ALE and capturing application.

3) Fast Evnet data insertion techniques

Database connector and file connector module make output of our tool's result.

Insertion method of event data to EPCIS is web service communication using XML file, which is suggested by EPCglobal Standard. But web service communication's performance is too low to create a large number of event data [9]. So our test data generation tool provides direct insertion method to relational database for fast insert of event data for EPCIS. And also we provide file output method too.

4) Performance Measurement function

EPCIS access controller module plays send a user defined query to EPCIS query interface and store response time of query and number of stored event in EPCIS.

After creating the EPCIS test event data, our tool provides performance measurement method for EPCIS. EPCIS queries will define by a user using our tool's interface cause of the performance test of EPCIS is highly depended on target environment. After query definition EPCIS access controller sends generated query to EPCIS query interface. And, if the EPCIS access controller receives a result of a query sent from EPCIS, then store response time of query result and number of event, which is stored in EPCIS.

C. Parameters

Test data generation tool have to know about a real installation environment to generate meaningful test event data. Since the user should input real environment information parameters for test data generation. Parameter can be divided into three XML files.

1) Tag Information

The first parameter is tag information. Tag information contains a code type of EPC, company, item, serial number, which is the use at real environment. And also tag information includes a number of tags, generation cycle. Table II shows more detail information of tag information parameters.

TABLE II. PARAMETERS FOR TAG INFORMATION

Parameter Name	Description
Company	Code type and companyprefix
item	Item prefix
serial	Serial prefix
Max	Number of tag will generate
Delay	Delay for generation cycle
Tag	Representative one tag
TagList	Representative list of tag

Configuration parser module read this information and sends to tag generator then the tag generator makes raw tag data using this information.

 ISBN: 978-1-61208-144-1

2) Event Information

The second parameter is event information. Event data will be completely different depend on deploying environment's features. So setting of event data is a necessity for reflecting the characteristics of the target environment's scenario. Event information parameter contains a type of event, type of action, business step, read point, business location, and the waiting time for the next event. Table III explains more detail of event information parameters.

TABLE III. PARAMETERS FOR EVENT INFORMATION

Parameter Name	Description
Type	Set event type
action	Event relates to the lifecycle of the entity being described[4]
bizStep	A vocabulary whose elements denote steps in business processes[4]
readPoint	Identify the most specific place at which an EPCIS event took place[4]
bizLocation	Designate the specific place where an object is assumed to be following an EPCIS event until it is reported to be at a different Business Location by a subsequent EPCIS event[4]
stayTime	Staytime for next event
Event	Representative one tag
EventList	Representative list of event

Configuration parser module reads event information and sends to event generator module then the event generator makes business event data using this information. Figure 6 shows an example of event parameters.

```
<?xml version="1.0"?>
- <EventConfig>
  - <EventList>
    - <Event>
        <Type>ObjectEvent</Type>
        <action>Add</action>
        <bizStep>Product</bizStep>
        <readPoint>10</readPoint>
        <bizLocation>Conveyorbelt</bizLocation>
        <stayTime>2</stayTime>
      </Event>
    - <Event>
        <Type>TransactionEvent</Type>
        <action>Remove</action>
        <bizStep>Transport</bizStep>
        <readPoint>5</readPoint>
        <bizLocation>Docks</bizLocation>
        <stayTime>3</stayTime>
      </Event>
  </EventList>
</EventConfig>
```

Figure 6. Example of event parameter

3) Time information and output information

The third part is starting time and output information. The test data generation tool will emulate passage of time, since the start time setting is required. And test data generation tool provides two parameters for store method. One is directly inserting method to the relational database, and another one is the file writing method. Table IV shows more detail information of time and output information parameters.

TABLE IV. PARAMETERS FOR TIME AND OUTPUT

Parameter Name	Description
Year	Year of start time
Month	Month of start time
Day	Day of start time
Minute	Minute of start time
Second	Second of start time
Unit time	Program unit time for one second
File out	Value for file out
DB insert	Value for database insert

Configuration parser module reads time and output information and sends to time controller then the time controller makes time data. After that result generator gathers the tag data, business event data and time data from the tag generator, event generator and time controller thereafter result generator generates event data.

Data generation processing is finished then the database connector, and file connector module will make an output. If a user sets the Database connector module to use then this module insert directly to the database. If a user set file connector module to use then this module write a file for EPCIS event data.

D. Implementation of test data generation tool

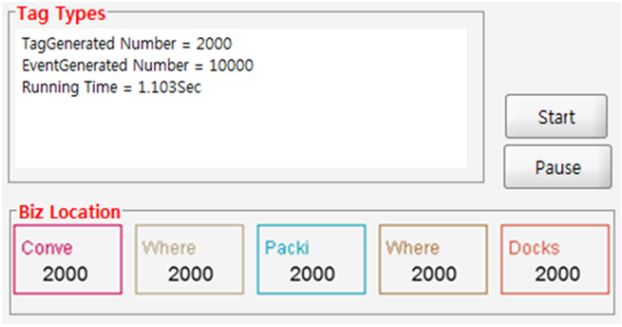

Figure 7. Test data generation tool

The performance evaluation tool for EPCIS is separated two parts. The first part is an event data generation part. This part shows the process of generating event data. If user click start button then analyzes a user input and draw biz location. After that tool starts generating event data directly and show progress of generation. Figure 7 shows implemented generation part.

 ISBN: 978-1-61208-144-1

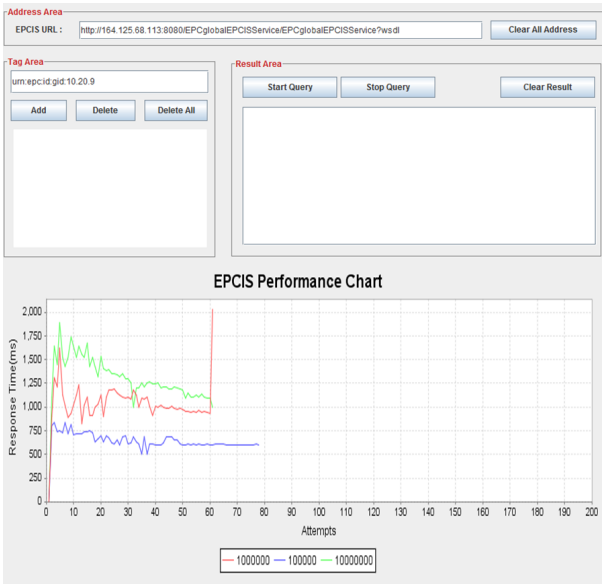

Figure 8. Performance Evaluation tool

The second part is performance test part. Figure 8 shows performance evaluation tool. First, user sets address of EPCIS and tag information for query. After that click start query button then evaluation tool start sending query to EPCIS. If the EPCIS reply query response then recording number of stored data and query response time to the database. Performance evaluation function provides not only point query, which search just one value, but also range query. Measured information shows a graph directly to the user. This function provides to users can easily test the performance of EPCIS.

V. PERFORMANCE OF TEST DATA GENERATION

We evaluated the effectiveness of our tool. For this regard, we compared our tool with an ALE test tool [8] by checking the time of event test data generation.

Since the ALE tool cannot generate EPCIS event data, we added the Capturing Application to transform tag data into event data. Figure 9 shows the architecture for test environments.

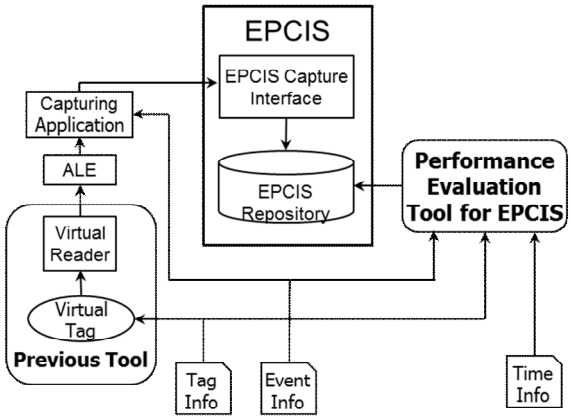

Figure 9. Test System Environment

All experiments were performed on an Intel Core 2 Quad Q6600 machine with 3.25 GB memory running Windows XP. We used an LIT-EPCIS which is the EPCglobal-certified software for EPCIS and ORACLE11g for EPCIS repository.

The parameter setting consists of one business location, one reader and one type of tag.

We have generated test event data from 1,000 to 500,000. We conducted this experiments 10 times and averaged experimental results.

Figure 10 shows the creating time for event data. Our tool shows a better performance than ALE test tool: it is 40 times faster than ALE test tool.

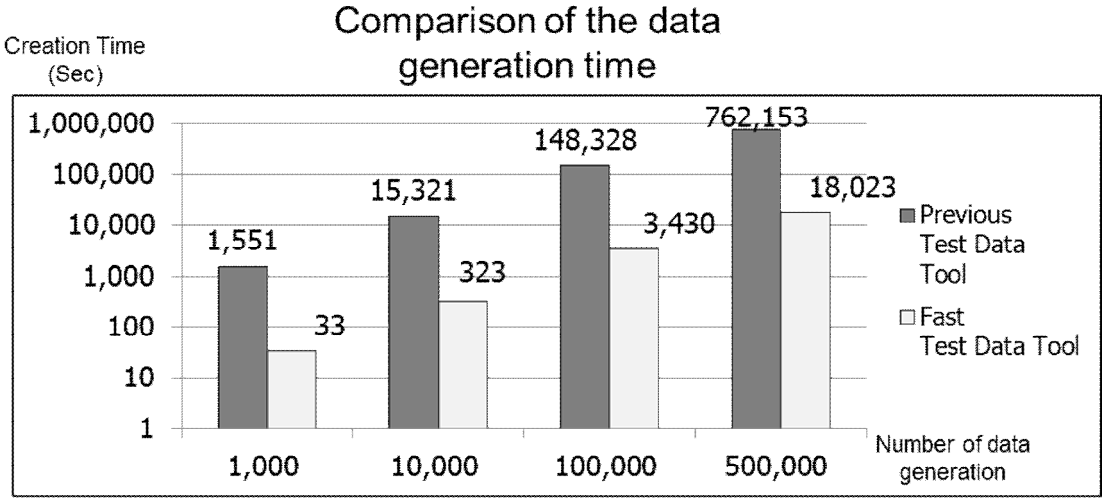

Figure 10. Result of Testing

 ISBN: 978-1-61208-144-1

VI. CONCLUSIONS AND FUTURE RESEARCH

EPCIS handles the large amounts of data since it can cause a performance problem. But there is no tool for testing EPCIS. Thus we propose an EPCIS performance evaluation tool. Its main functions are as follows: quick event data generation, fast insertion to EPCIS repository and performance evaluation. We have conducted an experiment to see the effectiveness of our tool. In our future work, we plan to consider the problem of optimizing RFID tags deployments and EPCIS based on our performance evaluation tool.

ACKNOWLEDGMENT

This work was supported by the Korean Ministry of Education, Science and Technology Grant" (The Regional Core Research Program/Institute of Logistics Information Technology)

REFERENCES

[1] Chun Hee Lee and Chin Wan Chung "Efficient storage scheme and query processing for supply chain management using RFID" Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, pp. 291-302.

[2] EPCglobal Inc., http://www.gs1.org/epcglobal, 05.29.2011

[3] EPCglobal Inc., "The EPCglobal Architecture Framework EPCglobal Final Version 1.4 Approved 15 December 2010"

[4] EPCglobal Inc., "EPC Information Services (EPCIS) Version 1.0.1 Specification"

[5] Himanshu Bhatt and Bill Glover, "RFID Essentials", O'Reilly, January 2006

[6] Haipeng Zhang, Wooseok Ryu, Bonghee Hong, and Chungkyu Park, "A Test Data Generation Tool for Testing RFID Middleware", in Proceedings The 40th International Conference on Computers & Industrial Engineering(CIE40), pp. 1-5.

[7] Jongyoung Lee and Naesoo Kim, "Performance test tool for RFID middleware: parameters, design, implementation, and features", Proceedings of the International Conference Advanced Communication Technology 1 (2006), pp. 149–152.

[8] Jekwan Park,Wooseok Ryu, Bonghee Hong, and Byeongsam Kim, "Design of toolkit of multiple virtual readers for scalability verification of RFID middleware", in Proceedings The Second International Conference on Emerging Databases (EDB 2010), pp. 56-59.

[9] Hazem M. El-Bakry and Nikos Mastorakis "Performance Evaluation of XML Web Services for Real-Time Applications", INTERNATIONAL JOURNAL OF COMMUNICATIONS Issue 2, Volume 3, 2009, pp. 25-33

 ISBN: 978-1-61208-144-1