Comparative Analysis on Adaptive Features for RFID Middleware

Siti Zaiton Mohd Hashim[1], Mardiyono[2*], Nurulhaini Anuar[3], Wan Mohd Nasir Wan Kadir[4]
Software Engineering Department, Faculty of Computer Science & Infomation Systems
University Technology Malaysia 81310 UTM Skudai, Johor, Malaysia
[*] *Electrical Engineering Department, Semarang State Polytechnic,*
Tembalang Semarang 50275 Jawa Tengah, Indonesia
email: [1]sitizaiton@utm.my [2], mardiyono@polines.ac.id, [3]nurulhaini@utm.my, [4]wnasir@utm.my

Abstract

Middleware is software that connects between hardware and application layer. Traditional middleware is limited in its ability to support adaptation while adaptive middleware enables modifying its behavior to conform to new situation. RFID applications grow widely and are used in many purposes such as supply chain management and ubiquitous computing enabled by pervasive, low cost sensing and identification. Implementing adaptive characteristic in RFID middleware will increase the capability of adaptation to specific environment such as different reader/tag, different application, and different platform. Adaptive middleware enables modifying the behavior of a distributed application after the application is developed in response to some changes in functional requirements or operating conditions. An extensive study has been carried out, and comparative analysis has been done on identifying the standard features that reflect the functionalities of RFID middleware and adaptive features that represent the non-functionalities of RFID middleware address to overcome the specific problems of application systems. This paper discusses the outcome of this study and adaptive middleware architecture for RFID applications is proposed that supports multi readers and multi applications.

I. INTRODUCTION

Radio Frequency Identification (RFID) is a technology to identify a tagged object using radio frequency. This technology comprises of reader, tag and host computer that captures, filters, saves and presents the information from the tag. RFID becomes a technology that changes bar code system. It has several advantages such as no require the tag to be in line-of-sight, multiple tags can be read simultaneously, can work in harsh environment [1], and available in short range (2 cm) until long range (15 m). That is why, RFID applications grow widely that were used in many purposes such as medicine [2], aeronautics [3], construction and maintenance of oil facilities [4], automotive [5], and retail [6].

Middleware is a software layer that connects and manages application component running on distribution host [7]. Middleware hides and abstracts many of the complex details of distributed programming from application developers [8][9]. Implementing middleware on RFID application has several benefits that refer to middleware functions [8], such as network communication, coordination, reliability, scalability, and heterogeneity. Thus RFID middleware has special purposes to collects a large volume of raw data from heterogeneous RFID environment, filters them and summarizes into meaningful information and delivers the information to application services [10].

Using traditional middleware in RFID application has a limitation in capabilities to support adaptation. Different reader, different platform and applications are several factors that influence the operation of RFID Middleware system. Therefore adaptive features that represent the non functionalities of RFID middleware have evolved from traditional middleware [11]. This paper identifies adaptive and standard features from RFID middleware literatures and selects several features to proposed RFID middleware architectures that support multi reader and multi application.

This paper is organized as follows. Section 2 discusses middleware technologies, section 3 reviews some commercial RFID middlewares, section 4 describes adaptive and standard features of RFID middleware from previous works. Comparative analysis and proposed architecture is discussed in section 5, and finally section 6 provides a conclusion and future work.

978-1-4244-1692-9/08/$25.00 ©2008 IEEE

II. MIDDLEWARE TECHNOLOGIES

This section describes the middleware technology that is focused on object-oriented middleware. It is the basis for the most research in adaptive middleware [11]. Three major object-oriented middlewares are Common Object Request Broker Architecture (CORBA) [12], Java Remote Method Invocation (RMI) [13] and Distributed Component Object Model (DCOM)[14].

CORBA is a distributed object framework proposed by the Object Management Group (OMG) [15]. CORBA has several components that support distributed object oriented computing across heterogeneous hardware device. The CORBA's components comprise of Object request Broker (ORB), Interface Definition Language (IDL), Dynamic Invocation Interface (DII), interface repository, dynamic skeleton interface (DSI), implementation repository, object adapter, ORB interface, general inter-ORB protocol (GIOP), and internet inter-ORB protocol (IIOP).

Java RMI [13] was proposed by JavaSoft to support the development of distributed Java-based application. RMI is built from three abstraction layers namely stub and skeleton layer, remote reference layer and transport layer. RMI uses *Java Remote Method Protocol* (JRMP) to handle a communication mechanism between client and server program in the network.

DCOM [14] was proposed by Microsoft as a distributed extension to the Component Object Model (COM) [16]. It is implemented only in windows platform. The DCOM architecture has several components such as the service control manager (SCM), object proxy and stub like stub and skeleton in CORBA, and COM library.

Comparing CORBA, RMI and DCOM technology is essential to know which technology is appropriate to be implemented in specific condition. Table 1 describes the comparison [17].

From table 1, it shows three technologies that have been discussed above provide similar mechanism for transparently accessing remote distribution object. DCOM is excellent choice to implement in organization has adopted Microsoft technology. RMI is the simplest and fastest way to implement distributed object architecture in Java application but it is not a good choice for heterogeneous application. CORBA supports diverse languages can be used as long as the IDL can be mapped to that language and operating system interoperability. CORBA is the logical choice for building enterprise-wide, open architecture, and distributed object application [18].

TABLE 1: COMPARING CORBA, RMI AND DCOM

Comparing Items	Middleware Technology		
	CORBA	RMI	DCOM
Diverse languages	√	X (java)	√
Interoperability support	√	√	X (windows)
Client/Server interface	√	√	√
Remote protocol	√ IIOP	√ JRMP	√ ORPC
Object Identification	√	√	√
Object Location and activation	√	√	√
Inheritance support	√	√	√
On demand activation	√	√	√
Exception Handle	√	√	√
Garbage collection	X	√	√

√ = support and X = not support

III. COMMERCIAL RFID MIDDLEWARE

Identifying the implementation of middleware technology in commercial RFID middleware is crucial to determine the correct features and components most relevant with the applications. The commercial ones that will be discussed are Sun RFID, Sybase RFIDAnywhere, Connec Terra/BEA, and GlobeRanger. Table 2 describes their comparison [19].

Table 2 indicates the features available in each selected middleware. It shows that features like filtering, data aggregation, manageable, XML message, monitoring and reader adapter exist in all products. Sun has unique features in supporting of multi databases that the component called RFID Information Server has been qualified with Oracle 8i, Oracle 9i, Oracle 10g, and PostgreSQL 8.0.3. ConnecTerra/BEA provides Application Level Event-Application Programming Interface (ALE-API) that supports to multi applications in different format such as web service, .Net application, Java application, and EAI framework. Sybase provide intelligent features that supports pluggable architecture, intelligent network of sensors, broad hardware support with abstraction layer, multi protocol RFID tag support and simulation capabilities to assist development and planning. Visual emulation is a special feature from Globe Ranger that provides a deployment emulation environment for testing and integration.

TABLE 2: COMPARISON OF COMMERCIAL RFID MIDDLEWARE FEATURES

Faetures	Commercial RFID Middleware			
	Sun	Sybase	Connec Terra/BEA	Globe Ranger
Filtering	√	√	√	√
Data Aggregation	√	√	√	√
Manageable	√	√	√	√
Monitoring	√	√	√	√
Alert system	√	√	X	√
XML message	√	√	√	√
Multi Database	√	X	X	X
Mobile Device	X	√	√	√
Reader Adapter	√	√	√	√
Multi Application	X	√	√	√
Visual Emulation	X	X	X	√

√ = support and X = not support

IV. RELATED WORKS ON RFID MIDDLEWARE

Many researches have proposed RFID middleware architectures such as RFID middleware Design[20], REMS and RBTS [21], Architecting RFID Middleware[22], and Distributed Large Scale System [23]. Table 3 shows the comparison of some features from previous researches.

TABLE 3: COMPARISON OF THE FOUR PROPOSED ARCHITECTURES

Features	Previous Researches			
	[20]	[21]	[22]	[23]
Messaging	√	√	√	√
Filtering	√	√	X	√
Data Aggregation	√	√	X	√
Manageable	√	√	X	√
Monitoring	X	√	X	√
Reader Adapter	X	√	√	√
Printer Agent	X	X	√	X
Multi Data Base	X	X	√	X

√ = support and X = not support

Table 3 shows that four proposed architectures have messaging features because it is the main function of middleware to deliver the message to intended destination in distributed host. Almost all architectures have features like filtering, data aggregation, manageable, monitoring, and reader adapter. Only Architecting RFID Middleware [22] provides printer agent to communicate with printer and database agent (multi data bases) to connect and manipulate with several relational data base management system (DBMS) such as Derby, Oracle, DB2, MySQL, Sequel Server, and Postgres.

V. COMPARATIVE ANALYSIS OF RFID MIDDLEWARE

Based on the discussion in section 4 and 5, it shows various features exist in RFID middleware; commercial version and proposed in the previous researches. The features can be grouped into 2 group's namely standard and adaptive feature. Standard feature is a functionality of RFID middleware. Otherwise adaptive feature refers to non functionality of RFID middleware and it is provided to support adaptation to new or different situation such as different reader, different application, different platform and network failure.

All of commercial ones support filtering, data, aggregation, manageable, monitoring, and XML message. Filtering can reduce the amount of information flowing from RFID interface/adapter by preventing duplicate data. Data aggregation stores incoming RFID data in repository. Manageable is a feature that makes the user easy to configure and manage the device or system. Monitoring provides the service that continuously monitors the resources [23]. Finally XML message is the feature to provide web service in XML format that can be sent over HTTP. Four proposed architectures from previous researches have messaging feature that deliver the message to intended application server.

The features that are grouped in standard features are filtering, data aggregation, monitoring, messaging, XML message, and supporting mobile device. Adaptive features that can conform to different situation are reader adapter (support multi reader), manageable (support self configuration, decentralized management, and adaptability to changes in RFID network condition [23]), multi data bases and multi applications.

Base on comparison study, this research proposes adaptive RFID middleware architecture that is built from selected features. This research selects several standard and adaptive features to support multi readers and multi applications. The selected standard features are messaging, filtering, monitoring, XML message, and data aggregation (repository) and selected adaptive

features are reader adapter and support to multi applications (message adapter). These selected features support functionality of RFID middleware included captures RFID tags, filter, aggregates, and delivers RFID data to intended server. Furthermore, this research selects reader adapter to support multi readers and message adapter to support multi applications and conform to new application. Figure 1 describes proposed architecture of adaptive RFID Middleware.

This architecture comprises of eight components in order to support multi reader and multi application which illustrated in figure 1.

Reader interface: provides events to read RFID tag and presents the information of tag ID by converting byte data to hexadecimal format..

Reader adapter: support multi reader with different specification or different manufacturer and manage or control all of the readers.

Monitoring: it monitors the aliveness of each managed readers and notify the events whenever some abnormal situations happen, such as reader disconnection.[21]

Filtering: prevent duplicate or redundant data and add metadata to the information going to messaging.

Messaging: provides component to deliver information to intended application server (app 1 to app n) and format the message to Extensible Markup Language (XML).

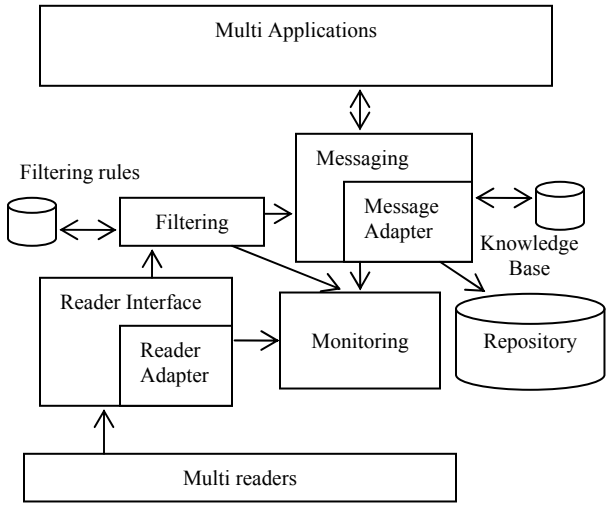

Figure 1. Proposed Adaptive RFID Middleware

Message adapter: route the information to intended server by retrieving tag information from knowledge base and conform to new application.

Filtering rules: save the rules for filtering RFID tag and metadata information.

Knowledge base: contains tag ID and owner application (app 1 to app n). this information is used for delivering message to intended application server.

VI. CONCLUSION AND FUTURE WORK

This paper presents a comparative analysis of adaptive features in RFID middleware from two perspective commercial and research aspects. The identified features can be grouped into two elements namely standard features that refer to the functionality of RFID middleware and second the adaptive features that represent non functionality of RFID middleware.

Future work will focus on the development of the adaptive RFID middleware based on the proposed architecture. The proposed middleware will be validated using several case studies developed.

REFERENCES

[1] S. Mike, "Radio Frequency Identification (RFID) Technology and its Applications in the Commercial Construction Industry", *Technical Report*, University of Kentucky Civil Engineering Department, April 24, 2003

[2] URL1, "Using RFID Technologies to Reduce Blood Transfusion Errors", http://www.cisco.com/global/IT/local_offices/case_history/rfid_in_blood_transfusions_finel.pdf

[3] URL2, "Boeing Tags Shipment to the DOD", http://www.rfidjournal.com/article/articleview/1587/1/1.

[4] DTI Basic Technologies, "RFID tagging for the oil industri- a brief introduction", *Petroleum Review*, 2004 http://www.basictechnologies.gov.uk/site/projects/SmartTagArticle.pdf.

[5] E. Fleisch, M. Strassner, "The Promise of Auto_ID in the Automotive Industry", *Auto_ID Center*, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2003.

[6] K. Albrecht, "Supermarket Cards: The Tip of the Retail Surveillance Iceberg", *Denver University Law Review* 79 (4) 534-554.<http://www.spychips.com/documents/Albrecht-Denver-Law.pdf>.

[7] J. Park, S. Kim, W. Yoo, S. Hong, "Designing Real Time and Fault-Tolerant Middleware for Automotive Software", *Proceeding of SICE-ICASE International Joint Conference*, 2006.

[8] W. Emmerich, "Software Engineering and Middleware: A Roadmap. In the Future of Software Engineering", *ACM Press*, 2000., Pages. 76-90

[9] R. Schantz and D. Schimidt, "Middleware for Distributed System – Evolving the Common Structure for Network-centric Applications", *In the Encyclopedia of Software Engineering*, John Wiley & Sons, December 2001.

[10] G. O. Oh; D. Y. Kim; S. I. Kim; S. Y. Rhew, "A Quality Evaluation Technique of RFID Middleware in Ubiquitous Computing", *Hybrid Information Technology, 2006. ICHIT'06. Vol 2. International Conference on* , vol.2, no., pp.730-735, Nov. 2006

[11] S.M.Sadjadi, "A Survey of Adaptive Middleware", *Technical Report*, Computer Science and Engineering, Michigan State University, Sept. 2003.

[12] Object Management Group, "The Common Object Request Broker: Architecture and Specification Revision 2.2", 492 Old Connecticut Path, Framingham, MA 01701, USA, Feb. 1998.

[13] Java Soft, "Java Remote Method Invocation Speci¯cation, revision 1.5", JDK 1.2 edition, Oct. 1998.
[14] Microsoft Corporation, "Microsoft COM Technologies-DCOM", 2000, http://www.microsoft.com/-com/dcom.asp.
[15] Object Management Group, "The Common Object Request Broker: Architecture and Specification, Revision 2.0", 492 Old Connecticut Path, Framingham, MA 01701, USA, July 1995.
[16] Microsolft Corporation, "COM: Delivering on the Promises of Component Technology", 2000. http://www.microsoft.com/com/default.asp
[17] H. Xiao, "CORBA, RMI and DCOM", http://www.cs.queensu.ca/home/xiao/DS/node9.html, 30th March 2005
[18] P. Agustin, "Comparing RMI, DCOM, & CORBA", http://ww.nolacom.com/publication/whitepapaers/comparing RMI DCOM CORBA.pdf
[19] B. Himanshu, G. Bill, *RFID Essential*, *O'Reilly*, United States of America ,January 2006
[20] F. Christian and L. Matthias, "RFID middleware design - addressing application requirements and RFID constraints", *Proceeding Joint SOC-EUSAI conference*, Grenoble, october 2005
[21] T. Cheong; Y. Kim; Y. Lee, "REMS and RBPTS: ALE-compliant RFID Middleware Software Platform", *The 8th International Conference Advanced Communication Technology ICACT 2006,* Volume 1, 20-22 Feb. 2006 Page(s):699 – 704
[22] J.E. Hoag and C. W. Thompson, "Architecting RFID Middleware", *IEEE Computer Society*, September - October 2006
[23] F. Bo, L.J. Tao, Z. Ping, G. J. Bo, D. Z. Hua, "Study of RFID Middleware for Distributed Large-scale Systems", *Proceeding of Information and Communication Technologies (ICTTA) 2006*, volume 2 page(s):2754-2759, 24-28 April 2006