Crop Production Recognize Frameworks using Mobile Enterprise Application

Haeng Kon Kim and Roger Y Lee

***Abstract*— In this paper we propose an Enterprise Application Framework using Mobile that uses CBD (Component-Based Development) as development methodology and Web Services for information interchange in the RFID-based distributed computing environment for crop production in agricultural area. We tried to analyze the requirements of mobile applications architecture and mobile enterprise application framework, some CBD attributes and Web Services for mobile crop production management, and EPC global network which is RFID-based network environment. We have demonstrated implementation of an example of RFID Application using AAF which is a light-version of EAF (Enterprise Application Framework). Also we will show how to construct the enterprise applications using the proposed Enterprise Application Framework by changing the contents of the Application Layer.**

***Keywords*— RFID, EPC global, CBD, Web Services, Middleware, Enterprise Application**

I. INTRODUCTION

Researches and developments for ubiquitous computing environment which is human centered future computing environment are widely preceded. One of them is the research and development of RFID (Radio Frequency Identification) technology [3, 4]. The EPC (Electronic Product Code) Network, which was developed by the Auto-ID Center and now managed by EPCglobal Inc., was designed and implemented to enable all objects in the world to be linked via the Internet [5].

The functions of the EPC Network using RFID tags are as follows: recognize, identify all objects, track, trace, monitor, trigger events and actions on those objects, and offer real-time view of assets and inventories throughout the global supply chain [6].

Web Services is an interface which describes a set of accessible network commands through standard XML messages. Also, Web Services can incorporate distributed components which are implemented by the existing different protocol, and can provide distributed computing over the Internet.

In this paper, to propose Enterprise Application Framework which is based on RFID and distributed computing, component environment, and use CBD (Component-Based Development) and Web Services where the distributed computing is available over the Internet.

The paper is constructed as follows: section 2 suggests related work, section 3 describes RFID-based Enterprise Application Framework using CBD and Web Services, section 4 demonstrates a prototype system which is implemented on the proposed Enterprise Application Framework using an RFID kit, and section 5 discusses the conclusion and the future study.

Heang-Kon Kim is with School of Information Technology, Catholic University of Daegu KyungSan, Daegu, 712-702, Korea (corresponding author to provide phone: 053-850-2743; fax: 053-850-2740;
e-mail: hangkon@ cu.ac.kr).

Roger Y Lee is with Department of Computer Science, Central Michigan University, Mt. pleasant, M.I. 48859, USA
email: lee1ry@cmich.edu

we will study RFID technology, integrate the distributed

II. RELATED WORKS ON SPATIAL INDICES

A. EPCglobal Network

The EPC Network, which was developed by the Auto-ID Center and now managed by EPCglobal Inc., was designed and implemented to enable all objects in the world to be linked via the Internet [5]. The architecture of EPCglobal Network is shown in Figure 1.

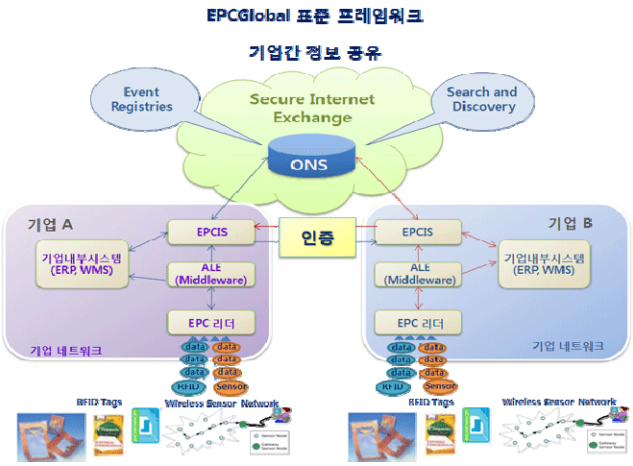

Fig. 1. The architecture of EPCglobal Network

[ubiquitous computing DB referenced]

EPC (Electronic Product Code) code is the numeric data transmitted by a tag which is the next generation of the UPC (Universal Product Code), that is bar code. Unlike the UPC, the EPC is designed to operate not only wirelessly, but to uniquely identify each individual object [6]. Typical EPC Network consists of three parts: Savant, EPCIS, and ONS.

SAVANT is middleware between RFID Reader(s) and application which passes requests from applications to reader(s) and receives unique tag identifiers and possibly other data from sensors, and passes that information to the applications [5]. Event management in Savants is main function such as filtering. Because readers may read data coming from multiple tags many times, it is inevitable to

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

eliminate redundant or unnecessary information. EPCIS (Electronic Product Code Information Service) communicates with databases and stores the tag data when it is read. ONS (Object Name Service) identifies the location of the server hosting the appropriate information needed by the application [5].

B. CBD (Component-Based Development)

Software components are binary units of independent production, acquisition, and deployment that interact to form a functioning system. Composite systems composed of software components are called *component software* [12]. Abstractions, such as procedures, classes, modules, or even entire applications could form components, as long as they are in a 'binary' form that remains composable. The benefit of component software is as follows: components are the way to go because all other engineering disciplines introduced components as they became mature. Component-Based Development claims to offer a radically new approach to the design, construction, implementation and evolution of software applications. Software applications are assembled from components from a variety of sources; the components themselves may be written in several different programming languages and run on several different platforms. CBD architecture is being used nowadays and the research on how to make it more efficient is the focus of this study. In this paper we tried to integrate the concept of CBD to develop a mobile enterprise application. [12].

C. Web Services

Web Services is a collection of technologies that accomplish future distributed computing environment. The reason why it is suitable for the distributed computing environment is due to the design objective of Web Services whose purpose is to establish interoperability among heterogeneous network environments. There are basic specifications in the Web Services such as SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), and UDDI (Universal Description, Discovery, and Integration) [7]. These basic specifications are based on Internet standard such as XML (eXtensible Markup Language), XML namespace, XML schema, and HTTP (Hypertext Transfer Protocol). SOAP is a protocol that explains message transmission method among heterogeneous network environments. WSDL is XML based language which describes the functions of Web Services. UDDI registers Web Services and provides the registry of Web Services with the users who are trying to find the services.

D. Enterprise Application Framework

An enterprise architecture framework defines how to use enterprise architecture. An architecture framework provides principles and practices for creating and using the architecture description of a system. To manage the scale and complexity of this system, an architectural framework provides tools and approaches that help architects abstract from the level of detail that builders work at to bring enterprise design tasks into focus and produce valuable architecture description documentation. Enterprise architecture regards the enterprise as a large and complex system or system of systems.

III. DESIGN OF ENTERPRISE APPLICATION FRAMEWORK

A. CBD and Web Services-based Enterprise Application Framework

EPCglobal Network is a distributed computing environment. The critical matter for the distributed computing environment is how to transmit data from one computer to another computer. Of course there are many ways to transmit data in this distributed computing environment, such as CORBA (Common Object Request Broker Architecture), RMI (Remote Method Invocation), DCOM (Distributed Component Object Model), and so on. There, however, are a few drawbacks in these technologies, respectively. DCOM is a technology that is based on COM (Component Object Model) model, which is binary standard that has shortcoming to distribute in the different environment except Microsoft Windows platform. And CORBA is relatively difficult to install the applications in the client and the sever model. Especially, it requires a client installation module for every client.

However, Web Services can make distributed application little more simple which was not feasible in the past. Because the architecture of Web Services inherits the characteristic of Web in the different fashion which the existing distributed programming methodology might take.

The reasons why the proposed Enterprise Application Framework adopts Web Services are as follows: Web Services inherits the existing Web standards, such as XML and HTTP. It utilizes platform neutral technologies. In addition, it has an assumption that any request for the service may be failed, thus Web Services should have the detecting mechanism.

B. Architecture of Enterprise Application Framework

Architecture of RFID-based Enterprise Application Framework using CBD (Component-Based Development) and Web Services is shown in Figure 2. Application Layer consists of domain dependent components in which domain specific modules such ERP (Enterprise Resource Planning), SCM(Supply Chain Management), or Warehouse System may be situated.

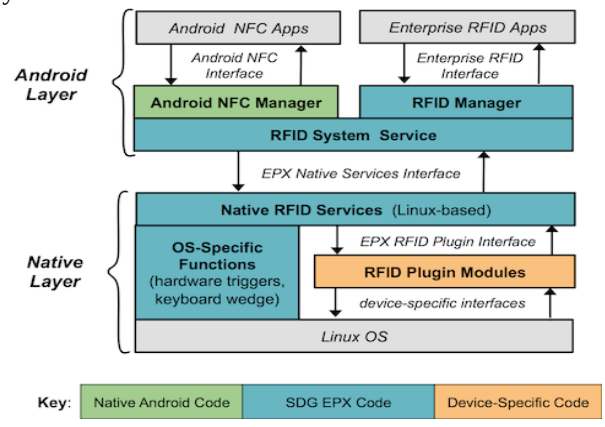

Fig. 2. Architecture of RFID-based Enterprise Application Framework

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Framework Layer consists of domain independent components in which domain neutral modules such as Security Services, Messaging Modules, Transaction Manager, and so on, may be situated. These modules need not to be modified even if the domain is changed. This is one advantage of CBD (Component-Based Development).

RFID Middleware Layer passes requests from applications to reader(s) and receives unique tag identifiers, and passes that information to the applications.

RFID readers consist of an antenna and control unit which monitors encoding and decoding, data check and save, tag and communication with host, and so on.

C. Application Layer

Application Layer consists of domain dependent components in which domain specific modules may be situated.

1. UDDI Access Module

UDDI is a collection of shared directory or protocol that we can register Web Services and can search for the registered service by the real time.

2. WSDL Module

WSDL is an interface language that informs outside users how to use functions offered by the Web Services. WSDL corresponds to IDL (Interface Definition Language) of CORBA.

3. Repository Module

Repository Module is a database that stores WSDL document whose contents are domain specific.

D. Framework Layer

Framework Layer consists of domain independent components in which domain neutral modules may be situated. These modules need not to be modified even if the domain is changed.

1. Authentication Module

Authentication module guarantees that only the users who are given rights can access to the application, and the module is divided by authentication and authorization. Authentication is a technique that transmits user's identity, and authorization is a technique that judge availability whether it comforts to the required access level. We will install an authentication server in the center only because there are difficulties in the key management if all servers try to authenticate clients directly in the distributed environment. Also we introduce the technique of single sign-on that allows a client to connect repeatedly to the several servers by the password authentication at one time.

2. Digital Signature Module

Digital Signature Module handles digital signature [10] which is used to secure the writer's identity by the electronic way, and has the functions of integrity and authentication at the same time. Also it can be used for the purpose of non-repudiation. This module can issue digital signature of XML document, and can provide the function of integrity with the document.

3. Encryption Module

The practice of hiding messages so that they cannot be read by anyone other than the intended recipient. There are two types of encrypting methods: symmetric and asymmetric. In the symmetric method, sender and recipient share a common key. In the asymmetric method, there are two keys – a public & a private key. The public-key, which may be known by anybody, and can be used to encrypt messages, and verify signatures a private-key, known only to the recipient, used to decrypt messages, and sign signatures. This is asymmetric because those who encrypt messages or verify signatures cannot decrypt messages or create signatures [8]. Our proposed Encryption Module supports both symmetric cipher and asymmetric cipher, and also provides the function of XML Encryption [9].

4. Reliable Messaging Module

Reliable Messaging Module securely guarantees reliable message between sender and recipient. Reliable message transmission is based on the session technology, which has ability to provide numbering with the messages in the same session, has acknowledgment function to prove the receipt of message, and has verification function for the specific message.

5. Transaction Module

The usual distributed transaction modules try to lock against transaction failure. Web Services, however, can not do that because the category and scale of transactions are so great [2]. The compensating transaction takes place whenever transaction failure is occurred.

6. Messaging Module

Messaging Module takes charge of creating SOAP messages which will be transmitted. If necessary, name space, header, body, encryption, digital signature, authentication information, and so on may be added.

7. Web Server Module

Web Server Module is in charge of transmitting SOAP messages. SOAP message may be applied to the SMTP protocol as well as HTTP protocol. However, SOAP message utilizes only HTTP since it is very difficult to control diverse protocols appropriately in the viewpoint of the specific protocols.

8. Logging Module

Logging Module stores information on the transaction failure, message transmission failure, system error, and so on.

IV. AN ALE APPLICATION SCENARIO

Figure 3 is a typical example of RFID Application using AAF which is a light-version of EAF (Enterprise Application Framework), suggested in this paper. AAF is constructed on the basis of ALE engine. EPCglobal network shows a new standard, called *Application Level Events (ALE)*, which is developed from the concept of a middleware, called *Savant*. The role of the ALE is to provide a means to process the event data which were collected by the RFID reader and deliver them to the higher-level applications [4, 11].

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

AAF is architecture of a framework whose purpose is to allow the user to develop ALE application with ease. After developing AAF, we are trying to extend it to EAF framework by replacing BP (Business Process) component of AAF by BEF (Business Enterprise Framework) component which is under development by another team of our research group.

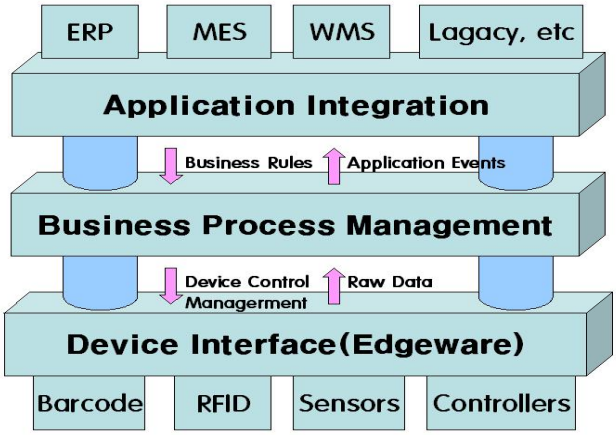

Fig. 3. A typical ALE Application Scenario of the application

Step 1: Client defines an interested EPC Event using ALE Client API, and produces ECSpec requesting for EPC event. Figure 4 shows an example of ECSpec defined in the Event Layer.

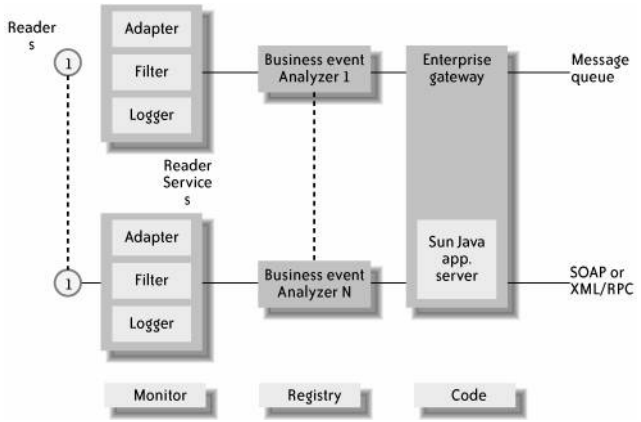

Fig. 4. An example of ECSpec

Step 2: AAF delivers ECSpec produced through Service Agent to ALE Manager via SOAP. Also it receives ECReport including EPC Event corresponding to ECSpec, and delivers the ECReort to EventHandler one by one. Figure 5 shows ECReport coming from ALE Manager.

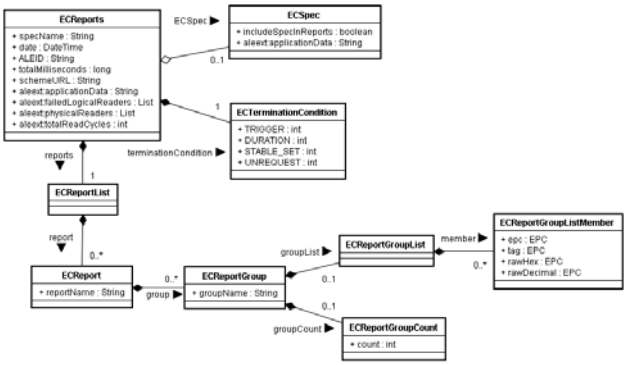

Fig. 5. ECReport corresponding to the ECSpec in step 1.

Step 3: EventHandler of Event Layer receives the delivered ECReport, creates a logical event corresponding to the pre-determined Logical Event Rule, and delivers it to the corresponding Business Rule. Figure 6 shows a logical event (Current) representing the present status.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <LogicalEvents date="2005-09-01T14:36:33.359+09:00" creationDate="2005-09-01T14:36:33.359+09:00" schemaVersion="1"
  xmlns="urn:rclit:aaf:xsd:1">
- <events xmlns="">
  - <event>
    - <current reportName="report1">
      - <group>
        - <groupList>
          - <member physicalLocation="127.654 37.551">
              <epc>urn:epc:tag:sgtin-64:4.011562.0557083.19206803</epc>
            </member>
          </groupList>
        </group>
      </current>
    </event>
  </events>
</LogicalEvents>
```

Fig. 6. A logical event (Current)

Figure 7 illustrates a logical event (Move) demonstrating the status of movement.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <LogicalEvents date="2005-09-01T14:36:33.359+09:00" creationDate="2005-09-01T14:36:33.359+09:00" schemaVersion="1"
  xmlns="urn:rclit:aaf:xsd:1">
- <events xmlns="">
  - <event>
    - <move>
        <epc>urn:epc:tag:sgtin-64:4.011562.0557083.19206803</epc>
      - <from>
          <logicalReader>door1_in</logicalReader>
          <date>2005-09-01T14:36:13.984+09:00</date>
        </from>
      - <to>
          <logicalReader>door1_out</logicalReader>
          <date>2005-09-01T14:36:25.234+09:00</date>
        </to>
      </move>
    </event>
  </events>
</LogicalEvents>
```

Fig. 7. A logical event (Move)

Step 4: Business Layer executes a business process corresponding to the delivered Logical Event. And this layer represents the logical event corresponding to Business Process, and can transfer this event to EPCIS or other legacy systems, too. In addition, this layer processes static information from EPCIS which is given by manufacturing company, and track & trace information from ONS which contains information on the supply chain.

Figure 8 shows a screen shot of a meaningful result of a logical event (Move) according to the business rule, and Figure 9 shows a screen shot of the storing result of a logical event (Current) according to the business rule in the EPCIS.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

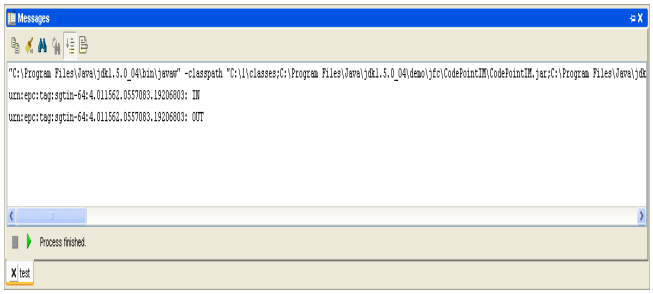

Fig. 8. A meaningful result of a logical event (Move)

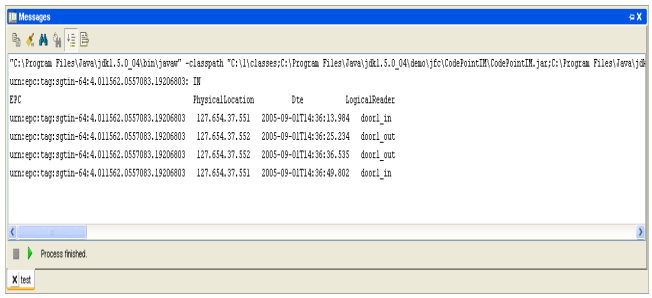

Fig. 9. The storing result of a logical event (Current)

V. CONCLUSION

In this paper, we have proposed Enterprise Application Framework which is based on EPCglobal and RFID technologies, and have utilized CBD (Component-Based Development) and Web Services to address the problem of the distributed computing over the Internet for drop production applications. Also we have demonstrated implementation of an example of RFID Application using AAF which is a light-version of EAF (Enterprise Application Framework). The characteristics of the Enterprise Application Framework are as follows: Application Layer may be modified when the domain is changed. Framework Layer, however, in the proposed system need not to be modified even if the domain is changed. Thus, we can reuse software which is fundamental technique in the software engineering.

In the future, we will design and implement the detailed system for the proposed model in this paper. Also we will investigate the business application framework in more detail to leverage linkage between enterprise application and middleware or EPC related information.

ACKNOWLEDGEMENTS

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the CITRC(Convergence Information Technology Research Center) support program(NIPA-2014-H0401-14-1008) supervised by the NIPA(National IT Industry Promotion Agency).

This research was also supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning(Grant number: NRF-2012K1A3A7A03057499).

REFERENCES

[1] Clark, S. et al., *"Auto-ID Savant Specification 1.0"*, Auto-ID Center, 2003.
[2] *"Dependency Spheres"*, http://www.Research.ibm.com/AEM/documents/dspheres_edoc2001.pdf.
[3] *"EPCglobal US"*, http://www.epcglobalus. org/.
[4] Harrison, M., *"EPC Information Service-Data Model and Queries"*, Auto-ID Center White Paper, 2003.
[5] Leong, K.S. et al., *"EPC Network Architecture"*, Auto-ID Lab, 2004.
[6] Meloan, S., *"Toward a Global "Internet of Things"* Technical Article of Sun Developer Network Community, 2003.
[7] Shin, M., *XML Web Services*, FREELEC Co., 2004.
[8] Stallings, W., *"Cryptography and Network Security, 3rd Ed"*, Prentice Hall, 2003.
[9] *"XML Encryption"*, http:// www.w3.org/ Encryption/2001 /.
[10] *"XML-Signature"*, http://www.w3.org/TR/ xmldsig-c Ore /.
[11] *EPCglobal*, The Application Level Events (ALE) Specification, Version 1.0 of February 8, 2005.
[12] Szyperski, C., *"Component Software"*, Addison-Wesley, 1998.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)