Implementing an Information Service System for Physical Object Distribution in Ubiquitous Network

Yuna Jung[1], Eenjun Hwang[2]

[1]Graduate School of Information and Communication, Ajou University, Suwon, 443-749 Korea

serazade@ajou.ac.kr

[2]Department of Electronics and Computer Engineering, Korea University, Seoul, 136-701 Korea

ehwang04@korea.ac.kr

Abstract

As computing devices are getting smarter and network technology is improved, ubiquitous network is being exploited widely. However, ubiquitous network would be more practical and applicable if huge amount of related information can be handled efficiently. In this paper, we address the importance of information service and propose service architecture to store and manage information in a ubiquitous application for physical object distribution. Especially, we focus on how to manage the distribution flow of objects and offer in-formation to users.

Key words: Ubiquitous, Information Service

1. Introduction

Recently, as computing and communication devices are getting smaller and more powerful, industry as well as our daily life is experiencing big changes in many aspects. In addition, emerging standards for data presentation and transfer, such as HTML, XML, and WAP, make information services even more available. Improvement of the international computing environment is leading the change of computing nature. That is, computing devices will be so pervasive and critical to our daily life [1]. Such small and smart devices effectively disappearing into the background, what we called as "Ubiquitous Computing," has been a motivating vision underlying many computer science researches in the past decade.

The key to the ubiquitous computing is the context awareness. Many difficulties in the existing computer systems have resulted from the fact that the system was not able to gather environmental information and forced users to input various data instead. One of the fundamental aspects of the ubiquitous computing is to deploy many computers and let the computer system as a whole recognize the environmental context, and offer the optimal service to the users. Here, context indicates various factors including positions and attributes of various objects even human, and other environ-mental factors such as temperature, humidity, illumination, date, and so on. RFID (Radio Frequency Identification) is expected to make it possible to recognize such factors.

RFID is a technology similar in theory to bar code identification. An RFID system consists of an antenna and a transceiver, which read the radio frequency and transfers the information to a processing device, and a tag, which is an integrated circuit containing the RF circuitry and information to be transmitted [2]. The significant advantage of RFID systems is the non contact, non line-of-sight nature of the technology. Tags can be read through a variety of substances such as snow, fog, ice, paint, and other visually and environmentally challenging conditions, where barcodes or other optically read technologies would be useless. In addition, RFID tags can also be read in challenging circumstances at remarkable speeds, in most cases responding in less than 100 milliseconds. Therefore, RFID has become indispensable for a wide range of automated data collection and identification applications that would not be possible otherwise. RFID technology can be applied to various applications. Mainly, it supports transportation, distribution, industrial and animal identification applications.

To implement a reliable and efficient ubiquitous environment employing RFID, however, we should manage RFID information efficiently and exchange them around heterogeneous systems in a ubiquitous network [3][4]. In this paper, therefore, we propose the information server managing RFID data and apply it for physical object distribution application.

The rest of the paper is organized as follow. In Section

* This research is supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Science and Technology(MOST) 21 Century Frontier Project R&D Program in Korea

2, we introduce some of related works about information service for ubiquitous network. In Section 3, we first discuss the required functionality of information service and information process to perform several functions. Then, we propose a language for representation and exchange. In Section 4, we describe the implementation details, and Section 5 is dedicated to conclusion.

2. Related Works

Recently, many researches have been performed to construct ubiquitous network based on RFID technology. In particular, Auto-ID center [5] developed a software solution for ubiquitous network and recently was expanded into EPCglobal [6], connecting with powerful international enterprise such as Wal*Mart, Gillette, and Hewlett-Packard. As a result, RFID-based ubiquitous network is being applied and tested in the practical application fields. In Japan, Ken Sakamura, professor of Tokyo University and director of Ubiquitous ID center, has led many researches for ubiquitous network. In this chapter, we describe their major contributions and examine the information service in particular.

2.1 Information service of Auto-ID center

In increasingly competitive global market, the key to be a successful enterprise is the relative efficiency with which the enterprises manage their assets. According to the needs of business fields, Auto-ID center aims to make enterprise able to control and track its assets. Such enterprise's ability makes it possible to deliver the right goods and services to the right place at the right time, in the proper quantity, quality at acceptable cost.

To achieve the goal, Auto-ID center employs four major components, electronic tags, Electronic Product Code (EPC), Physical Markup Language (PML) [7], and Object Naming Service (ONS). As an Electronic tag component, they used RFID tags. Electronic product code is a method for uniquely identifying physical objects. An EPC is a 96-bit code divided into four partitions; header, EPC manager, object class, and serial number. Not only one physical object but also a batch or lot of items, an EPC is assigned to. In case of temporary arrangements of objects such as shipments, pallet configurations and assemblies, they use the virtual EPC. A virtual EPC does not describe any physical object, but simply the configuration of physical objects. Physical Markup language is a common language for describing physical objects, processes and environments. It is used for remote monitoring and control of the physical environment. Finally, ONS gives the glue which links the EPC with its associated data file. ONS is a kind of an automated networking service and similar to DNS in Internet.

Among components of Auto-ID system, PML is for data storage and management in ubiquitous network. PML uses the same format and structure as the XML so we can use existing standards for data transmission and syntax such as HTTP, TCP/IP, and XML. Translation between schemas is unnecessary and reliable transmission and interpretation is provided since PML has a common implied schema. There is one PML instance per one data element. To manage PML information, they design and implement a PML server [8]. The PML server stores information about objects, batch orders and manufacturing recipes and makes the information available over the ubiquitous network. In the PML server, EPC is used as a key to access the PML data. Only an EPC is stored on the tag and the remaining product data is stored on networked information systems. When data need to be exchanged, they should be converted to PML format. That is, PML is a communication language. Therefore, PML server should translate data value represented by existing data format to the PML format. Through the translation process, PML server offers object information to other heterogeneous systems.

However, information service architecture of Auto-ID center operates well only in normal condition. They predefine the supply chain of objects, and assume that flow of both objects and data always follows predefined supply chain without any failure. However, their assumption makes ubiquitous network easy to break down. Nevertheless, they do not describe method for error recovery

2.2 Information service of Ubiquitous ID center

In 2003, Ubiquitous ID center [9] developed basic technologies to recognize objects using RFID, and to offer suitable services based on the environmental context and RFID information. Based on the basic technologies, they proposed the ubiquitous code (ucode) architecture supporting various code carrier media from the current optical barcode, to IC chips with advanced cryptographic functions.

In ucode system, codes that identify physical objects are stored on ucode tags, and then some additional information is stored within the local capacity. A reader that obtains information from ucode tags is called as Ubiquitous Communicator (UC). A UC accesses to information server according to the obtained ucode and receives the corresponding information. In the ubiquitous computing environment, the number of ucode tags and information servers available in the world is enormous. Therefore, they employ a distributed directory database, called as ucode resolution server, to maintain relationship between the ucodes and information servers.

3. Information Service System in Ubiquitous Environment

Today, the computer technology is improving at an impressive rate and advances furthering the realization of ubiquitous computing are announced almost daily.

Much of research and development activity in this area is focused on the hardware devices and the communication technologies. Of course, improved hardware and networking are clearly central to the development of ubiquitous computing, but an equally important and difficult challenge is data management in ubiquitous network [3].

In this paper, we first address the demand for data management in ubiquitous net-work environment and then propose an information service system to provide valuable information to all heterogeneous systems in the network and users. As an effort to this goal, we developed a prototype information system called information server and apply it to the physical object distribution. Physical object distribution indicates the whole process of an object from production to sale. In general, a physical object is produced, treated, transported, and then sold, before the object is consumed by a user. An example of physical object distribution is shown in Fig.1 where describes architecture of information service system for physical distribution in ubiquitous network.

3.1 Architecture of Information Service System

Information service requires that information should be managed consistently, and offered to users on their demand. Our prototype information service system embodies several functionalities for data management in ubiquitous network. Moreover, it is robust against errors while existing systems are not.

To be robust against various errors, our information server performs 3-step information processing; authentication, flow check, and information validation. By the 3-step information processing, our information server can manage data efficiently, control access to the information server, and recover from distribution flow errors such as human-made error (e.g. transformation error) or network failure.

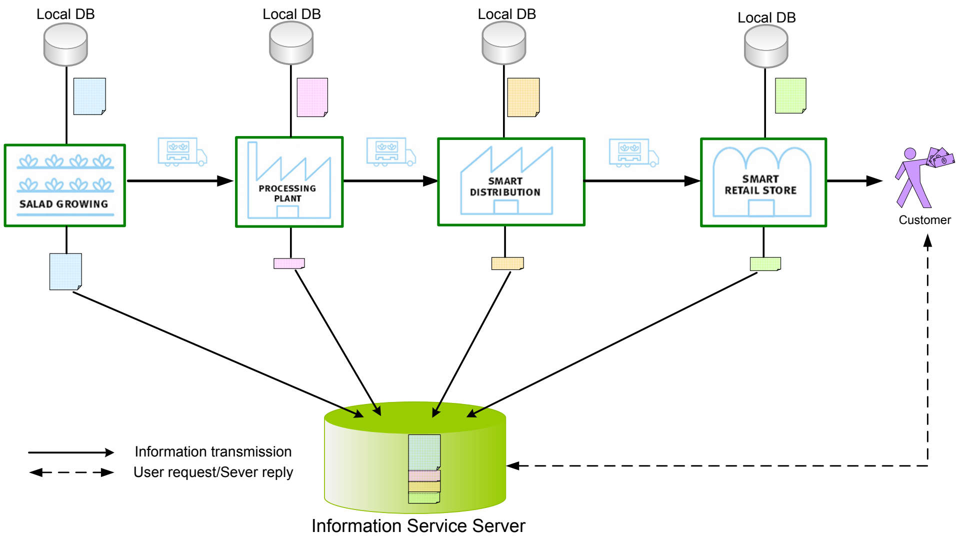


Fig. 1 Architecture of information service system for physical distribution application in ubiquitous network

3.1.1 Information processing

Information received from local database is processed in the following order.

1) Authentication - Information server checks whether a user who tries to access, has proper access right or not.

2) Flow check - At this step, the server confirms whether flow of a physical object and network condition are normal. To do this, the most recent destination of the object, which is stored in the server, is compared with identification of the system. Here, destination is represented by the identification of the receiver

3) Information validation (optional) - if there is a predefined set of rules, the server validates all the information based on the rule set. At this step, the server checks the semantic of information, not the syntax. (For example, transportation time can not exceed 10 hours)

3.1.2 Functionalities

Some of the important functionalities of the information server are as follows.

- Data management - When a physical object is produced, information about the object is generated. Information server collects such generated information and stores it. As the object flows along the path of distribution, new trace information about the object is generated. Therefore, information server should collect and maintain trace information following object's path [10].

- Access control - When a user or system attempts to access information about a physical object stored in the information server, the server confirms or denies such access based on the access right. If the user requests information which includes some beyond the access right, the server only offers authorized portion of information.

- Flow error recovery - Distribution process might be incorrect due to several reasons. For example, due to some network errors, the distribution flow can go wrong. Fig.2 shows a scenario where a transformation error happens and Fig.3 shows a network error. The information server should be robust against such errors [11]. To recover from such distribution flow errors, all local systems generating information about physical objects have membership identification. For example, when objects leave a local company (cid1 in Fig. 2), the local system sends destination information of the objects to information server in the center (step ① in Fig.2). When the objects arrive at another company (cid3 and step② in Fig.2), the local system sends its identification to the information server (step ③). Then, the information server compares destination information obtained from sender, with identification of receiver (step ④ in Fig.2). If they match, distribution flow was right. A pack flow between the information server and the local system in the normal case is shown in Fig. 4 (a). However, if they do not match, the flow was wrong. Such flow error can be caused by several reasons. For example, network failure or a mistake of truck driver can be the reason. A pack flow in the abnormal case such as authentication failure or distribution flow error is shown in Fig. 4 (b) and (c). To recover from distribution flow errors, we put the destination of each distribution step into object information. For the representation of object information and its exchange among systems (including information server), we need common information markup language.

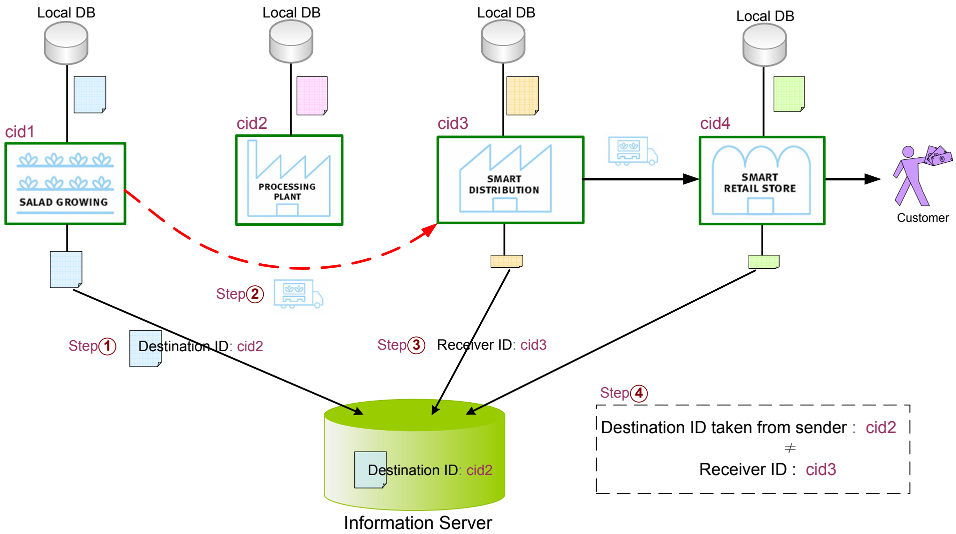


Fig. 2 Distribution flow error: transformation error

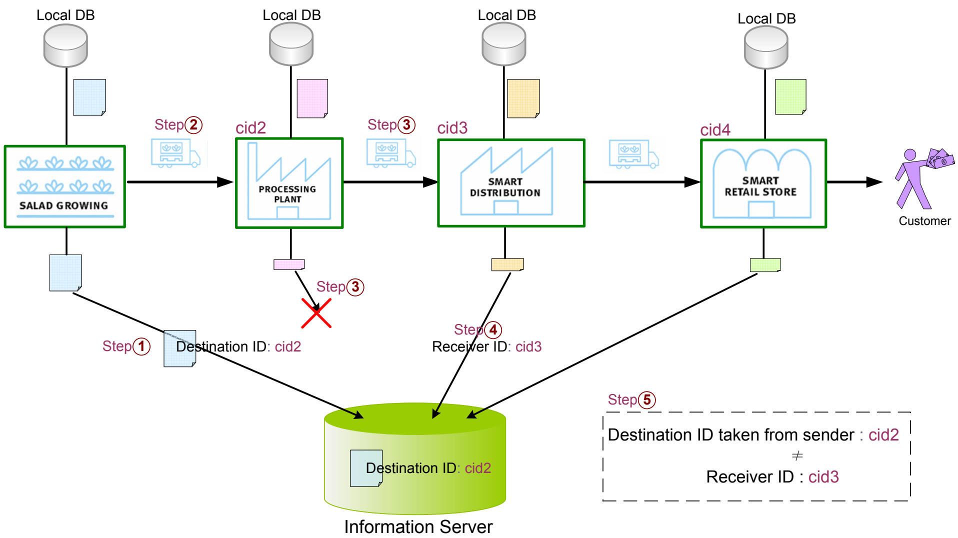
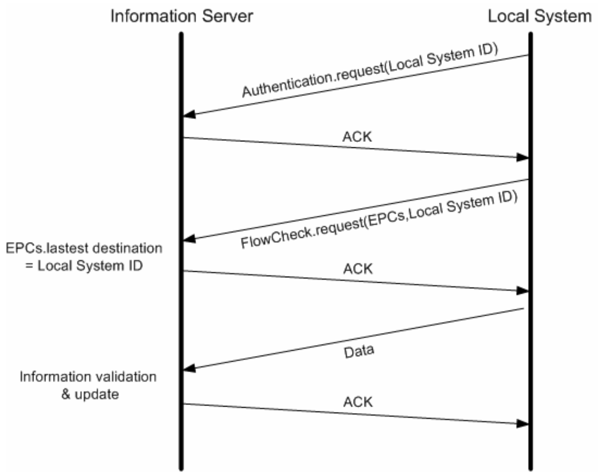
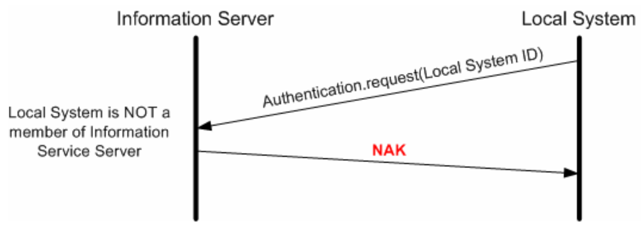


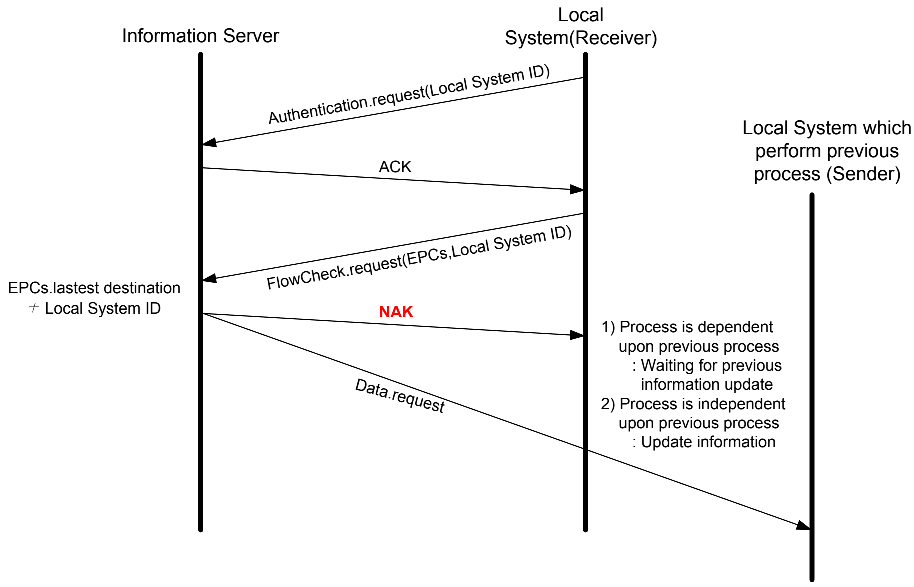
Fig. 3 Distribution flow error: network failure



(a) Normal case: Authentication, distribution flow, and data update is normal



(b) Exceptional case: Authentication failure



(c) Exceptional case: Distribution flow error

Fig. 4 Pack flow between information server and local systems

3.2 EPML (Extended PML)

Ubiquitous network can be constructed with heterogeneous systems. It means that data representation form of one system can be different from that of another system. There-fore, to make it possible to exchange information among heterogeneous systems, in-formation server should be able to understand the data representation forms of all the systems in ubiquitous network. However, to understand different representations could be a very exhaustive job. As a solution to the problem, a common representation for-mat called PML was proposed.

PML is an instance of XML, so PML also has the same format and syntax as the XML. PML was proposed by Auto-ID center. As we mentioned above, the system implemented by Auto-ID center is not able to recognize distribution flow errors be-cause PML does not contain any information about distribution sequence of objects. When their PML server takes trace information of objects, the server can not judge whether the distribution flow is correct or not. However, inability to control the distribution flow can cause serious problems. Let's assume that an object should go through various chemical processes in order. In this case, if one step of the chemical processes is missed, the information server should know that. If a chemical factory continues to process the object without knowing that missed step, the object might be ruined. Therefore, the information server should be able to control the distribution flow of objects. To make the information server control distribution flow of the objects, we modified and extended the PML, which we will call EPML in the paper.

To know the process sequence of objects, the information server requires the destination information of each step. The destination information is represented by the “destination” element in the EPML. The value of the destination element is an identification of the destination company. In our system, every company on the supply chain has identification represented by “ID” attribute of owner element in the EPML. The EPML schema is shown in fig. 5

In addition, the existing PML targets on physical object distribution application. However, information service in ubiquitous network is needed in other ubiquitous applications. Ubiquitous network for patient management in a hospital or for wildlife research also needs the information services to manage information efficiently. How-ever, existing PML can not represent such patient or wildlife information well enough. Therefore, we need a more general object markup language which can represent other type objects as well as physical objects.

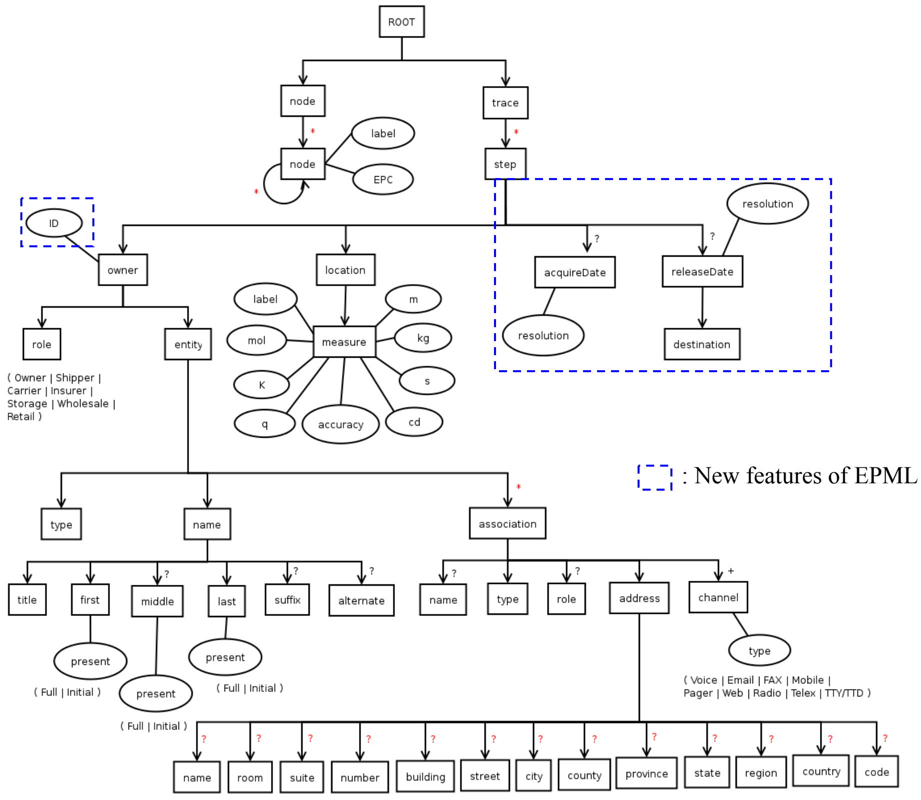


Fig. 5 Schema architecture for EPML

4. Implementation

Based on the system architecture, we implemented a prototype information service system for physical object distribution in ubiquitous network. The information server is based on FreeBSD operation system and employs the postgresql [12] database. It is implemented in java language and Castor [13] is used to convert relational data in postgresql to an EPML document.

Information server has two access groups: member and user. For the member access, the information server takes the object information and control the distribution flow. When an object is produced by a production factory, information about the object is created, and then the created information is sent to the information server. After the production factory send the information, the object can be transported to other factory to go through other processes, such as chemical or physical processing, assembling, packing, and so on. If the other factories is received the object for remaining processes, the factories work its process. As the result of object process, the process factory generates the additional information about the object, and then sends the trace information about the object to information server. Such information update is continued until that the object is consumed by users. To update trace information about the object, each factory on distribution line should perform the information process steps. First of all, they should be authenticated as a member of physical object distribution line.

If the information server confirms the membership of the factory, then the factory sends EPCs of objects, which the factory is going to process, to the information server for distribution flow check. When the information server takes EPCs of the objects, it compares the identification of the factory with the latest destination identification of the objects, which were stored in information server. If they do not match, it is clear that the flow was wrong somewhere. Therefore, the information server sends an error message to the factory, and then the factory will try again. A sample error message sent by the information server is shown in Fig.6.

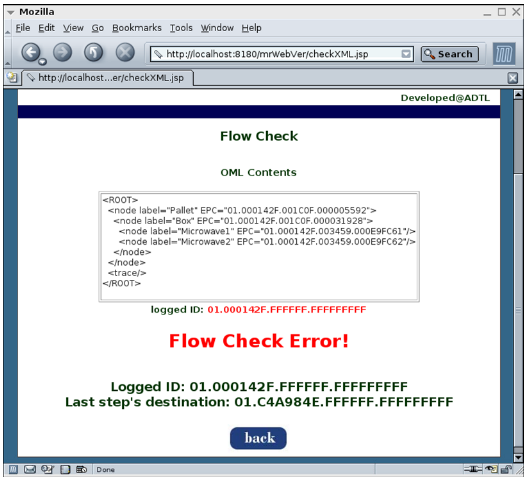


Fig. 6 Message for distribution flow error

However, if they match, then the distribution flow is considered to be correct. Fig.7 shows a massage for flow check approval. Here, the information server has old information about the objects which was generated by the previous factory then stored in the information server. Fig. 8 shows the data before the information generated by the factory is updated.

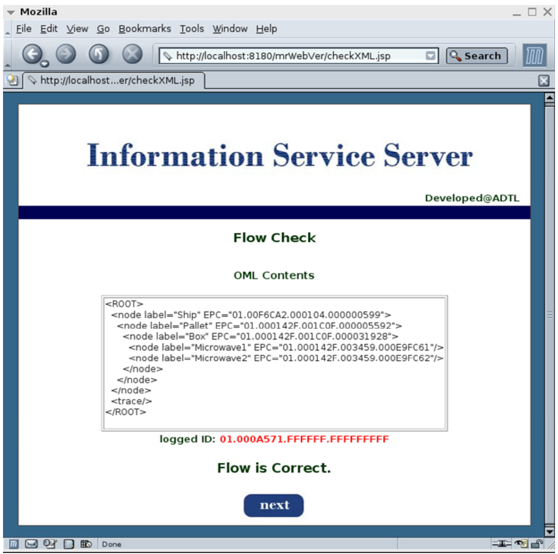


Fig. 7 Message for information flow approval

epc	label	pepc
01.00F6CA2.000104.000000599	Ship	NULL
01.000142F.001C0F.000005592	Pallet	01.00F6CA2.000104.000000599
01.000142F.001C0F.000031928	Box	01.000142F.001C0F.000005592
01.000142F.003459.000E9FC61	Microwave1	01.000142F.001C0F.000031928
01.000142F.003459.000E9FC62	Microwave2	01.000142F.001C0F.000031928

epc	sid
01.00F6CA2.000104.000000599	1
01.000142F.001C0F.000005592	1
01.000142F.001C0F.000031928	1
01.000142F.003459.000E9FC61	1
01.000142F.003459.000E9FC62	1

sid	owner_epc	role	type	dest_epc
1	01.000142F.FFFFFF.FFFFFFFFF	Owner	Person	01.000A571.FFFFFF.FFFFFFFFF

Fig. 8 Data in the information server before information generated by the factory is updated

Hence, the information server takes the information about the object from the cur-rent process factory, and then updates trace information about the objects. After information update, the information server has new information generated by the factory. The updated data in the information server is shown in Fig. 9.

epc	label	pepc
01.00F6CA2.000104.000000599	Ship	NULL
01.000142F.001C0F.000031928	Box	01.000142F.001C0F.000005592
01.000142F.003459.000E9FC61	Microwave1	01.000142F.001C0F.000031928
01.000142F.003459.000E9FC62	Microwave2	01.000142F.001C0F.000031928
01.02E1259.0013C9.000030F91	Truck	NULL
01.000142F.001C0F.000005592	Pallet	01.02E1259.0013C9.000030F91

epc	sid
01.00F6CA2.000104.000000599	1
01.000142F.001C0F.000005592	1
01.000142F.001C0F.000031928	1
01.000142F.003459.000E9FC61	1
01.000142F.003459.000E9FC62	1
01.02E1259.0013C9.000030F91	2
01.000142F.001C0F.000005592	2
01.000142F.001C0F.000031928	2
01.000142F.003459.000E9FC61	2
01.000142F.003459.000E9FC62	2

: Updated information

sid	owner_epc	role	type	dest_epc
1	01.000142F.FFFFFF.FFFFFFFFF	Owner	Person	01.000A571.FFFFFF.FFFFFFFFF
2	01.000A571.FFFFFF.FFFFFFFFF	Shipper	Person	01.C4A984E.FFFFFF.FFFFFFFFF

Fig. 9 Data in the information server after information generated by the factory is updated

For the user access, the information server takes the EPC of an object from users, and then displays the information about the object containing production, process, transportation information, and so on. Of course, systems on the distribution line can access to the information server as a user, when they want to know about objects. However, to do this, the system should already know the EPC information of the ob-jects. A snapshot of information screen offered to the users by the server is shown in Fig. 10.

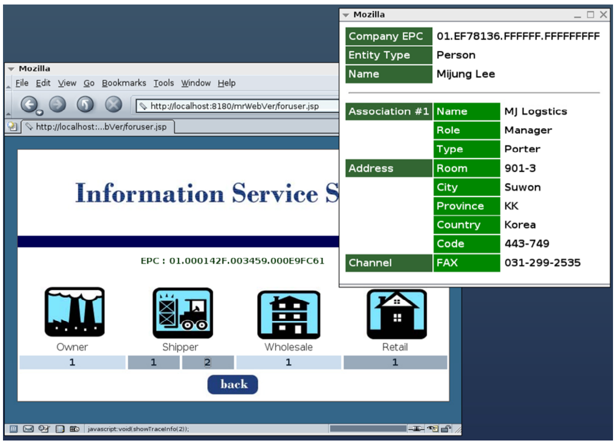


Fig. 10 Data in the information server after information generated by the factory is updated

5. Conclusion

Ubiquitous network is a promising area for the future that is moving closer to realiza-tion by improved hardware and networking technology. However, to be more practical ubiquitous network, ubiquitous computing systems should be able to manage informa-tion efficiently. In this paper, we proposed an information service system to store and exchange data between heterogeneous systems in ubiquitous network for physical object distribution. The proposed system can control the distribution flow to prevent erroneous processing and provide the object information to users.

To recover from distribution flow errors such as human errors (e.g. transformation error) or network failures, our information service system keeps the destination infor-mation of each distribution step. Such destination information is represented by EPML, which is an extension of the existing PML. The proposed EPML has new features. Fist of all, identification of local system is represented at each trace step in EPML, be-cause every local system has a unique identification in our information service system. In addition, local systems should report destination of the objects to the information server. Such destination information is represented in each trace step of EPML. Using destination information, the information server can recognize a distribution flow errors, and recover from such errors. We expect that flow control feature of our information service system will make physical distribution applications in ubiquitous network more practical.

References

1. Gregory D. Abowd and Elizabeth D. Mynatt.: Charting past, present, and future research in ubiquitous computing. ACM Transactions on Computer-Human Interaction, Vol. 7(1), (2000) 29-58
2. The AIM Global Network, What is RFID?, http://www.aimglobal.org
3. Michael J. Franklin.: Challenges in Ubiquitous Data Management. Lecture Notes in Computer Science, Vol. 2000. Springer-Verlag, (2001)
4. McFarlane. D: Towards Auto ID Based Control Systems. Proceedings of IEEE SMC, Tunisia (2002)
5. Sun's Auto-ID Center, http://www.autoidlabs.org
6. EPC Global, http://www.epcglobalinc.org
7. David L. Brock, Timothy P.M, et.al.: The Physical Markup Language, Auto-ID center's White paper, June 1, (2001)
8. Mark Harrison, Duncan McFarlane.: Development of a Prototype PML Server for an Auto-ID Enabled Robotic Manufacturing Environment, Auto-ID center's White paper, February 1, (2003)
9. Ubiquitous ID Center, http://www.uidcenter.org
10. Ministry of Economy, Trade and Industry, Interim Report, Conference on of Product Traceability, Japan (2003)
11. Tim Kindgerg and Armando Fox.: Systems Software for Ubiquitous Computing, IEEE Pervasive Computing, (2002) 70-81
12. PostgreSQL, http://www.postgresql.org
13. Castor project, http://castor.exolab.org