Master's Thesis Academic Year 2007

An Architecture for Multiple, Large User-specific Data in a Networked RFID

Graduate School of Media and Governance, Keio University

Kosuke Osaka

Master's Thesis Academic Year 2007

An Architecture for Multiple, Large User-specific Data in a Networked RFID

by
Kosuke Osaka

BACHELOR OF ENVIRONMENT AND INFORMATION STUDIES
KEIO UNIVERISITY, 2006

Submitted to the Cyber Informatics Division in Partial Fulfillment of the Requirements for the Degree of

MASTER OF MEDIA AND GOVERNANCE

at KEIO UNIVERSITY

JANUARY 2008

Thesis Supervisor: Jun Murai
Title: Vice-President, Keio University
Professor, Faculty of Environment and Information Studies
Chair, Auto-ID Labs Board of Directors

Thesis Supervisor: Osamu Nakamura
Title: Professor, Faculty of Environment and Information Studies
Associate Director, Auto-ID Lab. Japan

Thesis Supervisor: Jin Mitsugi
Title: Associate Professor, Graduate School of Media and Governance
Associate Director, Auto-ID Lab. Japan

Abstract of Master's Thesis Academic Year 2007

An Architecture for Multiple, Large User-specific Data in a Networked RFID

Summary

As RFID technology has been widely adopted, it has been acknowledged that user-specific data, as well as a unique identifier, carried by RFID tags, can be conveniently used. The typical user-specific data stored in an RFID tag is sensor data. There is also an industrial demand to include non-sensor data, such as sales records, repair history, etc., as well as sensor data. It is expected, therefore, that the future information system for networked RFID needs to incorporate heterogeneous user-specific data in an RFID tag in a consistent architecture. There are two fundamental challenges to handle such heterogeneous user-specific data in RFID tag. The first one is the establishment of a logical link between the application and the user-specific data in physical memory in a RFID tag. The other is the efficient retrieval of large user-specific data in RFID tag in practical radio noise environments. An illustrative example of large user-specific data is the retrieval of some kind of logging data in an RFID tag memory. In this thesis, new functions named "schema resolver" and "reader controller" at middleware layer are introduced to meet these two challenges in a networked RFID system. The schema resolver provides the link between the object identifier, which identifies the target user-specific data in the application interface, and the schema of RFID tag memory. Since the fundamental drawback of the schema resolver might be the lookup overhead, the overhead is estimated with a simulation based on experimental data. It was revealed that the overhead is equivalent to and in some cases even better than that of the existing method, which exclusively relies on the memory schema information in a tag memory. The retrieval time of a large data from an RFID tag was also examined in an experiment. The result indicated that it took a lot of time for large RFID tag data retrieval. An adaptive block size control mechanism in a reader controller is proposed in this thesis to solve the problem. The mechanism features the automatic reading block length adjustment, based on the retransmission frequency. We confirmed the priority of the proposed method from the mathematical simulation and the experiment.

Keywords

1, Networked RFID Middleware   2, Sensor Embedded RFID tag
3, Battery Assisted Passive Tag   4, User Data Collection Performance
5, RFID Tag Memory Schema Processing
6, Efficient Large RFID Tag Data Retrieval Method

Graduate School of Media and Governance, Keio University
Kosuke Osaka

修士論文要旨 2007 年度 (平成 19 年度)

An Architecture for Multiple, Large User-specific Data in a Networked RFID

論文要旨

近年、主に物流行程におけるモノの管理 (Supply Chain Management) を想定して標準化が進められてきた RFID 技術だが，物流行程でのモノの管理以外の，新しい利用モデルへの期待が産業界で高まりつつある．そうした利用モデルが拡大する中，RFID タグ自体に様々なユーザデータを保持させるモデルの有用性が分かってきた．当初代表的なユーザデータの例として、温度や湿度情報といったセンサデータなどから考えられたが、最近ではセンサデータ以外のユーザデータ，例えば RFID タグが貼付されたモノに関する様々な情報をもユーザデータとして取り扱いたいという産業界からの要望も出始めている．このような多様なユーザデータへの要求に統一的な仕組みで対応するためには，各 RFID タグのメモリスキーマ情報を解決すること，大量かつ多様なユーザデータの無線通信の効率化を解決することが必要となる．これらの課題を解決するため，本研究では Schema Resolver と Reader Controller という 2 つのミドルウェアレベルでの機能追加を提案する．Schema Resolver は RFID reader/writer が各 RFID タグから必要なユーザデータを読み取るために必要なタグのスキーマ情報解決を行う．Reader Controller は RFID タグからの大量のユーザデータ読み取りを効率化するため，外部のノイズエラーの影響を考慮したユーザデータの分割読み取り機能を実現する．

Schema Resolver は，スキーマ情報を解決する時間 (オーバーヘッド) の長さが弱点となりうる．本研究では一般的なノイズ環境下でのオーバーヘッド計測を、シミュレーションと実機を使用した実験計測の双方で評価を行った．その結果本研究の提案手法におけるオーバーヘッドは実用上問題ない (約 12msec) ことが分かった．Reader Controller は，大量のユーザデータ読み取り時における本研究で提案する分割読み取り手法と，従来の一括読み取り手法の読み取り速度の評価を行った．シミュレーションによる分割読み取りの速度評価と，実機を用いた分割読み取り速度の評価を行い，本研究で提案する Reader Controller の有用性を確認した．

キーワード:

1, ネットワーク型 RFID ミドルウェア　2，センサ統合型 RFID タグ
3, 電池搭載型パッシブタグ　4, ユーザデータ読み取り速度評価
5，RFID タグメモリスキーマ処理
6，大量 RFID タグデータ読み取り処理

慶應義塾大学大学院 政策・メディア研究科
苧阪 浩輔

Acknowledgments

First of all, I express my deep appreciation to my supervisors, Professor Jun Murai, Professor Osamu Nakamura and associate Professor Jin Mitsugi. They have given me continuous guidance and invaluable suggesion since I joined Murai Labolatory / Auto-ID laboratory. Regarding my master thesis, without their shepharding, I couldn't construct the logic of my thesis.

I also appreciate to Auto-ID Lab. staffs, Dr.Hisakazu Hada, Mr.Shigeya Suzuki, Mr.Tatsuya Inaba, Mr.Masafumi Nakane and Mr.Osamu Tokumasu (Panasonic Co., Ltd./Reseacher at Keio Univ.) for thier guidance and support on a routine basis.

I also would like to thank to Mr.Yuusuke Kawakita. I had got a lot of supports from him since I was a freshman in the laboratory. I got a lot of wisdom related with my thesis through the discussion with him.

Thanks also to my colleagues in Auto-ID Lab. for their daily assistance and encouragement. Especially I have to express my thank to Mr.Akira Yamamoto (Keio University, Department of Science and Technology), Mr.Naoyasu Kamiya, Mr.Taisuke Sato and Mr.Keigo Emura. I also thank to Mr.Ryu Sato, Mr.Tatsushi Fukui, Mr.Kenichi Sugimoto, Mr.Shuuhei Yamaguchi, Miss.Sonryo Kim and Mr.Masahiro Yamada. I could enjoy studying with them.

I want to appreciate to the master second year students in other research group in Murai Laboratory. I want to appriciate to Mr. Achmad Basuki, Mr.Yuuki Oyabu, Mr.Yuuichi Nakamura and Mr.Masahiro Mizutani.

I got a chance to study here in Keio University because of the support and understanding of my family. Lastly but most importantly, I would like to thank to my family for everything .

January 15, 2008

Kosuke Osaka

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, after the introduction of the background of this research, the problems to be solved are introduced. The objective of this research and the organization of this thesis are then described.

1.1 Background

In the past years, the networked RFID system, which is a combined technology of Radio Frequency IDentification (RFID) technology and Internet technology, has been designed and realized mostly for asset identification in supply chain management (SCM). Figure 1.1 shows a typical use case model in SCM.

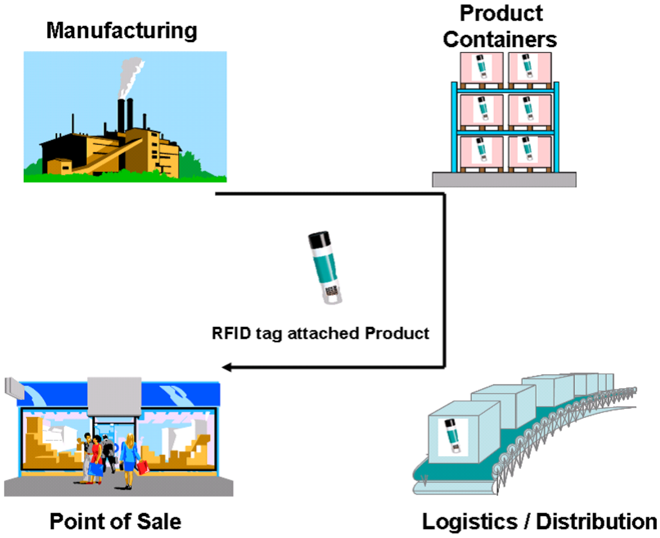

Figure 1.1: RFID tag attached the product can be tracked in SCM

EPCglobal network architecture [1], considered to be the de facto standards of the networked RFID system, has been designed to satisfy requirements for Supply Chain Management (SCM). Such a typical networked RFID system consists of a lot of components hierarchically (e.g. Application, Middleware, RFID reader/writer, RFID tag and so on)(Figure 1.2).

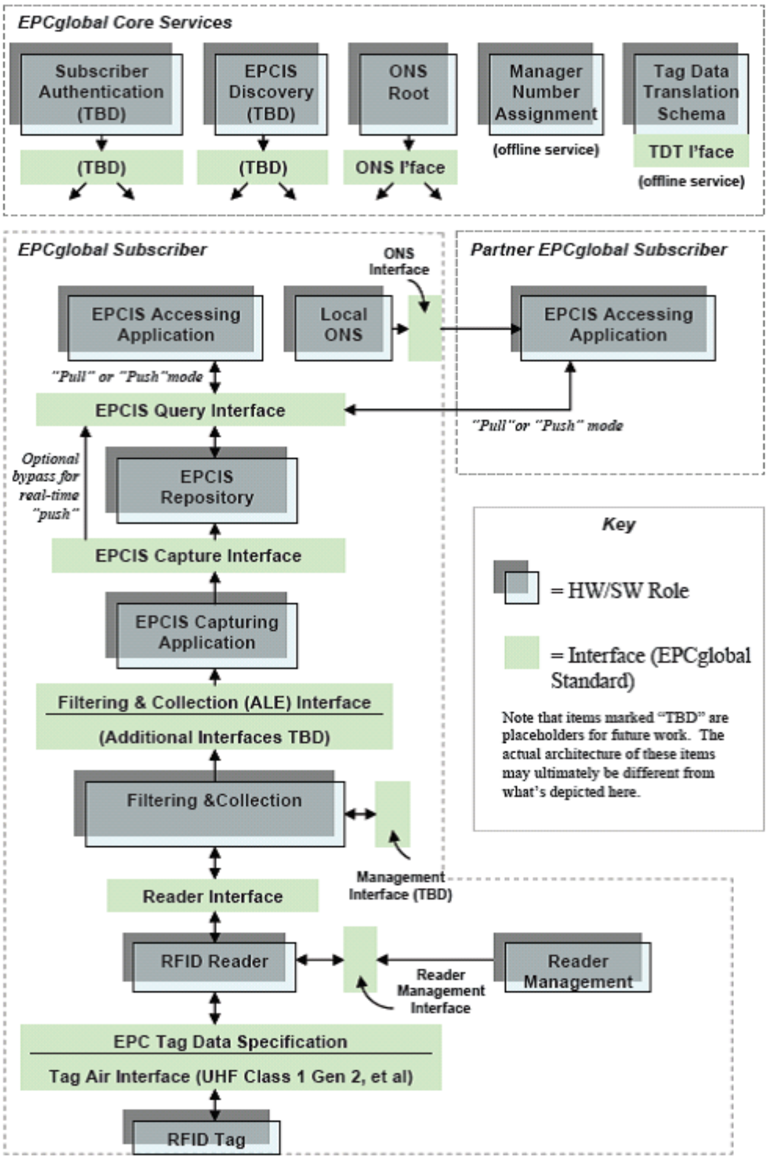

Figure 1.2: EPCglobal architecture reference model [1]

In such a reference model, an RFID tag basically only contains a globally unique identification number such as EPC (Electronic Product Code) [6]. An RFID reader/writer retrieves the raw data (ID information) from the RFID tag, and the raw data is transferred to an middleware (sometimes referred to as Filter and Collection). In the middleware layer, the raw read data is transformed to an event in conjunction with the timestamp. The event data is transferred to the capturing application and then shared with an associated repository service and query applications.

An exemplary of the user-specific data can be found in sensor integrated RFID systems, which records environmental information into the existing RFID system. Using variety of sensor devices attached on an RFID tag, the RFID tag (referred to as a sensor tag) enables to realize value added asset identification. For example, IBM Corporation proposed sensor integrated RFID solution, especially for temperature information in conjunction with unique ID [7] (such a system is called as Cold Chain Management in other literature [8]). Figure 1.3 shows an example RFID tag device which is used in IBM-DHL case example. In this manner, sensor integrated RFID system can realize not only asset traceability but the management of the proper temperature of the merchandizes which provides the additional value.

Figure 1.3: Temperature sensor device installed RFID tag

Figure 1.4: High capacity UHF tag (256Byte) [2]

Another exemplary of user-specific data can be found in life cycle management of consumer electronics. Major consumer electronics manufacturers have been conducting a research and testing on how to manage whole of the life cycle of the consumer electronics by using RFID technology [9][10][11]. They recommend storing the variety of data such as the manufacturing specific code, shipping, sales, repair history and more information in its own high-capacity tag memory. An example of the existing high capacity tag is shown in Figure 1.4 [2] Similar discussion has been taking place in the life cycle management of aircraft parts [12].

1.2 Problem statement

When it comes to realize the user-specific data handling in a networked RFID system (Figure 1.5), the reference model architecture such as EPCglobal network needs to incorporate the issues which are to be indicated as followed.

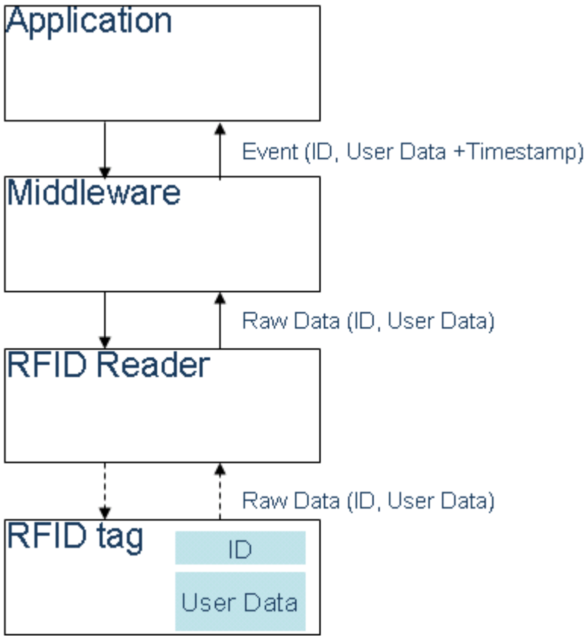

Figure 1.5: Future model of networked RFID architecture

How can an RFID reader/writer resolve each tag memory structure?

In the future, each RFID reader/writer needs to incorporate variety types of user-specific data in the heterogeneous types of RFID tag data. When an RFID tag comes to contain not only ID information like an EPC but also variety types of user-specific data in an RFID tag memory, it is desirable that an RFID reader/writer can retrieve tag data from several RFID tag memory area efficiently. In the consequence, the RFID reader/writer must have a priori knowledge of the each RFID tag memory schema (information of memory address mapped onto the object identifier, which is specified by the associated applications) before the retrieval of actual RFID tag data.

How do we efficiently transfer data between RFID reader/writer and tags?

Secondly we have to consider the efficient transferring tag data between RFID reader and RFID tag, particularly when an RFID tag comes to store a large amount of user-specific data such as logging sensor data.

It should be noted that, for the practicality of the study, we lend ourselves not to change the existing air protocols between the RFID reader and the RFID tag. The challenges, therefore, resides in the middleware (Filter and collection role) in the EPCglobal network architecture in Figure 1.2

1.3 Nomenclature

Active tag
: Has its own internal power source for which the data communication between the tag and an RFID reader/writer and the integrated circuits. Due to its onboard power supply, an active tag can transmit at higher power levels than passive RFID tag.

Application
: A set of procedures to produce output data from input data. The procedure is usually carried out by computers

Architecture
: A set of roles and interfaces in and among networking entities such as interrogators, sensors, tags, applications and information services

BAP
: Battery Assisted Passive tag

Block size
: An unit memory size of which RFID reader should retrieve data from RFID tag memory at one time

Data record
: A record or memory field storing data

Data base (repository)
: A composition of ”data” records

EBV
: Extensible Bit Vector. A data structure with an extensible data range.

Entity
: A logical unit representing a group of roles

EPC:Electronic Product Code
: A 96 bit code system of unique identifier in RFID tag

Event
: The occurrence of a change of significance in the physical state or information content of an agent

Information service
A software which is designated to serve another application

Interface
Standardized procedure to exchange information among entities

Interrogator
General expression for RFID reader/writer and sensor base station. Interrogator has communication capability with network entity

Inventory
The operation of identifying tags. Through the RFID inventory, a RFID reader detects the population of the RFID tags in the read zone.

Message
The exchange of information between agents

meta-data
is data about data. An item of meta-data describes an individual datum, or content item, or a collection of data including multiple content items.

Networked RFID
RFID system in which RFID tags, interrogators, applications are communicating. Basically an RFID tag only contains ID information and the other data related with the RFID tag are stored in the information services in the network.

Object identifier (OID)
A value (distinguishable from all other such values) which is associated with an object

Passive tag
An RFID tag without a power source for data transmission. When radio waves from the RFID reader reach the antenna of the chip, they create an electro-magnetic field. The tag draws power from the field and is able to send back information stored on the chip.

Protocol
Precise description of a vocabulary and interaction method of agents over air interface or wired connection

PUD
Pointer to User-specific Data. PUD indicates the offset address where the user-specific data is stored.

PPUD
Pointer to PUD. PPUD indicates the offset address of the PUD.

Query
A question posed in a formal language and relating to a relevant dataset in the system

Read event
A logical unit of interrogator read comprises Time stamp, Interrogator ID, Tag/Sensor ID and additional data (sensor data)

RFID tag
A microelectronic device including a memory and antenna for wireless data communications. Each RFID tag is assigned a unique ID. RFID tags are classified into passive RFID tag, semi-passive RFID tag and active RFID tag

Role
A particular function

SCM
Supply Chain Management

Semi-passive tag
Similar to active tags in that they have their own power source, but the power sources only assists running the power circuit but for signaling to the RFID reader.

Sensor
A device to measure physical amount quantitatively

Sensor tag
Tags, which have sensing capability. Examples are Class II∼IV tags in EPC. ID information does not confined to be EPC

SNR
Signal to Noise Ratio

Schema
A description of how an RFID tag memory is organized.

Unique identification number (Unique ID)
A positive integer representing an entity exclusively in a particular numbering system (for example, EPCglobal, DoD and ATA have their own numbering systems)

1.4 Organization

Related or involved works and studies are described in chapter 2. In chapter 3, firstly the requirements for multiple, large user-specific data handling in a networked RFID are surveyed. Then the proposed architecture overview and the two functions named the schema resolver and the reader controller are introduced. In chapter 4, the schema resolver is explained in detail by comparing with the ISO/IEC proposed model as an representative existing model. The evaluation of the overhead to resolve the tag memory schema information. In chapter 5, the reader controller which features the adaptive block size control retrieval mechanism is introduced. The effectiveness of the proposal is examined by a numerical simulation and an experiment. Finally, Chapter 6 concludes the thesis.

Chapter 2

Related Works and Studies

This chapter describes the related and involved studies of this research.

EPCglobal architecture is the de facto standard of networked RFID model. However, as was mentioned above, the domain of the RFID application has become more and more widely spread and the required functions for the model have been changed. In the EPCglobal standard committee [13], the discussion for user-specific data in an RFID tag handling in EPCglobal network has been started and some specification in the EPCglobal standards has been prepared in the future revisions. ISO/IEC (International Organization for Standardization) [14], which is also an important standardization body in RFID technology, has already started the discussion for user-specific data in an RFID tag handling. Other researches on sensor integrated RFID and the wireless sensor networks technologies are also surveyed.

2.1 EPCglobal Architecture

2.1.1 Architecture overview

The EPCglobal architecture (Figure 1.2) is a reference set of components which is consists of hardware, software and data standards, together with core services that can be operated by EPCglobal, all in service of a common goal of enhancing business flows and computer applications through the use of Electronic Product Codes (EPCs) [6]. The concept was originally proposed by Auto-ID Center and is developed and maintained by EPCglobal. Functions and structures of EPCglobal architecture is defined in [1]. Figure 1.2 shows the reference model. There are two types of components in the model: role and interface. Role is a collection of functions, whereas interface exists between roles. In the figure, some roles are defined separately but this does not necessarily mean that they are separately implemented in hardware or software. Also in an actual implementation, some of the roles and interfaces are not implemented if they are not required the implementation.

RFID tag
RFID tag is a microelectronic device including a memory and antenna for wireless data communications. Each RFID tag is assigned a global unique ID such as EPC. RFID tags are classified into passive RFID tag, semi-passive RFID tag and active tag.

RFID reader/writer
Make multiple observations of RFID tags while they are in the read zone.

Reader interface
Defines the control and delivery of raw tag reads from Readers to the Filtering and Collection role. Events at this interface say "Reader A saw EPC X at time T."

Filtering and Collection
This role filters and collects raw tag reads, over time intervals delimited by events defined by the EPCIS Capturing Application (e.g. tripping a motion detector).

Filtering and Collection (ALE) interface
Defines the control and delivery of filtered and collected tag read data from Filtering & Collection role to the EPCIS Capturing Application role. Events at this interface say "At Location L, between time T1 and T2, the following EPCs were observed," where the list of EPCs has no duplicates and has been filtered by criteria defined by the EPCIS Capturing Application.

EPCIS Capturing Application
Supervises the operation of the lower EPC elements, and provides business context by coordinating with other sources of information involved in executing a particular step of a business process. The EPCIS Capturing Application may, for example, coordinate a conveyor system with Filtering & Collection events, may check for exceptional quality and take corrective action (e.g., diverting a bad case into a rework area), may present information to a human operator, and so on. The EPCIS Capturing Application understands the business process step or steps during which EPCIS data capture takes place. This role may be complex, involving the association of multiple read events from multiple Filter and Collection, as in the loading of a shipment. Or it may be straightforward, as in an inventory business process where there may be "smart shelves" deployed that generate periodic observations about objects that enter or leave the shelf.

EPCIS capture interface
The interface through which EPCIS data is delivered to enterprise-level roles, including EPCIS Repositories, EPCIS Accessing Applications, and data exchange with partners. Events at this interface say, for example, "At location X, at time T, the following contained objects (cases) were verified as being aggregated to the following containing object (pallet)."

EPCIS Accessing Application
Responsible for carrying out overall enterprise business processes, such as warehouse management, shipping and receiving, historical throughput analysis, and so forth, aided by EPC-related data.

EPCIS Repository
Records EPCIS-level events generated by one or more EPCIS Capturing Applications, and makes them available for later query by EPCIS Accessing Applications.

The EPCglobal network can also accommodate the data flow relationship in a large scale network. The following roles are designed for cross-enterprise data related with EPC sharing.

Object Name Service (ONS)
Provides a function for lookup a reference to an EPCIS service or other service associated with an EPC. The list of services associated with an EPC is maintained by EPC Manager for the EPC, and typically includes services operated by the organization that commissioned the EPC.

ONS role can be divided into two types; the ONS Root and Local ONS. The Root ONS provides the initial point of contact for Local ONS which is maintained the target EPCIS services or other services associated with an EPC. The Local ONS fulfills the lookup requests for EPCs within the control of the enterprise that operates the Local ONS; that is, EPCs for which the enterprise is the EPC Manager.

EPCIS Discovery
Note that "EPCIS Discovery" is not yet a defined part, But EPCIS Discovery service will expect the following functions.

- Provides a means to locate all EPCIS services that may have information about a specific EPC.
- May provide a cache for selected EPCIS data.
- Enforces authorization policies with respect to access of the aforementioned data.

2.1.2 Recent standard activity for user-specific data processing in EPCglobal

EPCglobal standard committee has started the discussion for user-specific data handling in a networked RFID system. One of the representative example in the standards is ALE (The Application Level Events) specification [3].

The function of the ALE interface in the present specification within EPCglobal Architecture is to provide the independence between the infrastructure components that acquire the raw EPC data, the architectural components that filter and count that data and the applications that user the data. This allows changes in one without requiring changes in the other, offering significant benefits to both the technology provider and end-user. The current ALE specification provided only an interface for reading data (not writing), and only provided access to EPC data. While on the other hand, the next version ALE specification expands to address writing as well as reading, and both the reading and writing aspects address not only EPC data but also other user-specific data that may be present on RFID tag. To correspond to a variety types of user-specific data handling, not only reading operation but also writing operation, next ALE provides for some new specifications.

ALE interfaces

The new ALE specification defines five interfaces, as defined below.

Reading API
An interface through which clients may obtain filtered, consolidated EPC and user-specific. In particular, clients may read RFID tags using RFID readers.

Writing API
An interface through which clients may cause operations to be performed on RFID tag through an RFID reader.

Tag Memory Specification API
An interface through which clients may define symbolic names that refer to data fields of tags.

Logical Reader Configuration API
An interface through which clients may define logical reader names for use with the Reading API and the Writing API.

Access Control API
An interface through which clients may define the access rights of other clients to use the facilities provided by the other APIs.

ALE and ALE client application communicate with the fundamental unit referred to as an event cycle in a reading API or a command cycle in a writing API. A client application specifies the ECSpec (Event Cycle Specification) to subscribe the associated RFID tag data reading of associated RFID readers. The ALE produces a report to the client application for the subscribed ECspec. The interaction may be push or pull.

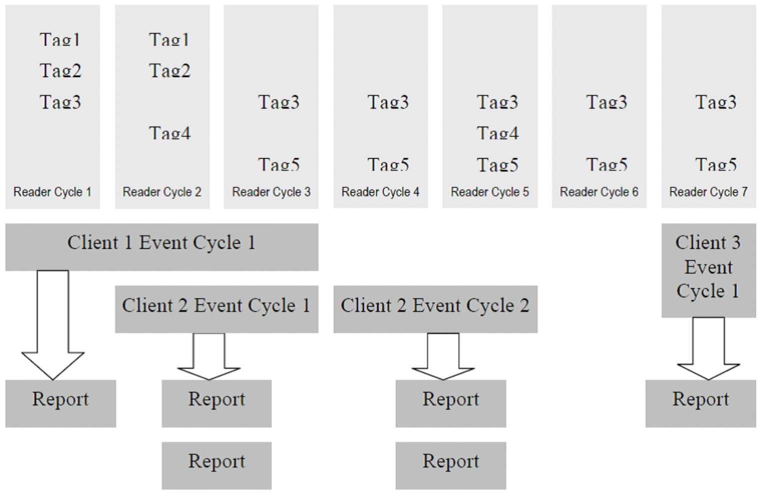

Figure 2.1: An example of event cycle model at ALE interface [3]

For the access of user-specific data in an RFID tag, ALE defines a fieldspec which is composed of fieldname, data type and format. The fieldname may be the memory bank, memory offset and its length (fixed-address fieldname)or object identifier or a symbolic fieldname(variable fieldname). The symbolic fieldname can be retrieved or created by using Tag Memory API. There is a reserved fieldname called UserBank fieldname, similar to EPCBank and TIDbank, which specifies the beginning of the each bank to end of data.

By specifying TMSpec(Tag memory specification), the ALE can retrieve a set of fieldname, memory schema of an associated tags. ALE also can create a new fieldspec by using define TMspec API. TMSpec may be TMFixedFieldSpec or TMVariableFieldSpec. A TMFixedFieldSpec relate a fieldname with its memory bank, offset and length. Alternatively TMVariableFieldSpec relates the fieldname and object ID (OID) defined in ISO/IEC 15962. The mapping of TMFixedFieldSpec onto a tag physical memory is an ALE out-of-scope.

As was explained in the above, the ALE only provide an ECspec or CCspec for the client application. The actual RFID reader control is the out of scope of the ALE.

2.2 ISO/IEC proposed model

This section shows the ISO/IEC proposed method for user-specific data processing in networked RFID system. Figure 2.2 shows the ISO/IEC reference model. In this section, ISO/IEC 15961[4]/15962[5] and ISO/IEC 24753WD[15] are introduced.

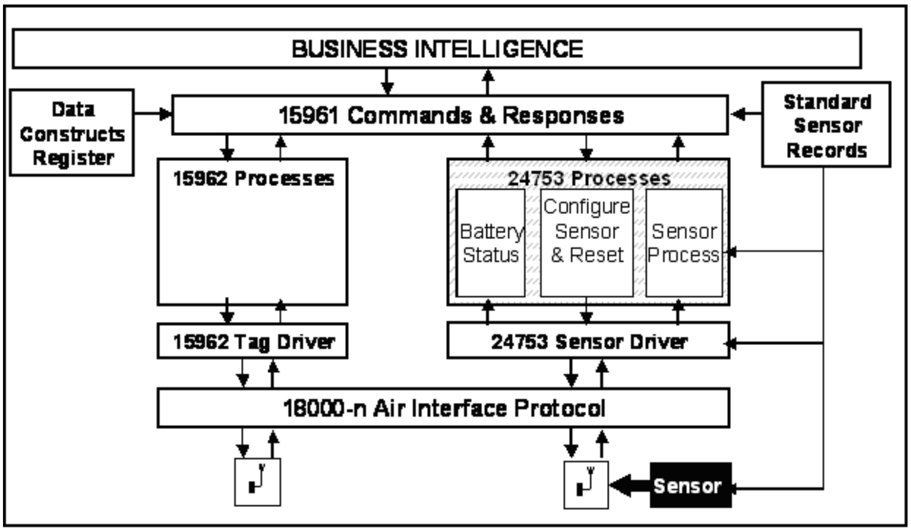

Figure 2.2: Basic application interface model in ISO/IEC model [4]

2.2.1 ISO/IEC 15961 and 15962

ISO/IEC 15961 [4] and 15962 [5] defines the application interfaces and the logical data format. The positions of the specification among other ISO/IEC specifications are illustrated in Figure 2.3.

ISO/IEC 15961 uses an object ID to specify a user-specific data. APIs are determined by using ASN.1 syntax. For example, the retrieval of single object ID is done through ReadSingleObjectModules. The ASN.1 syntax is defined as follows in Figure 2.4.

The syntax is usually encoded into a binary form (BER: Binary Encoding Rule) by means of an ASN.1 compiler or interpreter. The command is responded by the following ReadSingleObjectResponse in ASN.1 form (Figure 2.5).

In ISO/IEC 15961 and 15962 tag data is laid out on a tag logical memory. The layout can be either non-directory structure or the directory structure. The non-directory structure (Figure 2.6) stores a series of precursor, OID and object data including their length from the beginning of the logical memory.

The directory structure (Figure 2.7) is composed of directory area and data area. The data area is the same to non-directory structure. Preparing the

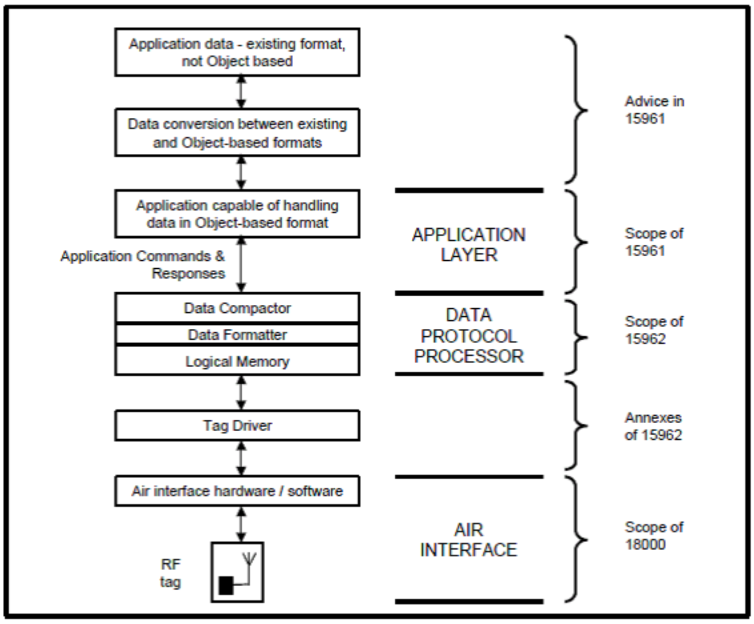

Figure 2.3: Schematic of protocol layers in ISO/IEC 15961 and 15962 [4]

directory provides an way to search for the directory first and then, retrieve the requested data itself. The directory entry specifies the lowest and highest byte address in the logical memory.

It is believed that the directory and non-directory structure in the logical memory map are directly mapped onto the tag physical memory, although it is the responsibility of the tag driver defined in ISO/IEC 15961 and 15962. If the RFID tag has other mapping such as ISO/IEC 24753 as explained in the next subsection, we can choose "self mapping" RF tag as the accessMethod parameter.

2.2.2 ISO/IEC 24753 WD

ISO/IEC 24753 Working Draft [15] is designed for the handling of sensor data which is installed to the RFID tag using various physical methods, but always using the air interface protocol such as ISO/IEC18000-6TypeC REV.1 [16] as a consistent means of communication between the RFID tag and the RFID reader. Figure 2.8 shows the logical interface model of this standard.

Each of the component parts of the model relevant to accessing sensors and management of batteries which is stored on the RFID tag is described below.

Sensor identifier

The sensor identifier enables communications to be established if more

```
ReadSingleObjectCommand
{iso(1) standard(0) rfid-data-protocol(15961)
 commandModules(126) readSingleObject(7)}
DEFINITIONS
EXPLICIT TAGS ::=
BEGIN

ReadSingle Object Command ::= SEQUENCE {
    tagId               OCTET STRING(SIZE(0..255)),
                        -- skip
    objectId            OBJECT IDENTIFIER,-- Full OID value
    checkDuplicate      BOOLEAN
                        -- If set to TRUE, the interrogator
                           shall check that there is
                        -- only one occurrence of the ObjectId
}

END
```

Figure 2.4: An ASN.1 syntax example of ReadSingleObjectCommand [4]

```
ReadSingleObjectResponse
{iso(1) standard(0) rfid-data-protocol(15961)
 responseModules(127) readSingleObject(7)}
DEFINITIONS
EXPLICIT TAGS ::=
BEGIN

ReadSingleObjectResponse ::= SEQUENCE {
    completionCode    INTEGER {
                         noError(0),
                         tagIdNotFound(8),
                         duplicateObject(10),
                         objectIdNotFound(13),
                         objectNotRead(15),
                         executionError(255)
                         },
    executionCode     INTEGER ,
           -- See Clause the specification and notes in this
           -- syntax for a full list of executionCodes
    object            OCTET STRING,
    compactParameter  INTEGER {
                         applicationDefined(0),
       -- The object was not originally encoded through the
       -- data compaction rules of 15962, and is as sent from
       --  the source application and might require additional
       --  processing by the receiving application.
                         utf8Data(2),
       -- Data has been externally transformed from a 16-bit
       -- coded character set to a UTF-8 string. The object
       -- needs to be processed through an UHF-8 decoder.
                         de-compactedData(15)
       -- The object was originally encoded through the data
       -- compaction rules of 15962 and de-compacted on this
       -- read operation and restored to its original format.
                         }(0..15),
    lockStatus           BOOLEAN
                         -- If TRUE, object is locked
    }
END
```

Figure 2.5: An ASN.1 syntax example of ReadSingleObjectResponse [4]

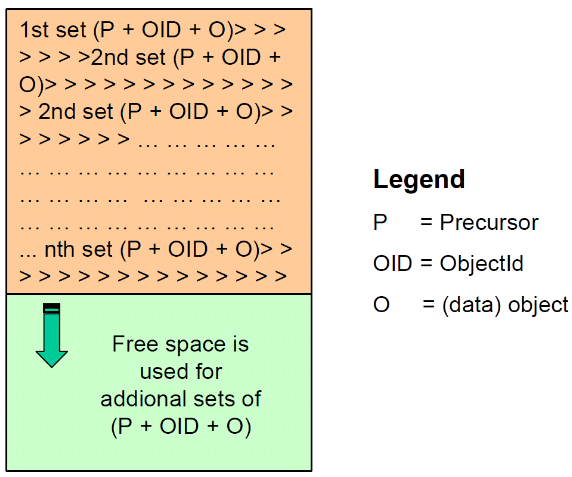

Figure 2.6: Logical Memory Schematic - Non-directory Structure [5]

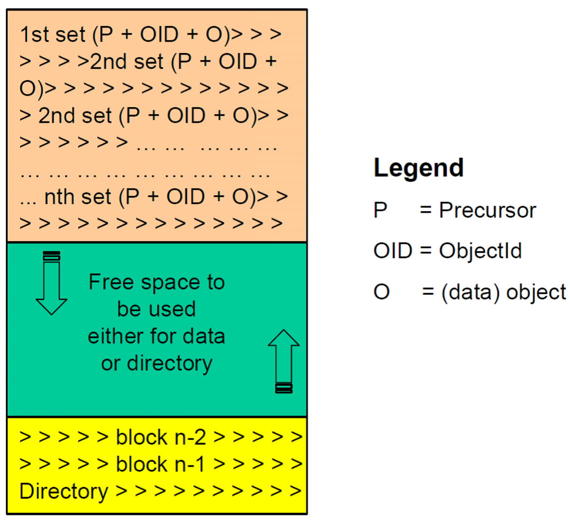

Figure 2.7: Logical Memory Schematic - Directory Structure [5]

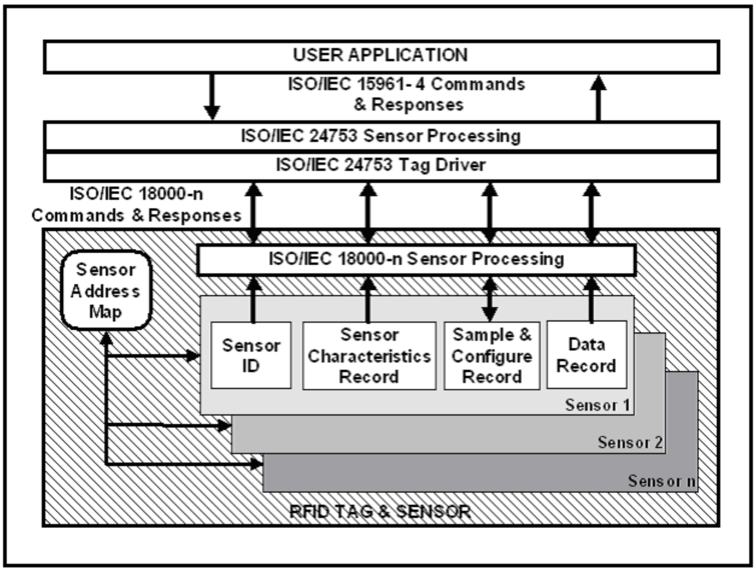

Figure 2.8: Sensor information model in ISO/IEC 24753WD

than one sensor can be incorporated on the RFID tag, or if sensor functions can be changed (e.g. through a port connection).

Sensor characteristics
The associated record defines the basic functional capabilities of the sensor in a reasonably concise manner. This international standard specifies the structure of the record, which is a read-only record (not rewritable), so that the sensor functions remain permanently identifiable.

Sampling and configuration
The associated sample and configure record allow the features, which the sensor is capable of monitoring, to be set (for example to particular threshold values). This record also defines the time-based sampling frequency, if applied. The sampling and configuration values allow the sensor to perform its functions independently of any further air interface communications and to begin to create its data record. To ensure that the sampling and configuration values remain set by an authorized user, the air interface protocols may impose additional security features so that only those with permission to change the configuration in the sampling may do so.

Data record
The data record encodes, in a binary format, the digital output from the sensor. The data-set can include a single entry, or a series of multiple entries, presented in a time interval data-set or event-triggered data-set.

A variety types of sensor information, which are generated by sensors installed on RFID tag, are stored on the some address space in the RFID tag memory. In ISO/IEC 24753WD, the schema information of such a RFID tag is described in the specific RFID tag memory hierarchically. Figure 2.8 shows such a structured tag memory which stores the schema information in the specific RFID tag memory area.

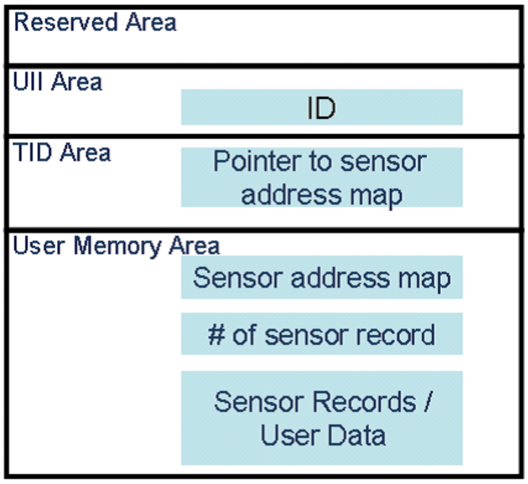

Figure 2.9: Structured tag memory in ISO/IEC 24753

The initial hook

First of all, since the encoding on an RFID tag is unknown until a line of communication is established between the RFID tag and the RFID reader, it is impossible to pre-determine whether the RFID tag supports a sensor and/or battery management. Therefore, each RFID tag shall provide a basic signal with which it can respond to the RFID reader to indicate support for these additional features. In ISO/IEC 24753WD, two options are discussed for pre-determining the type of an RFID tag as bellow.

Using DSFID
: The DSFID, as specified in ISO/IEC 15961-3, has a single bit value that signals functional extensibility. If this has the value 1, then it indicates that the RFID tag supports additional functions. The byte immediately following the DSFID then defines which functions are supported. With respect to this international standard, and the use of DSFID, the structure is defined in Table 2.1.

Using a memory map hook
: Some air interface protocols are capable of defining a particular memory

Table 2.1: Functional extensibility

	msb <————> lsb		
Description	RFU	Sensor(s)	Battery
number of bits	6	1	1

location, which can encode minimal information to indicate support for sensors and batteries. If this option is used, the location needs to be defined in the relevant air interface protocol and in the sensor driver.

Pointer to Sensor Address Map (PSAM)

Whether the DSFID plus functional extensibility method or memory map hook is used, the encoding that follows shall be to a common set of rules (Table 2.2). In the case of the DSFID method, the pointer to the Sensor Address Map (PSAM) shall immediately follow the functional extension byte. In the case of memory map hook, the PSAM shall be defined at a specific memory location in the air interface protocol and the sensor driver.

Table 2.2: Structure of the pointer to the Sensor Address Map (PSAM)

	RFU	MB	Address
Number of bits	6	2	24
Description	Reserved for future use	Memory bank selector	SAM start address (EBV-8)

The default value for the PSAM shall be 0 for no sensor. Battery assisted tags with one or more sensors shall have one PSAM with a value other than 0. For tags with a single memory, the value of the MB bits shall be 2, for tags with multiple memory banks, the value of the MB bits shall refer to the memory bank where the Sensor Address Map (SAM) is located.

Sensor Address Map (SAM)

The Sensor Address Map (SAM) contains the memory address and memory range of each sensor allowing port type access to sensor configuration records and sensor data through the use of air interface commands. The structure of the SAM shall be as follows: 2 bytes specifying the number of available sensors (NoS) followed by NoS SAM-Entries comprising 6 bytes each.

A single SAM-entry shall consist of the following structure, as shown in Table 2.3:

Table 2.3: Structure of SAM-Entries

	RFU	MB	Address	Range
Number of bits	6	2	24	16
Description	Reserved for future use	Memory bank selector	Sensor address (EBV-8)	Memory range (EBV-8)

Sensor memory record

The sensor memory record shall apply to a single sensor, regardless of the way how it is integrated on the RFID tag. If multiple sensors are present, then each shall have its own sensor memory record.

The basic structure of each sensor memory record is defined as bellow.

- The sensor identifier
- The sensor characteristics record
- The sampling and configuration record
- The data record

2.3 IEEE1451.4 Transducer Electronic Data Sheet (TEDS)

IEEE1451.4 specification provides the standard mechanism for accessing any type of transducer such as sensors or actuators regardless of their types, manufacturers and underlying information network.

In IEEE1451.4 specification, the normalized electronically format named TEDS (Transducer Electronic Data Sheet) is defined. By using the TEDS, plug and play capabilities to analog transducers among variety types of sensors and actuators can be realized. The way of deployment of the TEDS information can be divided into two ways as followed (Table 2.4).

The TEDS information can be divided into two sections. First one is named Basic TEDS which is comprising of the essential identification information, such as manufacturer ID, model number and so on. The other one is composed of

Table 2.4: The way of deployment of the TEDS information

types	description
TEDS (normal TEDS)	resides in embedded memory, typically a EEPROM, within the analog transducer, as defined in the IEEE 1451.4 standard.
Virtual TEDS	can exist as a separate file and downloadable from the Internet. By using the Virtual TEDS, legacy transducer and applications where the embedded memory or EEPROM is not available.

standard template information by each types of transducers such as calibration data, physical measurement range and so on. The table 2.5 shows an example of a TEDS information of an accelerometer.

ISO/IEC group who are interested in the incorporation of sensor in RFID has also discussed their next specification [15][16] of new data structures or air protocols which consistent with IEEE 1451 standardization.

2.4 Sensor integrated RFID

Integration of sensor and RFID architectures has been an active research field. Emery [17] proposed an example of the integration model of sensor and RFID. The sensor data is represented as a combination of physical quantity and the time with their units. The sensor data is transported as streams and processed with relations and operations. Introduction of 'Query' to represent a state as a combination of streams and relations and various data processing roles such as filtering, inference and differentiation are worth noting. The ELIMA project had been an EC-funded research endeavor that aimed to demonstrate the acquisition and management of product lifecycle data by collecting and exploiting data from products in different phases of their lifecycle including design, manufacture, use, maintenance and end-of-life product recovery. ELIMA partners built and tested a prototype Identification and Data Unit (IDU) to collect technical operational data from seven different sensors. Deng [18] proposed Sensor Embedded RFID (SE-RFID) which could be classified as hardware integration and a type of sensor tag. The sensor data is A/D converted and are read by RFID reader as user data. In the application referred to as HEMS (Real-Time Health Monitoring), additional integration model is stated as an extension of SE-RFID. Ranasinghe [19] overviews the EPCglobal network technology and its extension to incorporate sensors, fundamentally, at hardware integration level.

Table 2.5: An example of TEDS information

Basic TEDS	Manufacturer ID	43
	Model Number	7115
	Version Letter	B
	Serial Number	X001891
Standard and Extended TEDS (field will vary according to transducer type)	Calibration Date	Jan 29, 2000
	Sensitivity @ ref. condition (S ref)	1.0094E+03 mV/g
	Physical measurement range	$\pm$ 50 g
	Electrical output range	$\pm$ 5 V
	Reference frequency (f ref)	100.0 Hz
	Quality factor @ fref (Q)	300 E-3
	Temperature coefficient	-0.48 $\%/°C$
	Reference temperature (T ref)	23 $°C$
	Sensitivity direction (x, y, z)	x
User Area	Sensor Location	Strut 3A
	Calibration due date	April 15, 2002

He reveals the importance of introducing active tags and power-scavenging technology to realize the passive sensor tags. Zhang [20] presents the importance of RFID and wireless sensor network (WSN) technology convergence and proposes an integration of RFID reader into WSN by means of 'smart node'. Smart node has the capability of communicating with tags and with other smart nodes forming a WSN. The read data will be multi-hopped and feed into the information through sensor sink.

2.4.1 ISARI

Naruse proposed ISARI (Integration of Sensor and Actuator into RFID Infrastructure) model [21]. The main purpose of ISARI model is to operate RFID tag, sensor node, actuator node and sensor tag equally on RFID network model.

To realize the handling such a variety types of tag/node in the same way, he indicated the following issues.

Solve a tag/node information

- Number of sensor and actuator
- Memory address map of each sensor and actuator

Solve sensor/actuator information

- Manufacturer, Model, Version, Serial
- Sensing range, frequency
- Calibration data

Variety types of sensors/actuators can be accommodated in a same manner

To solve these issues, the following ideas are proposed in the middleware level (Figure 2.10).

- Describe to schema information and store at registry service
- Describe to spec sheet and store at registry service
- Prepare the protocol translator so that any types of tag/node can be communicate with the same application in the same manner

2.4.2 Virtual Tag Memory Service (VTMS)

Floerkemeier and Lampe proposed the Virtual Tag Memory Service (VTMS) in their research on the efficient accessing to RFID tag. The main idea of the VTMS is to preparing the virtual tag memory space in the network registry named virtual tag memory service (Figure 2.11). Using the VTMS, the following issues on RFID tag can be solved.

- limited tag memory size
- different memory organizations
- reduced write range
- stored the backup copy of RFID tag memory

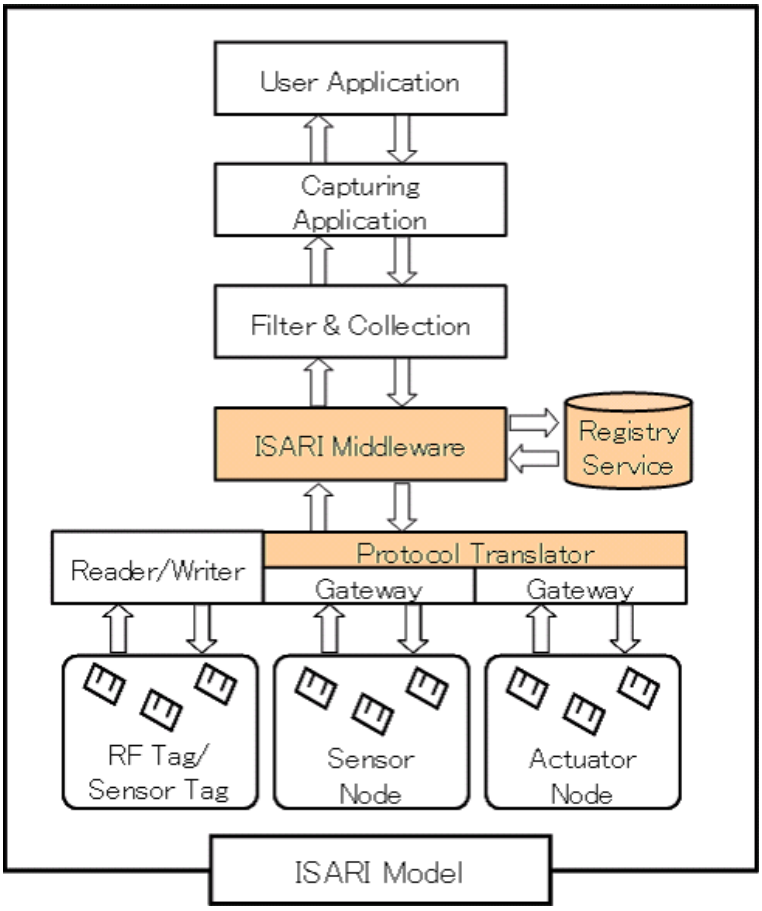

Figure 2.10: ISARI architecture

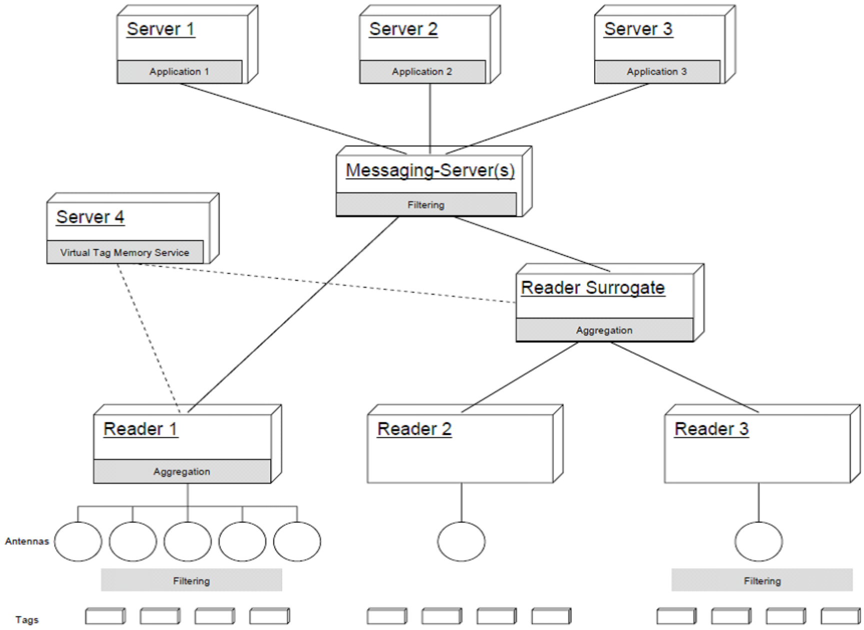

Figure 2.11: Deployment diagram containing Virtual Tag Memory Service (VTMS)

Chapter 3

Proposed Architecture

In this chapter, firstly the requirements for generalized handling of user-specific data collection and processing in networked RFID are introduced. Secondly, a networked RFID system architecture, which accommodates the requirements, is proposed.

3.1 Requirements

Accommodating to Generalized user-specific data

User-specific data may be sensor data or a certain type of record such as sales and repair history. Different from sensor data, which may be well structured as in [22], the structure of non-sensor data such as the sales and repair history cannot be described well structured. The whole or a part of data may also need to be encrypted or may need to be human readable or may need to be access controlled depending on the usage. As such, general user-specific data can be collected with the minimum set of the semantics in the data.

Resolving tag memory schema

An RFID reader/writer needs to resolve the tag memory schema, which depends on the types of user-specific data, and also the length of the memory block before collection. Moreover, the additional procedure to handle an user-specific data should not change the existing procedure to inventory ID tags.

Collection of dynamically changing data

User-specific data can be classified into static and dynamic types. In this paper, "static" user-specific data represents the data length and data types are consistent throughout operations. "Dynamic" user-specific data, on the other hand, has variable data length. The typical dynamic data is data logging.

Partial collection of user-specific data

The most rudimental method to collect user-specific data is to read all the tag data by every RFID reader/writer. This may cause significant reading performance degradation particularly when the target tag is equipped with a number of sensors or has a large memory, for example logging data. There may be a situation that the owner of the user-specific data demands to control which part of the data can be read depending on the RFID reader/writer.

Efficient retrieval of large amount of RFID tag data

As an RFID tag may store a variety types of user-specific data, the radio transaction between RFID tag and RFID reader/writer may be significantly degraded under a radio noise environment. The ordinary method to retrieve RFID tag is single data read by specifying the offset of necessary RFID tag memory address and the retrieval size. When the block size, the unit memory size of which RFID reader/writer should retrieve tag data from an RFID tag is large, the capability of packet error and as a result the reading efficiency is degraded.

3.2 Architecture overview

The proposed networked RFID system is shown in Figure 3.1.

There are two essential entities in the proposal. One is the schema resolver which can resolve each tag memory structure and instructs the RFID reader/writer where the necessary data is stored after the RFID tag inventory. Another one is the reader/writer controller which can enforce the RFID reader/writer to collect any types of RFID tag data efficiently even if the data may be huge data length. In the following sections, we will show the proposed architecture.

3.2.1 Application

The application role in the proposal overview (Figure 3.1) corresponds to the capturing application in the EPCglobal network architecture (Figure 1.2). The application collects the read event and schema information of each RFID tags, and then defines the action what to do upon the EPC inventory.

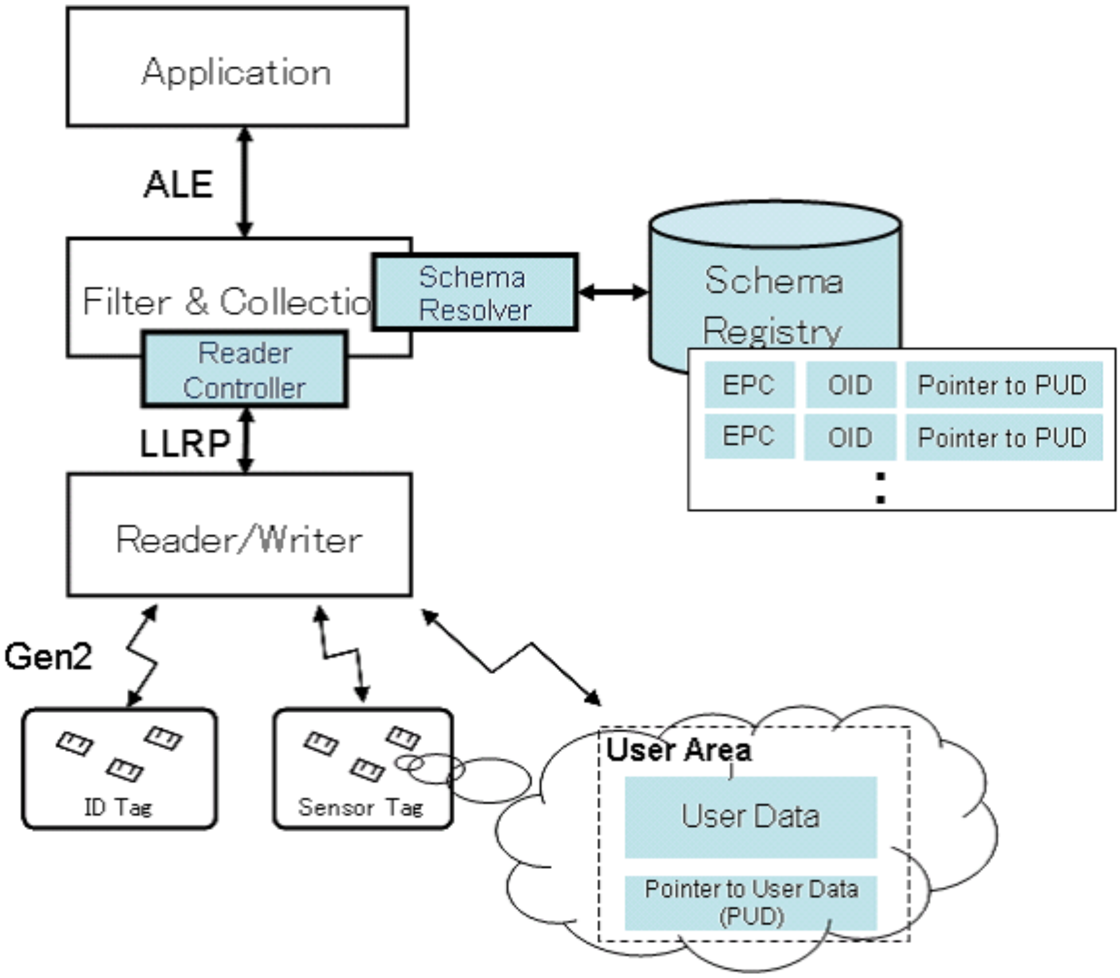

Figure 3.1: Principal entities of the proposal

3.2.2 Middleware

The middleware role in the proposal (Figure 3.1) contains the main ideas in this thesis. Each details of those are introduced in the following chapters.

The middleware in this proposal overview corresponds to the Filter & Collection role in the EPCglobal network architecture(Figure 1.2).

Schema Resolver

Upon the results of the RFID tag inventories, the schema resolver sends an tag EPCs inquiry into the schema registry and receives the schema information of each RFID tag entities. The schema information contains the information in Table 3.1

Table 3.1: Schema sheet profile

items	definitions
EPC	EPC is a globally unique identifier which is contained in each RFID tag.
OID	OID denotes an object identifier. OID is introduced in the ISO15962 specification.
Point to PUD (PPUD)	PPUD denotes the pointer to a specific physical address in the RFID tag memory.

Reader Controller

The reader controller instructs the reader/writer on the offset address of the pointer information which indicates the user-specific data record based on the schema sheet information. In the proposal model, the reader/writer should collect not only EPC information but also the user-specific data which are stored in an RFID tag memory. Since the reader/writer does not know where the necessary user-specific data are stored in advance, the reader controller, which is on the middleware in the proposal model, should instructs reader/writer to read the proper memory address in each RFID tag.

In addition, when it comes to read a user-specific data from an RFID tag, the length of the data can become large. In order to reduce the time to retrieve the user-specific data from RFID tag memory, the reader controller optimally adjusts the block size in accordance with the packet error rate.

3.2.3 RFID Reader/Writer

The role of an RFID reader/writer is usually only to collect ID information such as EPC from each RFID tag memory. However, in the proposal model, a role of an RFID reader will be increased. After the RFID tag inventory, the RFID reader/writer should collect the user-specific data from some specific RFID tag memory. As I mentioned in the previous section, the reader must resolve the schema information of the target RFID tag which is to be retrieved the user-specific data in the RFID tag memory. In EPCglobal standard, the reader protocol, which is the interface between RFID reader/writer and middleware, are to define the additional protocol for user-specific data collection in the next standard revision. We are going to take the standard protocol into the proposed architecture.

3.2.4 RFID tags

RFID tag comes to contain more and more additional data such as user-specific data. As I mentioned in Chapter 1, some RFID tags contains a/some sensor device(s) and/or high capacity memory spaces. Those tags may contains not only EPC data but also a variety types of user-specific data such as sensor data or non-sensor data.

In the proposed method, the high performance RFID tag contains the user-specific data, such as sensor data or non-sensor data in the tag memory. For example, Gen2 tag contains the user-specific data in User data area in tag memory. In addition, such an RFID tag, which contains additional data, also prepare a part of an index information, Pointer to User Data (PUD) in its tag memory spaces (Table 3.2). PUD indicates the offset address where the user-specific data is stored.

Table 3.2: Schema sheet profile in RFID tag

items	definitions
Pointer to User Data (PUD)	PUD is stored in User area in Gen2 tag.

3.3 Qualitative comparison

Against the requirements compiled in Section 3.1, the qualitative comparison of the existing architecture (explained in Chapter 2) and the proposed architecture are given in Table 3.3.

Table 3.3: Qualitative comparison

Requirements	ISO15961, 15962	ISO24753, ISO18000C REV.1	ISARI	Virtual tag memory	Proposal
General User-specific data can be accommodated	◯	Is tailored to sensor data. Sensor identifier cannot be directory applied to non-sensor data	Is tailored to sensor data. The data format of the data structure is similar to TEDS sheet.	Out of focus but can be accommodated	◯
Resolving tag memory schema	Reader/writer needs to retrieve at least all of the directory in directory structure	◯	◯	Out of focus but can be accommodated	◯
Collection of dynamically changing data	ObjectID which represents the pointer to Object need to be introduced	◯	Main focus is static data	Out of focus	◯
Partial collection of user-specific data	Out of focus (could be taken cared by Tag Driver)	Out of focus	◯	Out of focus	◯
Efficient retrieval of large tag data	Out of focus	Out of focus	Out of focus	Out of focus	◯

Chapter 4

Schema Resolver

The schema resolver helps RFID reader/writer to resolve the schema sheet, which is described the RFID tag memory address mapping information. The schema resolver satisfies the requirements mentioned in the previous Chapter 3, accommodating general user-specific data and resolving RFID tag memory schema.

4.1 Transaction sequence in the proposal

Figure 4.1 shows the typical transaction when the user-specific data is collected in the proposal. The RFID reader/writer collects the user-specific data in the following procedure (Figure 4.1) when an associated application sends a inquiries to the middleware for any types of user-specific data handling.

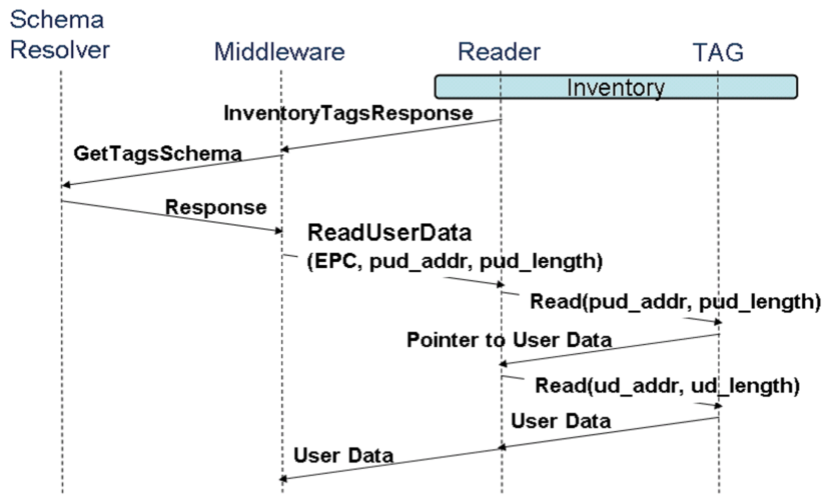

Figure 4.1: User-specific data collection procedure

1. The RFID reader/writer collects globally unique IDs such as EPC in RFID tag (UII) by tag inventory.
2. Upon the receipt of the unique ID, the RFID reader/writer replies with InventoryTagResponse to the middleware.
3. The middleware generates GetTagsSchema inquiry to the Schema Resolver with requested OID from application.
4. The Schema Resolver examines the unique ID in the read event and instructs the middleware of the range of the tag memory which needs to be collected.
5. The RFID reader/writer collects Pointer to User-specific Data (PUD) and determines the memory of the actual user-specific data.
6. The RFID reader/writer collects the body of user-specific data and delivers them to the associate applications.

For the purpose of comparison, the user-specific data collection in ISO24753WD [15] and ISO/IEC18000C Rev.1 [16] can be summarized as follows (Figure 4.2).

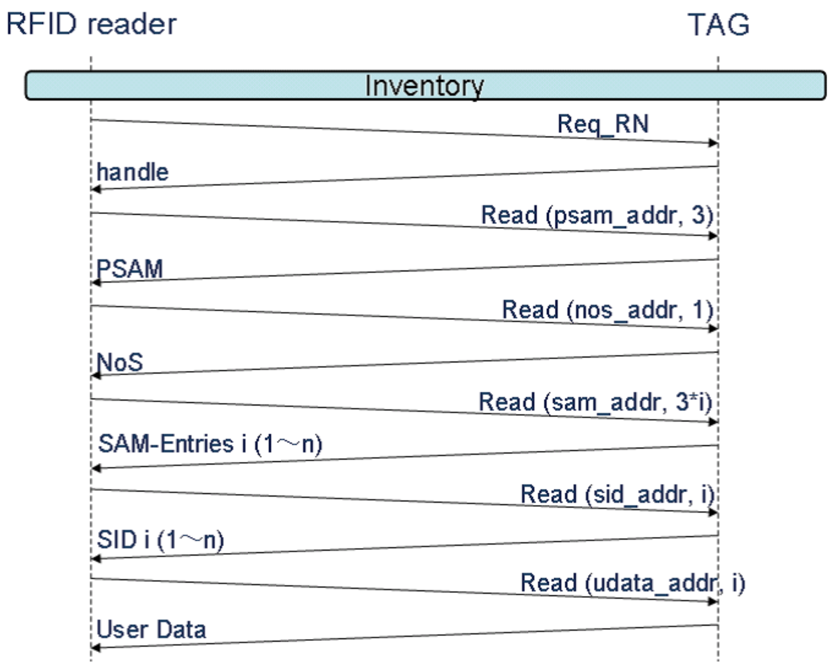

Figure 4.2: Transaction for resolving memory schema on tag in ISO model

1. The RFID reader/writer collects globally unique ID in RFID tag by tag inventory and identifies the sensor data capability either by user memory indicator (UMI) or XPC (eXtended PC bit) or XI (eXtended PC bit Indicator).
2. The RFID reader/writer accesses a particular bit of area in RFID tag and collects meta-data named PSAM (Pointer to Sensor Address Map) by next tag inventory.
3. Upon the receipt of PSAM, the RFID reader/writer collects the NoS (Number of Sensors) that indicates how many sensor devices are attached on the RFID tag. c
4. The RFID reader/writer collects the SAM-Entries that contains sensor record addresses.
5. From NoS and SAM-Entries information, the RFID reader/writer collects SID (Sensor Identifier) and then judge which sensor record needs to be acquired by an associated application.
6. The RFID reader/writer collects the body of user-specific data and delivers them to the application.

It should be stated that, in the ISO method in Figure 4.2, the RFID reader/writer needs to collect all the tag data from the RFID tag unless the RFID reader/writer has the knowledge on which sensor data needs to be collected. Similarly in a write operation without an inquiry to the schema resolver, the RFID reader/writer only can write the predefined pattern of data to the RFID tag or even cannot determine if it is requested to write a data or to collect a data. In this regard, the proposed method (Figure 4.1) and the existing method (Figure 4.2) can be complementary. If we have a network connection, the RFID reader/writer may place an inquiry depending on the UIM or XPC to the schema resolver to find any updated request on the user-specific data collection and writing. If there is no updated instruction, the RFID reader/writer collect or write in the predefined pattern, which may collect all the data or no data.

4.2 Data structure design

In this section, the proposed data format design of both in the schema registry and RFID tag memory are introduced. As mentioned above, in order to realize generalized use-specific data handling in networked RFID system, it will become very big issue how to resolve the tag memory schema information upon the RFID tag inventory.

The concept of my proposal is similar to that of the ISO24753 WD [15], In both way, the data format of the schema information are represented hierarchically. In ISO24753 WD, the whole schema information is stored in the RFID tag memory itself hierarchically2.9. In the proposed method, on the other hand, a part of the schema information is stored in the repository named Schema Registry, which is introduced as followed, in the network, and the rest is stored in the RFID tag memory itself. We will introduce the details of the structure of the data format in the followings.

4.2.1 Schema registry

Schema registry stores the following information by each RFID tag.

- EPC
- OID
- PPUD (Pointer to PUD)

Pointer to PUD

PPUD is similar to the PSAM in ISO/IEC 24753WD[15]. The detail of the data structure is as followed. PPUD is referred from UMI or XPC or XI. In the proposal method, since the PPUD is stored in the schema registry, the RFID reader/writer does not have to access to the RFID tag memory. This point is the different from the ISO/IEC 24753WD[15].

Table 4.1: Structure of the Pointer to Pointer to User Data (PPUD)

	RFU	MB	Address
# of bits	6	2	24
Description	Reserved for Future Use	Memory Bank selector	PUD start address

RFU denotes the Reserved for Future Use and the RFU is defined in ISO24753 [15]. MB denotes the Memory Bank selector. ISO24753 basically has been designed for ISO18000-6 Type C Rev.1[16]. In other words, ISO24753 is designed for Gen2[23] tag. Since there are 4 memory banks in Gen2 tag memory spaces, MB has been prepared for 2 bits memory spaces. Address denotes the physical start address of PUD.

4.2.2 RFID tag memory

RFID tag has an unique identifier such as EPC, the pointer to user-specific data (PUD) and user-specific data record. Figure 4.3 shows an example, based on an Gen2 tag, with a PUD for two types of user-specific data is stored in the User Memory.

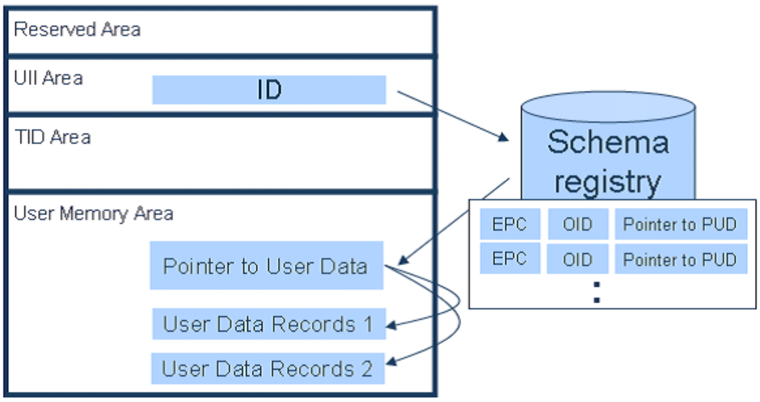

Figure 4.3: An exemplary of data structure in the proposal model

Pointer to User-specific Data (PUD)

A PUD entry shall consist of the following structure, as shown in Table 4.2.

- 6 bits reserved for future use (RFU)
- 2 bits identifying the memory bank (MB)
- a 24bit EBV-8 specifying the memory address inside the denoted memory bank
- a 16bit EBV-8 specifying the memory range occupied by the sensor

Table 4.2: Structure of Pointer to User-specific Data (PUD)

	RFU	MB	Address	Range
# of bits	6	2	24	16
Description	Reserved for Future Use	Memory Bank selector	User Data start address	Memory Range

User Data record

User Data record shall consist of the following structure.

- Object ID (OID)
- User Data record

4.3 Registry lookup overhead estimation

Since the proposed schema resolving method is a combination of a structured memory map in an RFID tag and the schema resolver, it is important to quantitatively evaluate the turn-around time for placing an inquiry to the schema resolver, which is referred to as the registry lookup overhead in this paper. We have already measured the registry lookup time in [21] in accordance with the number of user-specific data. But we don't have a reference overhead to compare with since we don't have an RFID reader/writer that can be instructed to collect user-specific data in accordance with the procedure in Figure 3. For this, we measured the actual user-specific data collection time using a commercial RFID reader/writer and a tag assuming we have already resolved the memory schema before data collection. The lookup overhead involving both the processing and transmission time between the RFID reader/writer and the tag was analyzed in detail and computationally re-assembled to estimate the collection time using the procedure in Figure 3. It should be noted that the collection of a body of user-specific data was out of the scope of this evaluation since it is the same irrespective of how the memory map is resolved.

4.3.1 Experimental setup description

The experimental setup to measure the lookup overhead is shown in Figure 4.9 and 4.5. The commercial RFID reader/writer (Figure 4.6 and 4.7) is connected to a battery-assisted passive (BAP (Figure 4.8)) tag [24] by a cable, and user-specific data is collected using an ISO/IEC 18000-6 Type C [25] command. The overhead was measured by monitoring the demodulated signal and the backscattering signal at the tag using an oscilloscope (OSC).

The configuration of the communication between the RFID reader/writer and the RFID tag is summarized in Table 4.3.

4.3.2 Estimation result

With the above experiment environment, we estimated the overhead time for the protocol shown in Figure 4.10. Since we do not have the actual RFID

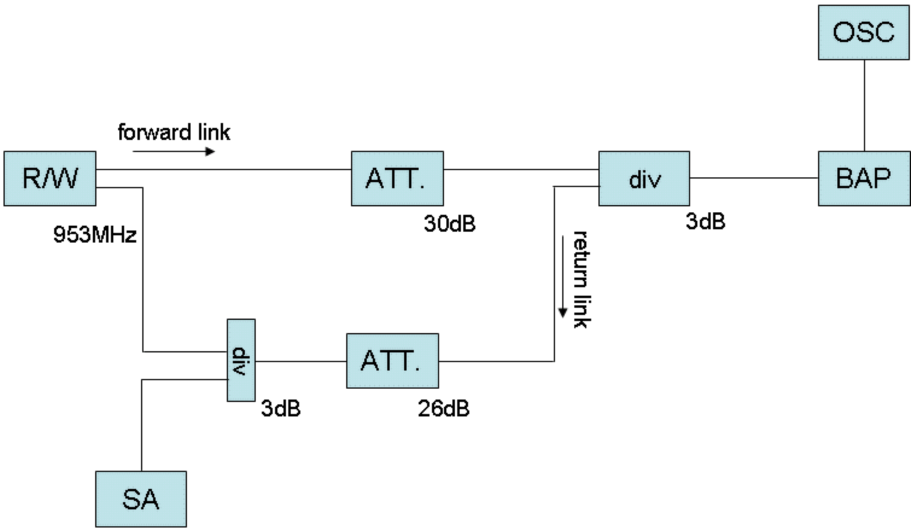

Figure 4.4: Experimental set up for overhead measurement

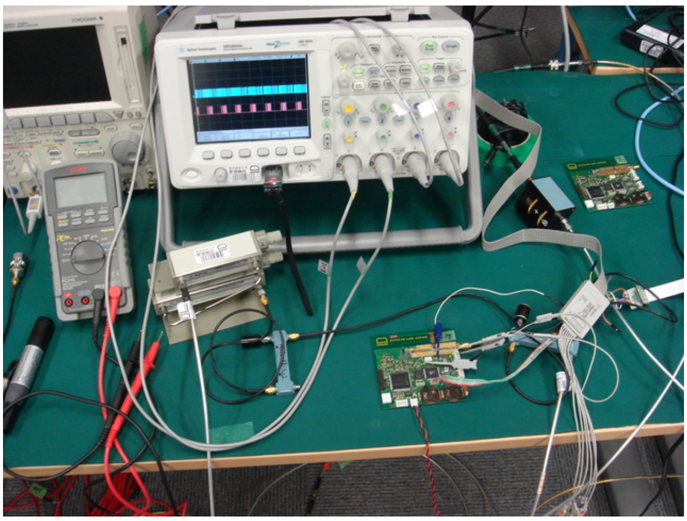

Figure 4.5: An example of experimental scenery

Figure 4.6: SkyeTek M9 Reader/Writer device

Figure 4.7: Panasonic KU-U1600

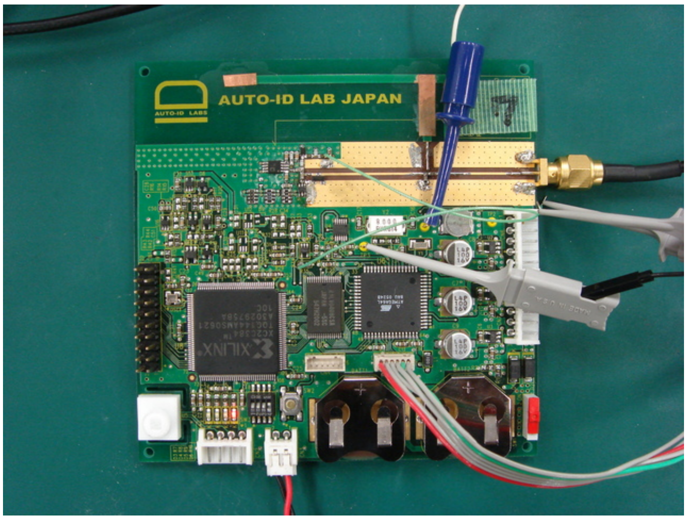

Figure 4.8: Multipurpose battery assisted passive tag

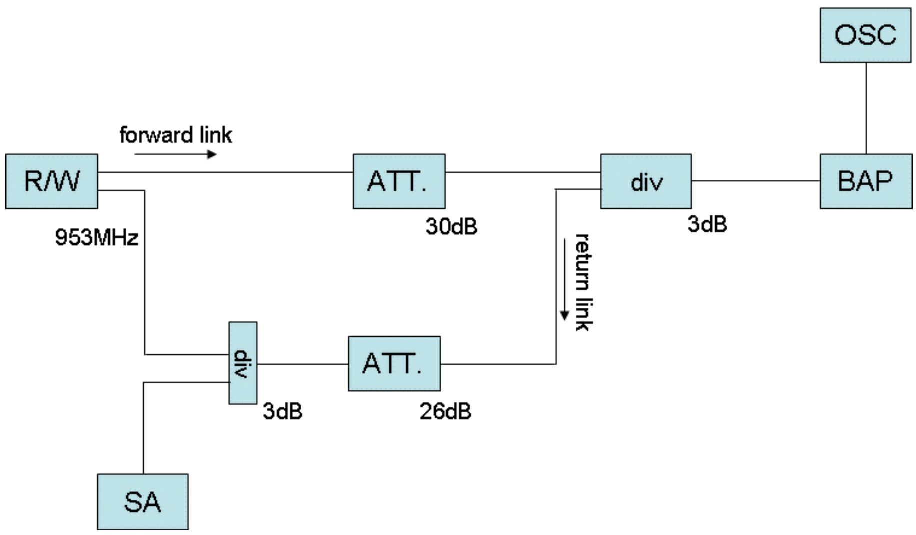

Figure 4.9: Experimental set up for overhead measurement

Table 4.3: Experimental configuration for memory schema resolving overhead measurement

item	value
Air protocol	ISO18000-6 Type C
Air communication speed	Forward link: 40kbps Return link: 40kbps
Air propagation loss	Forward link: 30dB Return link: 32dB
Center frequency	953MHz
Band width	200KHz
R/W output PWR	30dBm
R/W device	SkyeTek M9 (Figure 4.6)

reader/writer which we can control the raw protocol sequence directly, we examined the required time for which a certain block size command is transmitted and then estimated the required time for the protocol sequence as follows (Figure 4.10).

The numbers of SAM-Entries and SIDs depend on the number of sensor devices installed in a tag. If a certain tag contains numerous sensor devices, the length of the data that should be read by the RFID reader/writer becomes longer. The data length of one SAM-Entry and one SID are 3 words and 1 word, respectively, for a total of 4 words (1 word = 16 bits).

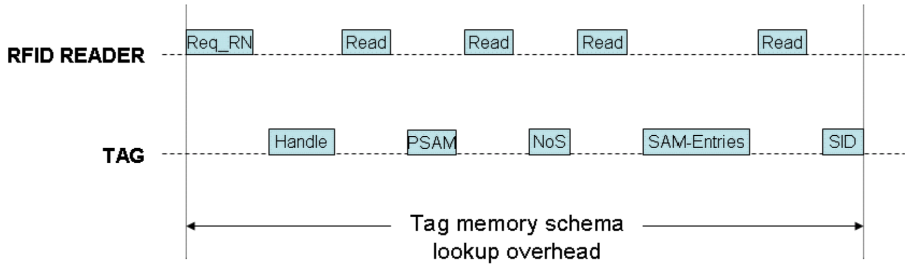

Figure 4.10: Air protocol sequence

Figure 4.11 shows the result of tag memory schema lookup overhead in the two methods. The legends "proposed method" and "existing method" in Figure 4.11 represent the procedures in proposed and existing method, respectively. The horizontal axis denotes the number of sensor devices installed on one tag and the vertical axis is the time (msec) required to collect that amount of user-specific data. The lookup overhead of the proposal was found to be around several 10 msec, which is almost equivalent to that of the existing method and even shorter when the number of sensor devices installed on a tag is increased. This is because we set up the registry to be network-wise close to the RFID reader/writer and the communication protocol is relatively slow, resulting in longer time for plural sensor data collection.

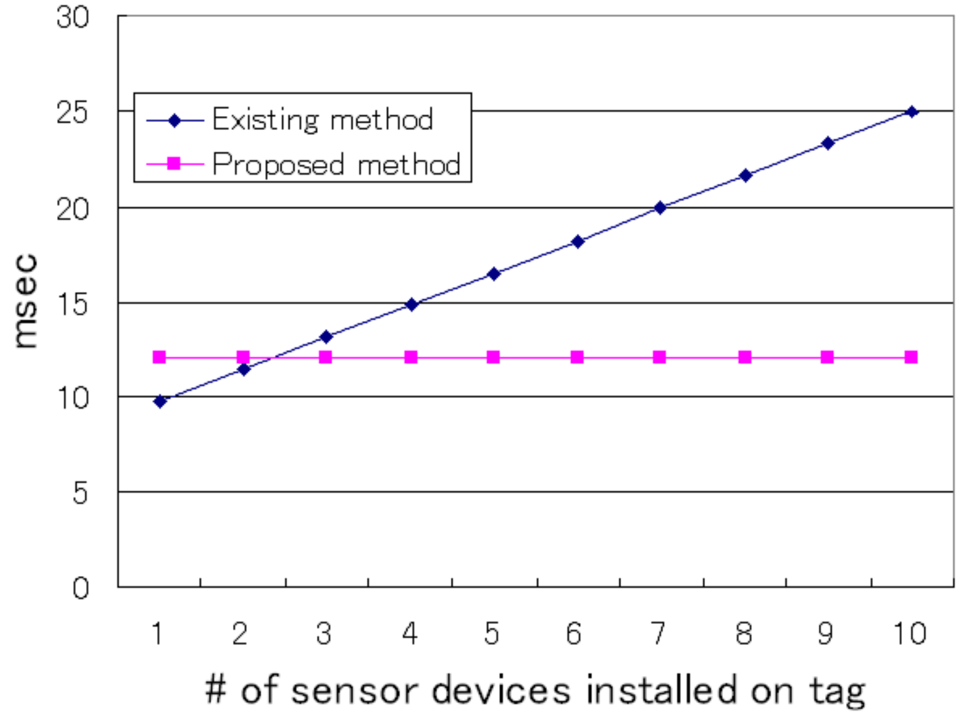

Figure 4.11: Tag memory schema look up overhead

Chapter 5

Reader Controller

In this chapter, we will discuss the reader controller. The reader controller enforces the RFID reader to collect any types of RFID tag data efficiently even if the RFID tag data contains a huge data length. To satisfy the above requirements, the reader controller receives and analyze the inquiry for RFID tag data read or write operation from an associated application, and then instructs the RFID reader of the segmented range, which is adjusted into the optimal block size to be read considering the radio noise, to collect the user-specific data efficiently. Considering the outer radio noisy affection, the reader controller adjusts the block size which the RFID reader may read in one operation.

5.1 Pilot experiment

Before reviewing the adaptive block size control algorithm for large RFID tag data retrieval, we measured RFID tag data retrieval time under a noisy environment in existing method using a commercial UHF RFID reader (Figure 5.2). The experimental setup is shown in Figure 5.1. A commercial reader (R/W) is connected to a battery assisted passive (BAP) tag [24] by a cable and an user-specific data is collected using ISO/IEC 18000-6 Type C [25] Command. The user-specific data collection time is measured by monitoring the demodulated signal and the backscattering signal at the BAP using an oscilloscope (OSC). Attenuators (ATT) simulates the radio communication loss. Moreover, to emulate an actual packet error caused by noise and interference, a signal generator (SG) which generates the random noise at a certain noise level is connected to the return link (From BAP to R/W link).

The configuration of the experiment is summarized in Table 5.1.

The Figure 5.3 shows a screen shot of the both signals, forward link protocol sequence and return link protocol sequence with the oscilloscope when user-specific data (30 words) is retrieved in above experimental.

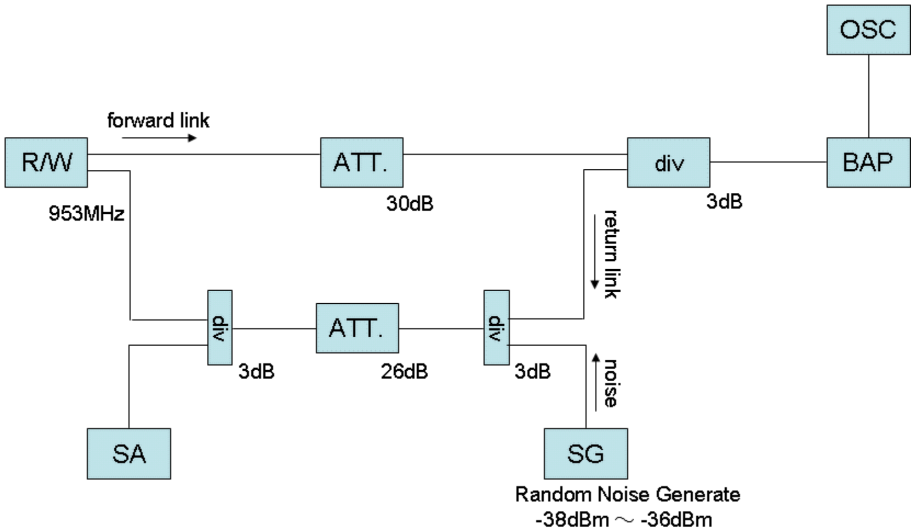

Figure 5.1: Experimental setup overview

Figure 5.2: Panasonic KU-U1600

Table 5.1: Experimental configuration for large tag data collection time measurement

item	value
Air protocol	ISO18000-6 Type C
Air communication speed	Forward link: 40kbps Return link: 40kbps
Air propagation loss	Forward link: 30dB Return link: 32dB
White noise level	Return link: 1e-6 ∼ 1e-4
Frequency	953MHz
Band width	200KHz
R/W output PWR	30dBm
R/W device	Panasonic UHF R/W

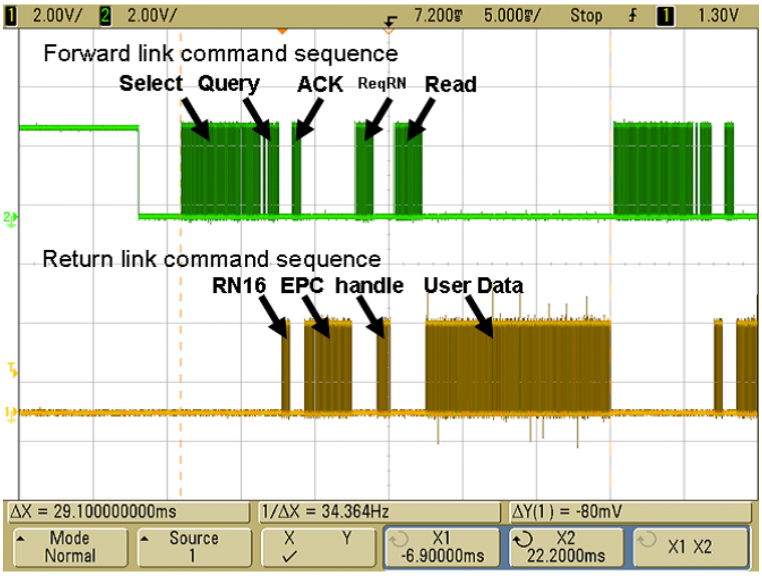

Figure 5.3: Screen shot of Oscilloscope

With the existing method, the RFID reader tries to collect the RFID tag data at one time even if the RFID tag data may be huge length. In addition, there is only CRC (16 bits) but no error-correcting code in the RFID air protocol Gen2 [23]. Hence, the retrieval time of large RFID tag data comes to be longer because of outer noisy environment. Following factors causes by the packet error occurred.

- Fail of the transmitting of the packet itself
- Success the packet transmitting, but retransmitted many same packet (Figure 5.4 & 5.5)

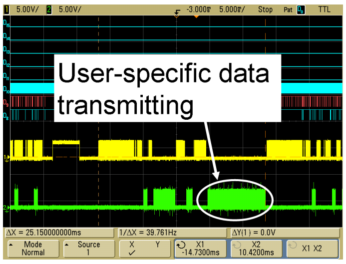

Figure 5.4: An example of success signal of an user-specific data retrieval

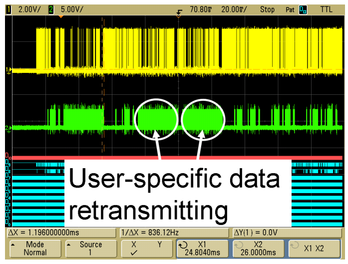

Figure 5.5: An example of retransmitting signal for an user-specific data retrieval

The Figure 5.6 shows the experimental result of large RFID tag data retrieval in certain a radio noise level environment. The horizontal-axis denotes the number (word) of user-specific data to be read and the vertical-axis denotes the elapsed time to collect the requested large amount of user-specific data.

It is shown that it took a lot of time to collect the large amount of user-specific data in RFID tag and noisy environment resulting in the delay of collection time of user-specific data. The longer the size of user-specific data becomes, the longer the required time to collect the tag data takes. From the result, it should be stated that it is necessary to retrieve the large amount of user-specific data with the retransmission of segmented block size and reduce the time to collect the necessary tag data.

5.2 Evaluation

To reveal more about the relationship between the outer noise level and the optimized block size to throughput can be faster, we evaluated mathematically and experimentally the large tag data retrieval time with a certain fixed size of the segmented block.

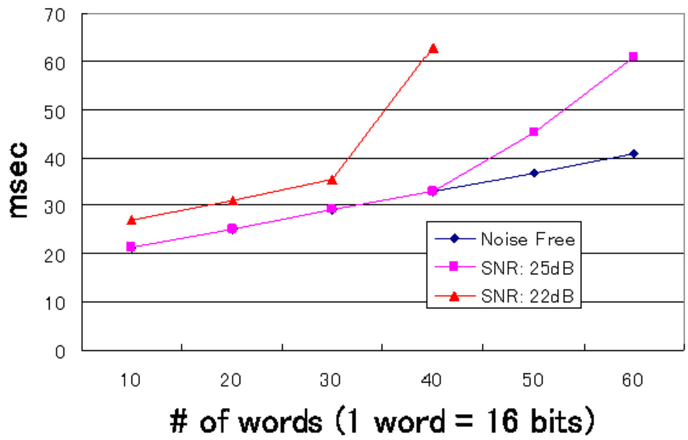

Figure 5.6: Large tag data retrieval time in noisy environment

5.2.1 Mathematical evaluation

Figure 5.7 shows the assumed protocol sequences when the large tag data is transferred with segmented block units in sequence. L denotes the length of the target tag data. l denotes the length of the segmented block unit. N denotes the number of times for retransmissions. As shown in Figure 5.7, in my assumption, each segmented block has to contain a certain length (not so short) of overhead in the front of actual necessary tag data. The reason is that each trial has to be started from the *Select* command.

The following formula 5.1 represents the Packet Error Rate (PER) under a certain Bit Error Rate (BER). l can be represented as $l = f + g$. f denotes the length of the overhead of a segmented block unit. g denotes the length of the length of necessity tag data.

$$PER = 1 - (1 - BER)^{f+g} \tag{5.1}$$

In case the packet error has occurred and the retransmission of the same block is transmitted N times, the Packet Error Rate (PER_N) can be represented as follows.

$$PER_N = \left(1 - (1 - BER)^{f+g}\right)^N \tag{5.2}$$

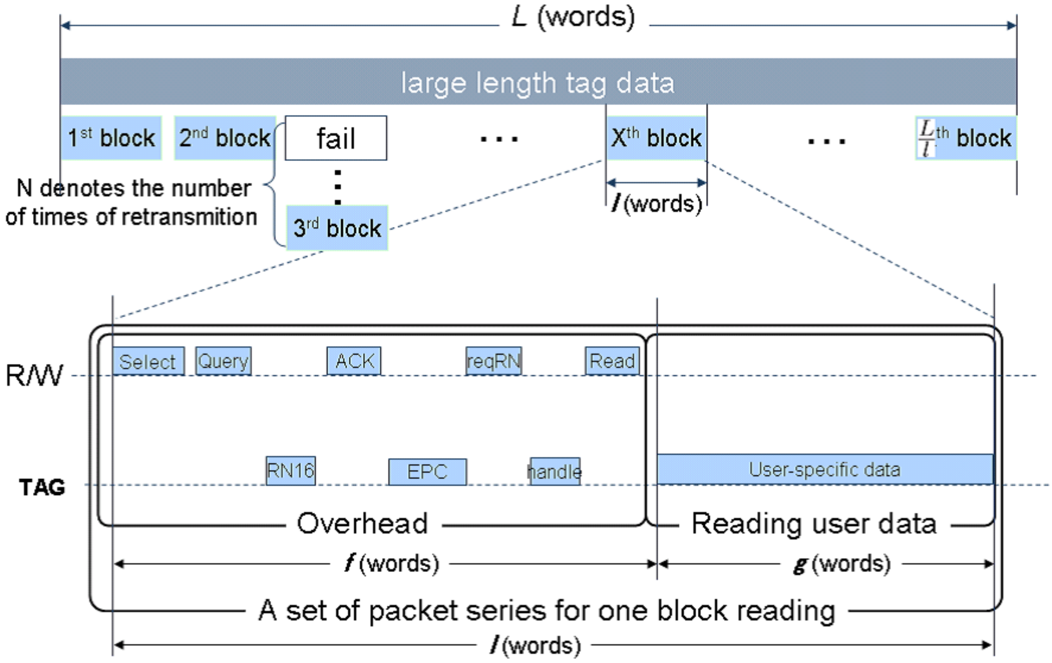

Figure 5.7: The assumed protocol sequences for the segmented block reading

As we have been acknowledged the threshold for the success of the packet transmission, the required Packet Error Rate($reqPER$) can be represented as follows.

$$PER_N \leq reqPER \tag{5.3}$$

From these formula, we can calculate the throughput ($TP = \frac{1}{N}$) as follows.

$$TP = \frac{\log\left(1-(1-BER)^{f+g}\right)}{\log\left(reqPER\right)} \tag{5.4}$$

Here the required time (RT) to be read whole the necessary tag data can be calculated from the total number of the required block units whether the block unit is transmitted successfully or failed. Therefore, RT is represented as follows.

$$RT = N * \frac{L}{l} * t = \frac{1}{TP} * \frac{L}{f+g} * t \tag{5.5}$$

t denotes the required time to be read the fundamental block unit (*1 word*). RT is represented as follows.

$$RT = \frac{\log(reqPER)}{\log\left(1-(1-BER)^{f+g}\right)} * \frac{L}{f+g} * t \tag{5.6}$$

At first, we calculated the RT with the following parameters (Table 5.2).

Table 5.2: Simulation parameters

parameters	values
Tag data length to be read (L)	60 (words)
ReqPER	1.00E-02
Bit rate	40 (kbps)
Overhead	15.06 (msec)

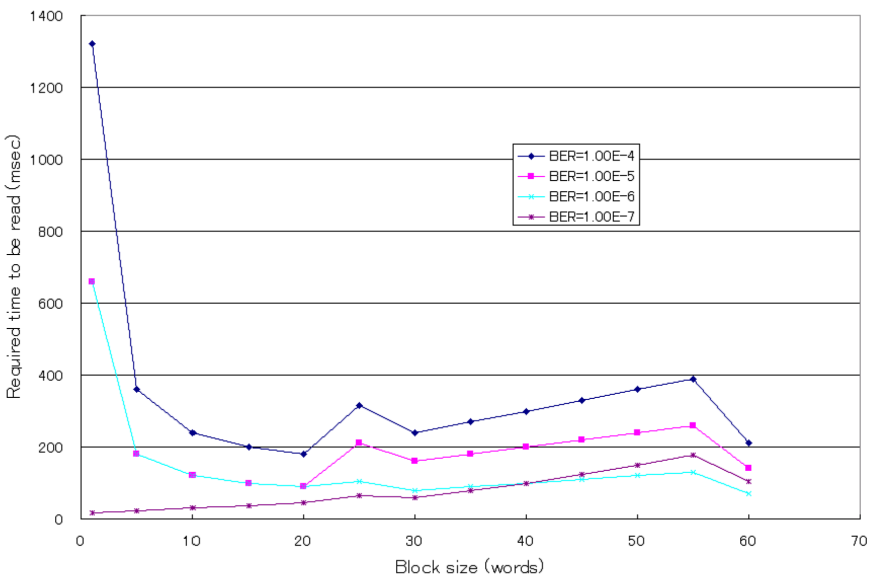

Figure 5.8: The tag data retrieval time with a segmented blocks(simulation)

The simulation result is shown as follows (Figure 5.8). The horizontal-axis means the block size (word order) and the vertical-axis means the required time to be read (msec order). The different parameter among each graph is Bit Error Rate ($BER = 1.00E-4$, $1.00E-5$, $1.00E-6$ and $1.00E-7$, respectively).

The result of the calculation revealed the following issues.

- Regardless of the BER, each graph has same turning value. In this case, the turning values turn up at 20 words, 25 words and 30 words respectively.
- In case the BER is small level($BER = 1.00E - 7$), the retrieval time is faster when the segmented block size is smaller. However, the BER level is equal level with the actual environment, the retrieval time is faster when the block size is about 20 words.
- Although the retrieval time in the existing one time reading method (in this case 60words retrieval) is not so bad.

5.2.2 Experimental evaluation

In a continuing series, we experimented the partition retrieval time of large tag data with a certain block size.

The experimental setup is as follows (Figure 5.9).

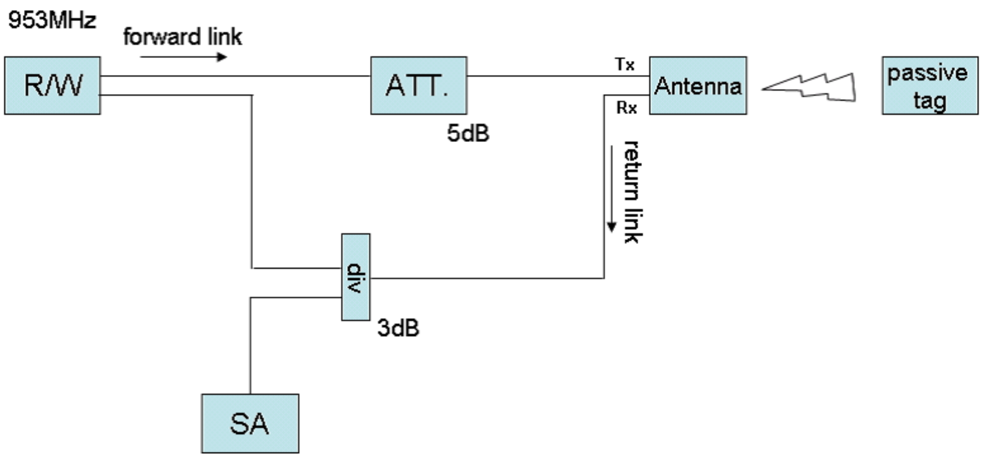

Figure 5.9: Experimental setup overview

In this experiment, we utilized a commercial RFID reader/writer (Figure 5.10) and a high capacity passive tag. The RFID reader/writer communicates with the passive tag with the radio frequency (953MHz). To emulate the radio noise environment, a signal generator (SG) which generates the pseudo random noise at a certain level was prepared and the Rx signal was monitored by a Spectrum Analyzer (SA)(Figure 5.11).

The Figure 5.12 shows the result of the partition retrieval time with fixed block size in the experiment. The horizontal-axis means the block size (word) and the vertical-axis means the required time to be read (msec). The result of the experiment revealed the following issues.

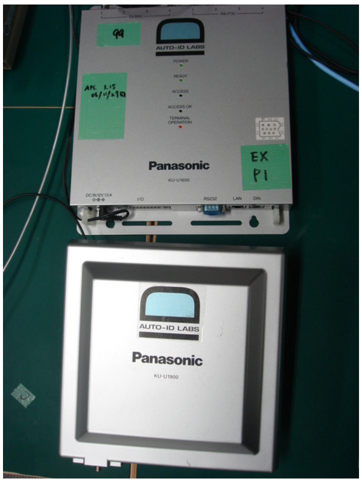

Figure 5.10: Panasonic KU-U1601

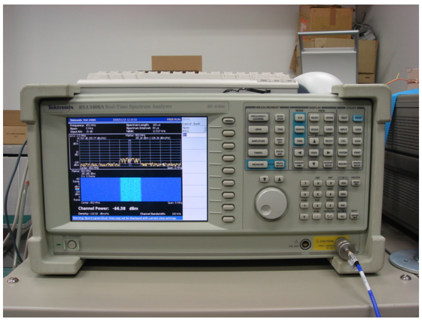

Figure 5.11: Spectrum analyzer (SA)

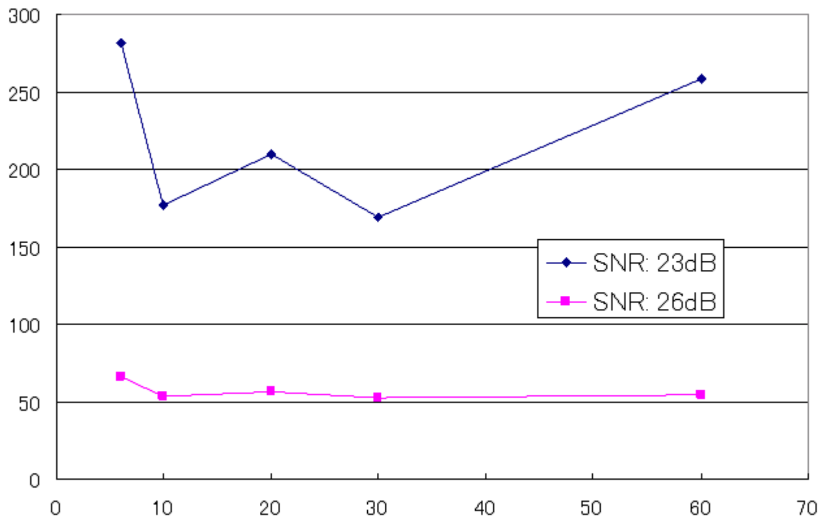

Figure 5.12: The tag data retrieval time with segmented blocks(experiment)

- In case the block size is small, in other words, when the number of the times of retransmission, the required time to be read is increased.
- Similar with the simulation result in the previous section, there are some turning values where the block size is optimized. (in this case, 10 words and 30 words)

The feature of the graph is similar to the result of the simulation (Figure 5.8).

5.2.3 Discussion

In order to improve the throughput of the retrieval of user-specific data from RFID tag, we have proposed the adaptive block size optimizing mechanism. While the rate of the packet error may be reduced by dividing a large tag data into many small segmented block size tag data, however, the overhead of the each segmented block shall increased and the through time may be increased. When we implement the block size control mechanism, we have to consider the following issues.

Packet errors can be occurred in a short interval when a large tag data is transmitted

The previous pilot experimental (Figure 5.8) result revealed that it took a lot of time to collect a large block size tag data under the actual noisy environment. The reason why is that large block size retrieval causes packet errors at a short intervals and the requesting the same block is very high cost because the each block size may be very huge.

Small block size retrieval can lead to the increase of the large overheads

If the large tag data would be divided into many segmented small block size, the frequency of packet error could be decreased. However, when it comes to increase the number of segmented block size, the overheads of the each ones has come to be increased. In other words, small block size retrieval can lead to the increase of the large overhead. In this research, we measured and/or calculated the throughput on the assumption of the overhead commands described in Figure 5.7.

The optimized block size can be the submultiple number.

The experimental result (Figure 5.12) reveals, in the contradiction to the result of the pilot experiment (Figure 5.8), that the required time where the block size is small is very large. The reason why is the large overhead which is attached in front of each block packet. Moreover, we found that the optimized block size can be one of the submultiple number from the mathematically and the experimental evaluations. (Figure 5.8 and Figure 5.12)

With the result, we proposed an adaptive control mechanism as follows.

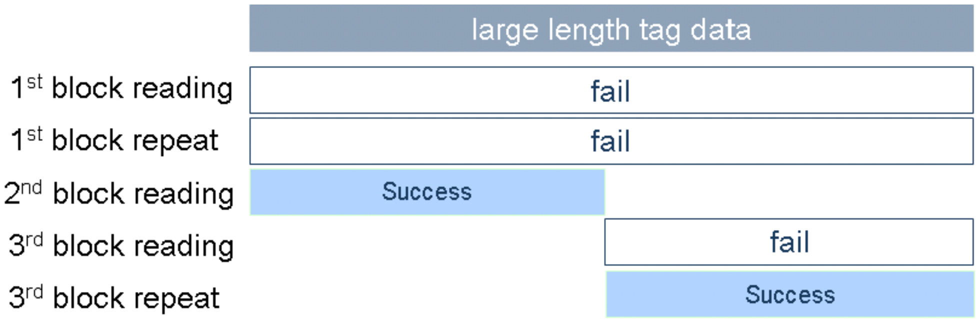

Figure 5.13: Adaptive Block Size Optimization mechanism

The Figure 5.13 shows the fundamental algorithm of adaptive block size optimization. Firstly the reader controller instructs the RFID reader of the offset and the length of the necessary tag data. If the retrieval of the necessary tag data successes, the trial could be finished successfully. However, when the first trial fails in some times, the reader controller instructs the RFID reader of the smaller length than before and retransmits the segmented block size packet adaptively.

Since there are not RFID reader/writer device which can be programmed the proposed algorithm at the middleware layer, the actual evaluation of the proposed method could not be completed. However, from some evaluations I had done, we consider that the priority of the proposal can be confirmed.

Chapter 6

Conclusion

As the industrial adoption of networked RFID expands, the importance of user-specific data has been acknowledged. An exemplary user-specific data is sensor data. There is also an industrial demand to store non-sensor user-specific data such as sales records and repair history. When we consider handling such heterogeneous user-specific data in a consistent architecture framework, a generic method to handle user-specific data needs to be developed. The requirements for generalized handling of user-specific data in the networked RFID can be summarized as follows.

- The identification of memory schema for the user-specific data in an RFID tag
- The efficient data transfer between RFID reader/writer and the corresponding RFID tag

The requirements can be satisfied when we implement the roles, "schema resolver" and "reader controller" in the middleware layer.

The schema resolver resolves the RFID tag memory schema information for efficient retrieval of user-specific data. The schema resolver may instruct the RFID tag to retrieve user-specific data in the specified memory range or may instruct a write operation. The schema resolver can be complementary to that of structured tag memory, in which all the information required to collect user-specific data is stored in tag memory, tailored to sensor data. Despite the qualitative advantage of having a schema resolver, the mandated registry lookup by the schema resolver may degrade the reading performance. Our experiment and simulation revealed, however, that the registry lookup is equivalent to or even faster than that of structured tag memory because the interaction (wireless communication) between an RFID tag and corresponding RFID reader/writer in the latter is relatively slow and a number of data needs to be exchanged to determine the range of RFID tag memory to be collected.

The reader controller helps the RFID reader/writer to collect a large amount of RFID tag data efficiently. In this thesis, the adaptive block control mechanism for large tag data reading is proposed. The mechanism features adaptive reading block size adjusting in accordance with the packet error rate. To reveal the large tag data retrieval performance under the actual radio noise environment, we measured the throughput using a commercial RFID reader/writer. The result indicated that it took a lot of time to retrieve the large tag data at one time.

Then we evaluated the large tag data retrieval time performance with a simulation program and the actual experiment respectively. From the both evaluation, we acknowledged that the priority of the adaptive block size changing mechanism can be confirmed.

Bibliography

[1] EPCglobal Architecture Review Committee. *The EPCglobal Architecture Framework Final version of 1 July 2005*, 2005. EPCglobal.

[2] Fujitsu, 2007. http://www.frontech.fujitsu.com.

[3] EPCglobal Architecture Review Committee. *The Application Level Events (ALE) Specification, Ver.1.1*, September 2007.

[4] ISO/IEC 15961. *Information technology -Radio frequency identification (RFID) for item management- Data protocol: application interface*, 2004.

[5] ISO/IEC 15962. *Information technology -Radio frequency identification (RFID) for item management- Data protocol: data encoding rules and logical memory functions*, 2004.

[6] EPCglobal Architecture Review Committee. *EPC Tag Data Standard (TDS) Standard*, 2006. EPCglobal.

[7] IBM Corporation. *DHL breaks new ground with RFID-based real-time tracking of sensitive shipments*, 2007.

[8] R. kasper. *The Cold Chain for pereshable products in the air freight*, 2007.

[9] T Yoshimura. RFID: Expectation and Requirement from Consumer Electronics Industry. In *Proceedings of APMC 2006 workshop on Emerging Technologies and Applications of RFID*, pages 121–124, aug 2004.

[10] Mizuho Research Institute. Actual condition servey for product safety construction by profit user of ic tag. Technical report, Technical report, 2006.

[11] J Mitsugi. Rfid as the consumer product lifecycle information recorder. In *IEICE society conference 2007, BS-12-5*, 2007.

[12] Aerospaceid, 2007. http://www.aero-id.org.

[13] Epcglobal, 2007. http://www.epcglobalinc.org.

[14] International organization for standardization (iso). http://www.iso.org/iso/home.htm.

[15] ISO/IEC WD 24753. *Information technology - Radio frequency identification(RFID) for item management - Application protocol: encoding and processing rules for sensors and batteries*, 2006.

[16] ISO/IEC WD 18000-REV.1. *Information technology - Radio frequency identification(RFID) for item management* -, 2007. Part6: Parameters for air interface communication at 860MHz to 960MHz.

[17] K. Emely. Distributed eventing architecture : Rfid and sensors in a supply chain, 2005. http://db.lcs.mit.edu/madden/html/theses/emery.pdf.

[18] M. Sqigger K. Garcia O. Ogan R. Kougianos E. Hai Deng, Varanasi. Design of sensor-embedded radio frequency identification(se-rfid) systemsl. *in: IEEE International Conference on Mechatronics and Automation*, pages 792–796, 2006.

[19] Leong K.S. Ng M.L Engels D.W. Cole P.H. Ranasinghe, D.C. A distributed architecture for a ubiquitous rfid sensing network. *In: 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 7–12, 2005.

[20] Zhi Wang Lei Zhang. Integration of rfid into wireless sensor networks: Architectures, opportunities and challenging problems. *In: Fifth International Conference on Grid and Cooperative Computing Workshops*, GCCW'06:463–469, 2006.

[21] D. Naruse. Integration of real space information based rfid model. Master's thesis, Keio University, 2006.

[22] Nist. Ieee-p1451 draft standard. http://ieee1451.nist.gov/.

[23] EPCglobal Architecture Review Committee. *Class 1 Generation 2 UHF Air Interface Protocol Standard*.

[24] J. Mitsugi. Multipurpose sensor rfid tag. *APMC 2006 workshop on Emerging Technologies and Applications of RFID*, pages 143–148, 2006.

[25] ISO/IEC. *Information technology - Radio frequency identification(RFID) for item management* -, 2006. Part6: Parameters for air interface communication at 860MHz to 960MHz Amendment 1: Extension with Type C and update of Type A and B.