18

RFID Context-aware Systems

Jongmyung Choi
Mokpo National University
Korea

1. Introduction

For several decades, RFID technology has been advancing and widening its application fields by penetrating into the traditional industries with the benefits of efficiency, costs saving, and fast processing. As the result, nowadays it is widely used in logistics, inventory, security, supply-chain, manufacturing, and retails industries. For example, Wal-Mart required its top 100 suppliers to attach RFID tags to their goods [1], and logistics and manufacturing companies including Siemense have been adopting RFID rapidly [2]. Experts expect that it will widen faster its application fields thanks to low tag cost, advance in network technology, and various trials of combining ICT (information and communication technologies). With ICT, RFID systems are providing users with smarter services such as world-wide object identification, and world-wide monitoring and tracking objects.

In the ubiquitous computing, context-awareness is one of the hottest issues in the academia and the industries. The basic idea of context-awareness is that the systems are able to recognize not only environmental condition and events but also internal system status, and provide users with smart services that fit on users' intention and their current situation. We call these external and internal conditions and events as context. Therefore, context-aware systems usually have sensing sub-systems that recognize the environmental condition and events and a smart module that determines the current context and the services according to the context. For sensing sub-system, they adopt camera, temperature sensor, GPS, motion sensor, and other sensors. As you might expect, RFID's identifation facility is also an important source of sensing the external condition and events. In fact, there have been a myriad of works that apply RFID to context-aware systems. Some read user data from tag to provide smart services according to user's role [3], and anothers identify user location by reading tags attached to fixed objects to provide services related to the location [4,5]. Still anothers recognize user's intention by reading object identifiers and provide services about the objects [6]. These RFID-based context-aware systems are very diverse and the number of these systems is increasing.

Though RFID has been gaining interests in context-aware systems, little attention has been paid to studies on survey of existing systems or studies on context-awareness in RFID systems in detail. These studies are needed to help system stakeholders understand RFID-based context-aware systems. Therefore, we survey the existing systems and introduce our RFID context-aware system in detail. We have two purposes for this article. The first one is to survey and classify existing RFID context-aware systems. By surveying the existing applications, we identify what kinds of context are used, what kind of context-aware services they provide. We group these applications into five categories according to the data

Source: Sustainable Radio Frequency Identification Solutions, Book edited by: Cristina Turcu,
ISBN 978-953-7619-74-9, pp. 356, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

type of tag value: user identifier, location, object, neighborhood, and behavior. We also group context-aware services into five categories: branch, trigger, resource scanning, follow-me, and context recording. The classification of tag value types and context-aware services can help stakeholders to understnad the RFID based context-aware systems. Specially, these are helpful to analysts and designer with understanding and modleing context and services. After survey, we define some concepts on context-awareness including context attribute, context, and other concepts in measurable terms, and we also introduce some desing issues on context-aware system.

The second purpose is to show how we apply RFID to our museum exhibition guide system, named MyGuide. This system is able to provide user specific information on the exhibit to users. For example, when a user does not have knowledge about the exhibit and he/she speaks Korean, it explains the exhibit with plain Korean. It can also provide multimedia data to help the user to understant the exhibit when he/she has a multimedia-enabled device. For this context-aware service, it adopts four context information: location, background knowledge level, device facility, and language. RFID is used to locate users in the musemum, and the system gets other information from user registration on the web. After considering context modeling issues such as context extensibility, inference, simplicity, and implementation, we draw two contexts -TEXT and MULTIMEDIA - from four context attributes. For software architecture, we adopt ECAM (event-context-action-model) architecture. The architecture is suitable for RFID context-aware systems, because it supports event-drivenness and context-awareness. MyGuide prototype system is implemented in Win32 platform with Java language, 13.56 MHz RFID readers, card type tags, Oracle DBMS, and Apple's Darwin Streaming Server. MyGuide shows how to model context from multiple context information including RFID and how to provide user-specific contents according to the context.

This article consists of five sections. In Section 2, we describe about RFID system: its elements and various existing systems. After this, we define some concepts on context-awareness such as context attribute and context and explain context-aware features in RFID systems in Section 3. In Section 4, we introduce our RFID system, MyGuide, which provides exhibition guide according to user's location and other user's information and preference. Finally, we reveal our conclusions in Section 5.

2. RFID system overview

2.1 RFID system elements

RFID is an identifying technology that uses wireless radio frequency, and it is widely used in various fields because of its merits: efficiency, costs saving, fast recognition, multiple recognition, and long recognition distance. In general, an RFID system is basically made up of three elements as shown in table 1: readers, tags, and servers. Table 1 shows the three elements and their classification.

Tags keep a small amount of data such as identifier in their internal memory, and they are classified according to their memory types: read-only and read-write. Most of tags have mobility because they can be carried with users; however, some others are immovable because they are attached to fixed objects. Mobile tags have close relationship with users because they move with users, but immovable tags are related with fixed objects or locations.

Elements	Criteria	Classification		Description
Tag	Power Source	Active		Long recognition distance
		Passive		Short recognition distance
	Memory	Read Only		Unchangeable value (ex. Identifier)
		Read/Write		Changeable value (ex. State, money)
Reader	Mobility	Stationary		Attached to fixed location
		Mobile	Mobile	Connected to wireless network, PDA type
			mRFID	Connected to CDMA or GSM, Cell phone type
Server	Service	Primary		Primary server for the services
		Secondary		Auxilary server

Table 1. RFID Elements Classification

RFID readers read data from tags and pass the data to the primary server. They can be classified into two groups: mRFID [8,9] and stationary RFID according to whether they are connected to a mobile communication network including CDMA and GSM. The former is connected to mobile communication networks, and the later is connected to wire-networks. The mobile RFID has two advantages: mobility and connectivity with the network. It is mobile because it is attached to a user's hand-held device such as a PDA or mobile phone. It is connected to the network anytime and anywhere because it uses CDMA or GSM network. Therefore, users can use it any places.

There are two kinds of servers: primary server and secondary server. The primary server is the server that receives the data passed from readers and processes it according to its business logic to meet users' requests. In stationary RFID systems, computer systems usually play the role of the primary server. However, in mobile RFID systems, hand-held devices usually play the role of the primary server. The secondary server is the server that helps the primary server by processing requests from it. ONS (Object Naming Server) in EPC (Electronic Product Code) systems [10] is one example of a secondary server. As RFID systems are getting complicated, the importance of the secondary servers increases. In addition, a variety of secondary servers come out. In mobile RFID systems, a service broker [8] is also an example of the secondary servers.

Typical RFID systems are connected to each of three elements shown in Fig. 1. In the figure, T_1 and T_2 mean RFID tags, and R represents RFID readers: fixed reader or mobile reader. P is a primary server and S is secondary servers for the system services.

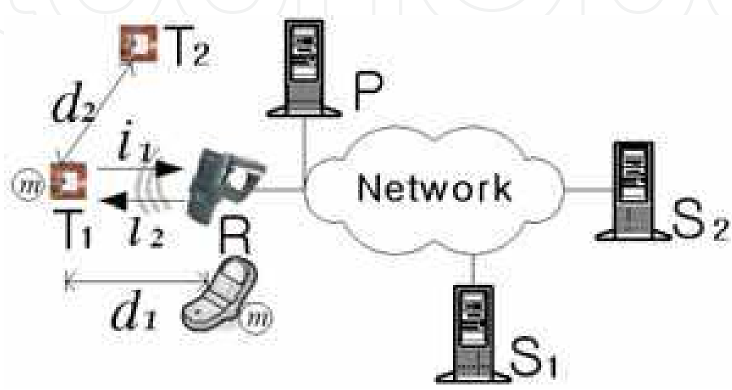

Fig. 1. RFID System Model

As illustrated in Fig. 1, tags have mobility in stationary RFID systems; however, readers have mobility in mobile RFID systems. In both cases, readers can access the tag data within the distance d_1. When readers recognize the tag data i_1, they transmit the data to the primary server P over the network. Then primary server P configures the services, and if it needs support from the secondary servers S_1 and S_2, it transmits the requests to them. Finally, the primary server P processes the data and provides the services to the users. In some cases, tags are writable, and readers may write data i_2, into tags. Furthermore, some applications, such as smart box [7], use several tags at the same time, and they should stay within a constant distance d_2.

2.2 Various RFID systems

We survey the existing RFID systems, and identify various types of them. During the survey, we find out that most of tags keep identifer valus, but the meaning of them are all different. Therefore, we classify the tag values into five groups according to their meaning or types: user, object, location, neighborhood, and behavior. Table 2 shows the five types of tag values, their meaning, and their usage patterns in the RFID systems.

Tag value	Meaning	Usages Patterns
User	A person who owns the tag	User's ID, role, preference, etc
Object	An object that a tag is attached	Object ID, object = action[28], monitoring
Location	The location where a tag or a reader is	Location based services, Follow-me, etc
Neighborhood	The relationship among the objects to which tags are attached	Cooperation among objects, Relationship among objects, monitoring, etc
Behavior	User's behavior or intention	Monitoring

Table 2. Tag values in RFID Systems

2.2.1 User

One of the most popular areas of RFID applications is identifying users. These systems identify users for their own purposes; access control, security, payment, and managing people. With these systems, users keep their own tags and there are stationary readers at some location such as gates, doors, or other places. The systems identify users by reading the values stored in the tags and matching them to their databases. Fig. 2 shows a user identifying system. By approaching the user's tag to the reader, the reader scan the value in the tag and the system determine who the user is or whether to allow him/her to access to the cetifacte data or places.
User information in the tag can be combined with other data in the server system and provide more sophisticated services. RFID applications considering user's preference or tastes usually use this information plus his/her preference information together for determining user specific services. Another example is the role-based services, in which user information is used to determine the role of the system. This kind of applications are closely related to the security, authorization, and authentication. However, other areas can also

apply this role based service to their services. For example, Bravo [3] adopted two types of roles - student and teacher - in his application. If a teacher approaches, the system shows the lecture notes. However, if a student approaches, it shows his/her homework.

Fig. 2. User Identification

2.2.2 Object

A RFID tag may contain information – usually identifier - about an object which it is attached to. By reading the information, RFID applications can identify objects, get information about the objects, and keep track of them. These applications are used for logistics, inventory, and retail. In logistics, applications classify and deliver goods using RFID tags attached to them, and monitor to keep track of the goods while delivery. In inventory, they use tags to check which goods are stocked in and which items are out of stocks. In retail, companies can figure out which goods are sold. Retail industries enjoy the benefits of fast processing and preventing shoplifting.

Object identifiers in tags are the gateway which connects the physical world and the virtual world. Users can get information about the object and control the object from the virtual world. The EPC(Electronic Product Code) is the standardized object code developed by EPCglobal [10]. Using EPC, RFID applications send requests to EPCIS(EPC Information Service) which manages the information about objects and provides the services for the inquiries. ONS(Object Name Service) provide the location of data about the object which is indentified by the object identifier. EPC Discovery Service is the search service for the data location for the objects.

We find that some interesting researches which treat the object as the users' intention. It means that reading object identifier triggers an action which is closely related to the object with users' intention. For exmaple, during shopping a shopper lifts an item, it means that he/she has interests on the item. Therefore, the shopping assistant application may show information about the item or suggest another items that match the picked item. Fig. 3 shows the Smart Shopping Assistant system researched by Michael Schneider [11].

Another example is Trevor Pering [6]'s Elope system. The system identifies objects and provides a services which are related to the purpose of the objects. For example, users scan the tag on the remote controller for presentation, and the system allows users to use presentation. Fig. 4 shows the architecture of Elope system.

These RFID systems consider objects as the user intention, and this view changes the user interface in the computing environment from WIMP(Windows, Icons, Menu, and Pointing device) to object = action paradigm. Ailisto [12] argues that object=action paradigm can

Fig. 3. Smart Shopping Assistant

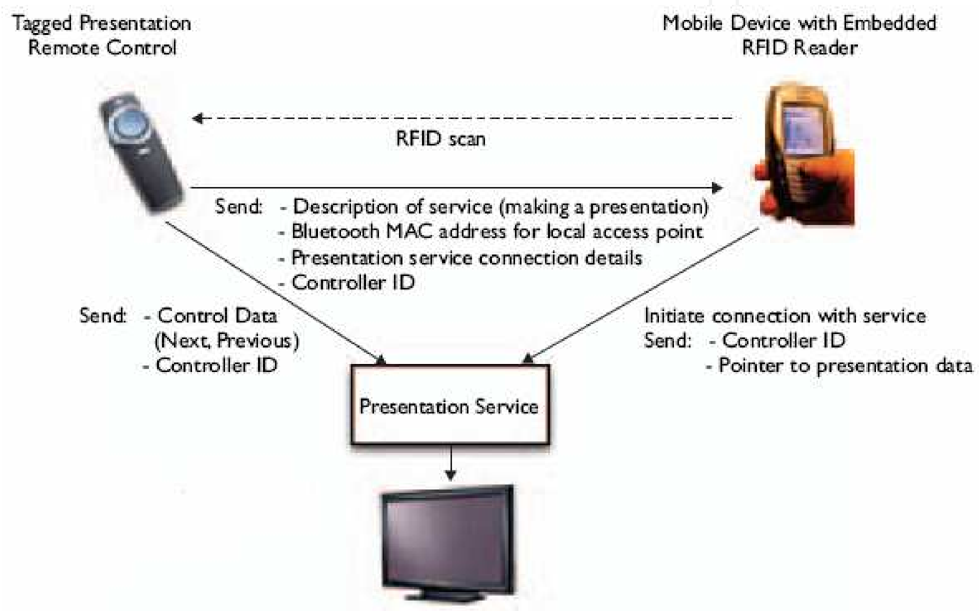

Fig. 4. Elopse System

change user interface paradigm of desktop computer by touching objects with mobile RFID readers. The typical example is the business card with RFID tag [13]. When a user touches the card with his/her cell phone equipped with RFID reader, the phone calls the person automatically.

2.2.3 Location

In context-aware systems, location is very important context information. Location based services (known as LBS) are good examples of context-aware services using location as context information. Searching for nearest restaurant from the user's location and weather service according to users' location are typical examples of LBS. There are two types for location system: one for absolute location with latitude and longitude and the other for symbolic location with relative location to the base location. There are several technologies for determining location inclusing GPS, RF singnal length, triangulation, and RFID. RFID is good for determing indoor location because it is easy to implement. Lionel's [14] does researches on determining a user's location and movement in the building. Lionel uses active tags and determines a user's location by calculating users' location with reference tags and a tracking tag.

There are RFID systems that use location information. Michael Crawford [15] developed the internet-based cooperating system. This system provides user's physical location for cooperation with people spread around the world. The location information helps the participants understand each other. For example, they know others' local time and know who participates and who leaves. There are serveral tour guide system that utilize RFID to locate users. Sherry [4] introduces an exhibition guide system using RFID. The main feature of this system is its bookmarking function. It means that visitors can bookmark any exhibit that draws his/her interests, and then he/she can make web pages with the information about the exhibits. Jani [5] also introduces an RFID-based system, mTag, which provides information to the pre-specified devices when the user carrying a tag approaches the RFID reader.

2.2.4 Neighborhood

Neighborhood is similar to the symbolic location, but it has unique features that it uses multiple RFID tags within a limited scope for cooperation of them. The systems determines their services according to the combination of tags within the scope. This neighborhood is captured from the symbolic location information. There are several interesting systems using neighborhood as context information. The representative system is Lampe's Smart Tool Box [7]. This system notifies users when he/she left tools outside tool box. In the system, the tool box embeds a RFID reader and tags are attached to each tool. Fig. 6 shows Smart Tool Box.

Fig. 6. Smart Tool Box

The neighborhood information can be used with user information. Bravo's classroom system [3] provides services based on the neighborhood information and user ID. When teacher moves to the blackboard, the system shows lecture note. However when a student moves to the blackboard, it shows the answer that the student makes.

2.2.5 Behavior

RFID can be used to identify user's behavior [16]. Context-aware systems should be able to figure out user's intention in order to provide suitable services according to context. User's behavior is a good hint for figuring out his/her intention. There are several methods to identify user's behavior: camera and active badge. However these methods are good for identifying short term behavior, but it is not suitable for long term behavior. Joshua [16] created a glove equipped with RFID reader, which reads RFID tags attached to things. This system can figure out what the user is doing. Fig. 7 shows i-Glove system, which is used to figure out user's behavior in a hospital.

Fig. 7. iGlove System

3. Context-aware RFID systems

3.1 Definitions and terms of context-awareness

Context-aware systems are able to choose and provide the most suitable service from the having the same semantics but different implementations to meet the changing contexts. In general, context-aware systems have two kinds of services: normal business services and context-aware services. Context-aware services are the business services whose contents and formats are determined by the context. That is, context-aware services are business plus context-aware features and the latter is our main concern. Context logic has a close relationship with the implicit interaction between users and external environment, and it also has a relationship with business logic because it triggers or determines the business services. Fig. 8 shows the relationship among business logic, context logic, and user interface.

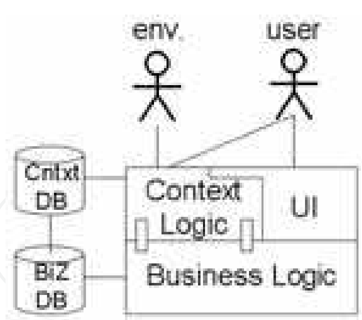

Fig. 8. Conceptual Layer of Context-aware Systems

In the context-aware systems, context is the most important issue, but there is no commonly accepted definition on it. Shilit and Theimer [17] define the context as location, identies, and objects. Hull [18] describes context as the aspect of the current situation. Dey and Abowd [19] define context as "any information that can be used to characterize the situation of entities that are considered relevant to the interaction between a user and an application, inclusing the user and the application themselves."

In this Section, we define context in more concrete and measurable terms by extending our previous work [20]. First, we define the most basic piece of information of context, and we call it context attribute. A context attribute is a semantic data type of value sensed from the external world via a sensor or an internal state value stored in a variable. Therefore context attribute is defined as Def. 1.

Def. 1. Context Attribute

Context attribute, a_i, is a sematic data type of an atomic value sensed via a single sensor or a sematic data type of a value stored in a variable.

Temperature, location, and time are good examples of context attribute. Temperature and location are semantic data types of external values sensed by sensor subsystems, but time is a semantic data type measured internally. We use the term of “semantic data type“ in order to distinguish it from the meaningless data type in the programming language. For example, 75 is represented as “int“ data type in the programming language, but it is meaningless in the context-aware systems because we do not know what it means. Therefore, the value should have the semantic data type such as temperature, pressure, or other concrete meaning to be used as context information.

Since context attribute is data type, there are possible values set for each context attribute. We call it context attribute value, and it is defined as Def. 2. For example, if a context attribute, H, is humidity type, and the current humidity is 60, then the 60 is the context attribute value for H.

Def. 2. Context Attribute Value

A context attribute value, v, is an element of the possible value set of context attribute, a.

Some system use one context attribute for their context, but others adopt multiple context attributes for their contexts. Therefore, we need to identify what kinds of context attributes are used in the systems, and we define the set as Def. 3.

Def. 3. Context Attribute Set

Context attribute set, A, is the set of context attributes used in a context-aware system.

Context is an abstract and single concept in human mind, but we need context attribute values to determine what the current context is. For example, when we think about “talking“ context, it is an abstract and simple concept but we have to check context attribute values whether their values are in the range of “talking“ context. The more complex thing is that there can be multiple sets of context attribute for a context. Let’s think about “talking” context. A computer system is able to notice the “talking“ situation by sensing the speech via a noise sensor, however another system may use a camera and recognize talkers. Therefore, we have to define abstract context and concrete context.

Def. 4. Abstract Context

Abstract context type, AC, is an abstract and simple concept for the situation.

Concrete context template is a tuple of context attribute for an abstract context. We can define it as shown Def. 5.

Def. 5. Concrete Context Template

Concrete context template, T, is a tuple of the finite number of context attribute: $T = \langle a_0, a_1, \ldots \rangle$.

Context instance is an instance of abstract context, and it consists of context attribute values. In general, context instance is determined by checking attribute values or inferencing from attribute values. For example, we can determine “talking“ context if nosy frequency value is within audio frequency, and nosy pattern is in human voice pattern. Therefore, context instance is determined by checking attribute values as defined in Def. 6.

Def. 6. Context Instance
Context instnace, c, is an instnace of context, and it is determined by evaluation of c = f<v_0, v_1, …>, where f is the evaluation function which interpretes attribute value tuple and generate a context instnace.
Most context-aware systems have several situations in their lifecycle, and we call the set of contexts in a system the context set.
Def. 7. Context Set
Context set, C, is the set of contexts used in a context-aware system for its life time.
In context modeling, we determine the context set, concrete context template, and evaluation function algorithm f. During context modeling, there are context-related issues to be considered carefully: context extensibility, dynamic re-configurability, dynamic adaptation, inference, simplicity, and ease of implementation [21].

- Context extensibility: Is it easy to extend context? Is able to add more contexts or context attributes?
- Dynamic re-configurability: Is it possible to re-configure the system after extending contexts?
- Dynamic adaptation: Is the system possible to adapt itself to the change of context dynamically?
- Inference: Does the system need inference facility for generating context from context attribute values? Is the inference correct and efficient?
- Simplicity: Is it easy and simple to model context?
- Ease of implementation: Is it easy to implement context model?

We can classify context-aware systems according to extensibility of context into two types: closed and open. The systems of the first type have the fixed context set; i.e. the contexts of the system are fixed at design time and never change. In other words, the context-aware services are also determined at design time.
Def. 8. Closed
A context-aware system is called 'closed' when its context set C is fixed and determined at design time.
On the other hand, the systems of the second type use open context set. These systems allow new context and new context attributes to be added at run time. If the context attribute set changes at run time, the system is open context attribute and it should support the ability to handle it.
Def. 9. Open Context
Context-aware system is called 'open context' when its context set C is not fixed at design time. At runtime, there is a time t that meets the condition; $|C|_t = n$ and $|C|_{t+1} = m$, where $n \geq 1$, $m \geq 1$, and $m \neq n$.
Def. 10. Open Context Attribute
Context-aware system is called 'open context attribute' when its context attribute set A is not fixed at design time. At runtime, there is a time t that meets the condition; $|A|_t = n$ and $|A|_{t+1} = m$, where $n \geq 1$, $m \geq 1$, and $m \neq n$.
Context-aware services are another important issue to be considered. They influence system usage pattern, service scenarios. As the result, it also influence on systems architecture, design pattern, and development platform. We survey the existing context-aware systems, and classify context-aware services into five types according to the role of context in the services [20]. The five types are as follows:

- Branch: context determines a service among several ones, its contents, or its output format.
- Trigger: change to context triggers the execution of a service.
- Resource Scanning: the system scans the external resources and provides services that are combined with external resources. Context plays the role of medium of cooperation.
- Follow-me: the system keeps user's context, follows him/her, and provides services with the help of surrounding context. Context plays the role of the medium that connects the past and the present.
- Context Recording: the system records and saves the current context for later retrieval.

The service types listed in the above are not exclusive to each other, so that a service may be included in one or more groups. For example, a context-aware service named "A" is triggered on the change of context, and the contents of the service are determined by the context. Then the service is included in both Branch type and Trigger type.

3.2 Features of RFID systems for context-aware services

RFID systems can get user's intention by noticing the change of a tag value or understand the surrounding conditions by reading tags attached to objects, humans, or locations. In general, RFID has five interesting features compared to other sensors. First, the value that it reads is clear and explicit. Other sensors may have unclear values because of jitter or noise. RFID systems can read multiple data concurrently at higher speeds than other sensors. Furthermore, in many cases, tag values are categorized and registered in a database. It means that we do not need to apply complex algorithm such as statistics, probability, or filtering to get tag values as context attribute values.

Second, to perceive environmental values, RFID needs at least two elements: a tag and a reader. Therefore, when the system reads a tag value, it means that the system has at least two values: the tag value and the reader's ID. In most cases, the reader's ID plays the role of location identifier. However, in mobile RFID systems, the reader can provide the location of the user and the user identifier because it is combined with the user's mobile phone.

Third, a tag value may have a variety of meanings based on its data type. For example, the data from a tag may mean a user identifier or a location identifier according to the meaning of the data. In this case, the user identifier and the location identifier are significantly different when determining the contents of services.

Fourth, RFID is easily combined with other sensors. RFID cannot sense the physical environment such as temperature and humidity by itself, but it is able to fuse with other sensors and to perceive the physical environment. Even though the system adopts multiple sensor types, RFID will play the main role in the system because in most case, RFID triggers the services in the system.

Fifth, RFID system is event-driven. RFID system starts its operations on the event of reading tags. Therefore, the system architecture should be designed to meet event-driven features, and implementation should be done in event-driven programming.

3.3 Analysis of context-aware RFID systems

3.3.1 Analysis

There is no clear cut boundary of what RFID systems are context-aware and what are traditional. Therefore, we classify them according to whether they have smart services or not. If it provide smart services with the help of RFID, we consider it as a context-aware system. With this criteria, we summarize the existing context-aware RFID systems, and show the context and the service types in Table 3. The six types of context are user (U), object (O), location (L), neighborhood (N), and behavior (B). In addition, the service types are Branch (S), Trigger (T), Resource Scanning (C), Follow-me (F), and Context Recording(R).

Systems	Features	Context Attribute	Service Types	Context Extensibility
Bravo's [3]	Provide services according to a user's role	U	T, S	Closed
Elope [6]	Provide services with the paradigm of object = action	O	T	Closed
City Tour Guide [4,5]	Provide services based on the user's location	L	T, S	Closed
Refrigerato [29]	Provide services based on the objects in neighborhood	N	S, C	Closed
iGlove [16]	Recognize user's activity or record user's activity	B	R	Closed
RFID Game[30]	Provide users with game score or rules based on the internal status	O	S	Closed
Experience Blogging [31]	Record user's experience and context, and share them with others	L, O	R	Closed

Table 3. Context-aware Services

3.3.2 Context attribute model for RFID systems

From the five data types of tags, we get the following information: Who (User), What (Object), Where (Location), With Whom/What (Neighborhood), and Why (Behavior). With tags and readers, we get answers to the 5W questions. When combined with internal status, the context information will be much richer because it can provide time and extra information. With tag, reader, and internal status, we can model the context of RFID systems.

Our context models are of two types according to the reader's types: stationary and mobile. We separate them because the context information in each reader is different. Fig. 9 shows the context model for stationary RFID systems. In these systems, the reader has identification information. And it provides location information that is about absolute position or about symbolic position. Therefore, tag values may be used for user identification or object identification. The context may need time and purpose information, and these are maintained as internal values within the system. Context also needs extra sensory data for environmental information.

The context model for mobile RFID systems is a little different from stationary RFID systems because the mobile RFID reader provides both location information and user identification. In mobile RFID systems, the reader is attached to the cell phone or PDA. Fig. 10 illustrates the context model for mobile RFID systems.

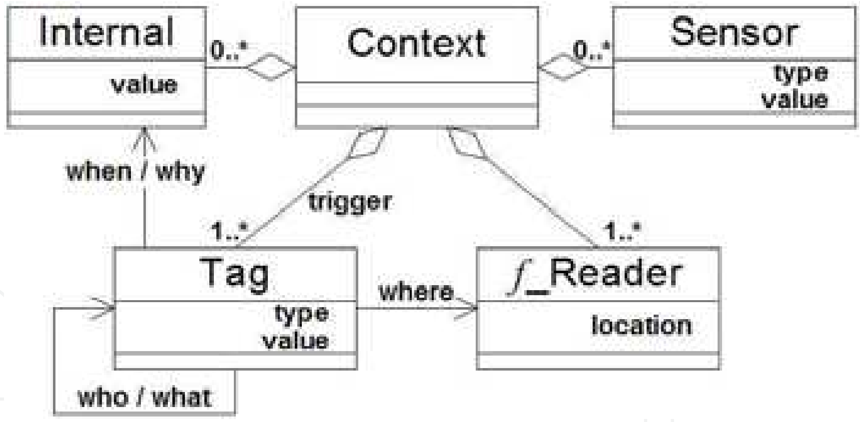

Fig. 9. Context Model for Fixed RFID Systems

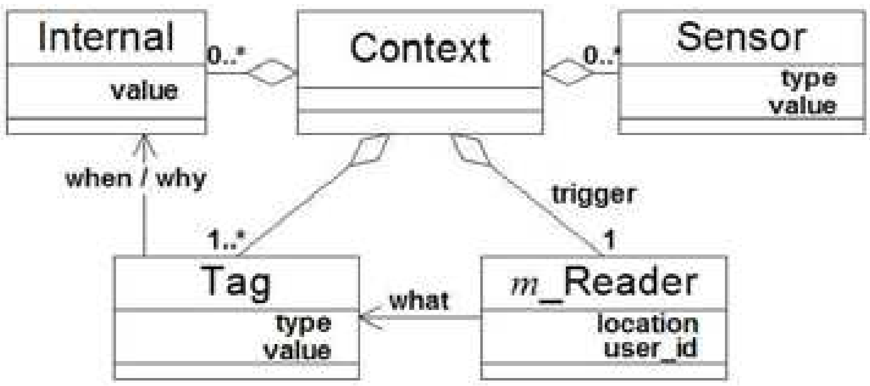

Fig. 10. Context Model for Mobile RFID Systems

3.3.3 System context trasition model

In general, RFID systems start their operation based on event-driven approach, and they have internal states. Their states changes according to the events. Similarly, the context-aware RFID systems have their current context and have all possible contexts as their context set. During the service life time, the context changes from one to another and the services also change according to the change of context. Therefore, we can represent the transition of context with context transition diagram. The context transition diagram is similar to the state transition diagram and it shows the change of contexts in the system. Fig. 11 shows the example of the context transition diagram. In the figure, state "Idle" and "Calling" are business logic, and "Guiding" is a context-aware service. In the context-aware service, there are state transition between "Multimedia Guide" and "Text Guide."

3.3.4 System architecture

Software architecture is important enough to determine the system's framework and design. Most of RFID systems trigger their services on the event of RFID tag reading, so they are considered event driven. With this feature, architects determine the event driven architecture for context-aware architecture, there are event-driven architecture with context information. We propose ECAM (Event, Control, Action, Model) pattern [22] for RFID context-aware systems because it supports event-driven and context-aware. This architecture is the extension of the ECA (Event, Control, Action) [23] pattern by adding Model for context information.

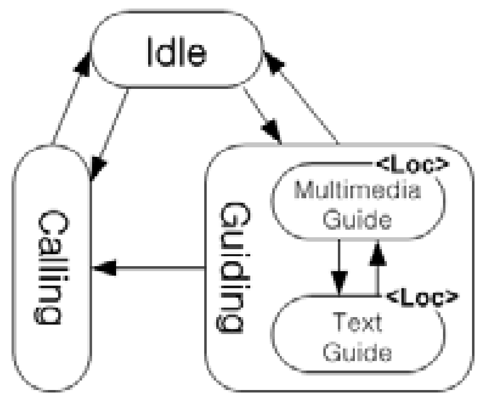

Fig. 11. An Example of Context Transition

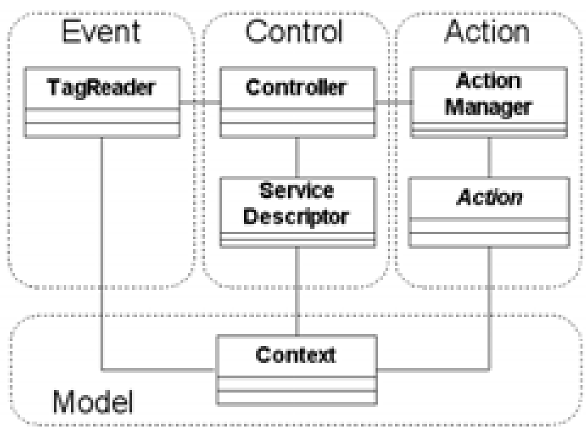

Fig. 12. ECAM Architecture

In ECAM pattern as shown in Fig. 12, TagReader reads the RF tag and generates a READ event. The event is transferred to the Controller in Control, and Context in Model. Context gathers context information from multiple sources and determines the current context of the system. Therefore, it should implement the context evaluation function defined in Def. 6. Controller determines the service contents based on the Service Descriptor and the current context determined by Context. Then the Controller transfers the information about the service to ActionManager. ActionManager provides the service.

4. Case study of a context-aware RFID system: MyGuide

4.1 System overview and requirements

There have been several RFID-based and context-aware tour guide systems because it is a interesting research topic for the academia and the tourism industry. We also decide to design and implement a RFID-based and context-aware guide system, named MyGuide [24], but we have to add more context information into our system to meet the system requirements. The main goal of our system is to provide the most suitable information to the visitors in the most suitable format. The following scenarion shows the basic usage senario and functional requirements of the system.

Scenario

When a visitor arrives at the museum, he/she registers his/her information (background knowledge, language) and his/her mobile phone information (facilities, phone number), and receives an RF tag.

After registration, the visitor walks around the exhibits, finds an interesting exhibit, and accesses the RF tag to the reader attached to the exhibit. Then the system sends a message to the visitor. The message contains the information about the exhibit, or the URL for the multimedia information about the exhibit.

From the scenario, we draw the sequence diagram as shown in Fig. 13. We figure out RFID reader, database for context information, context-server, and contents server from the scenario. The RFID reader sends the tag value to the context server, and it identifies the visitor. After that, it also can access context information stored in the context database using the tag value as a key. The contents server keeps the explanatory contents about the exhibits, and provides suitable contents to the visitor.

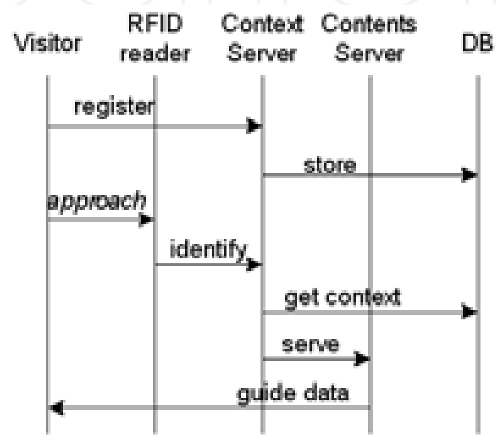

Fig. 13. Sequence Diagram for Scenario

From the basic scenario and the sequence diagram, we determine the overall architecture of the system. Fig. 14 shows the overall architecture of MyGuide system. When a visitor approached to the exhibit, the context server determines the context, and contents server provides guide information through the network.

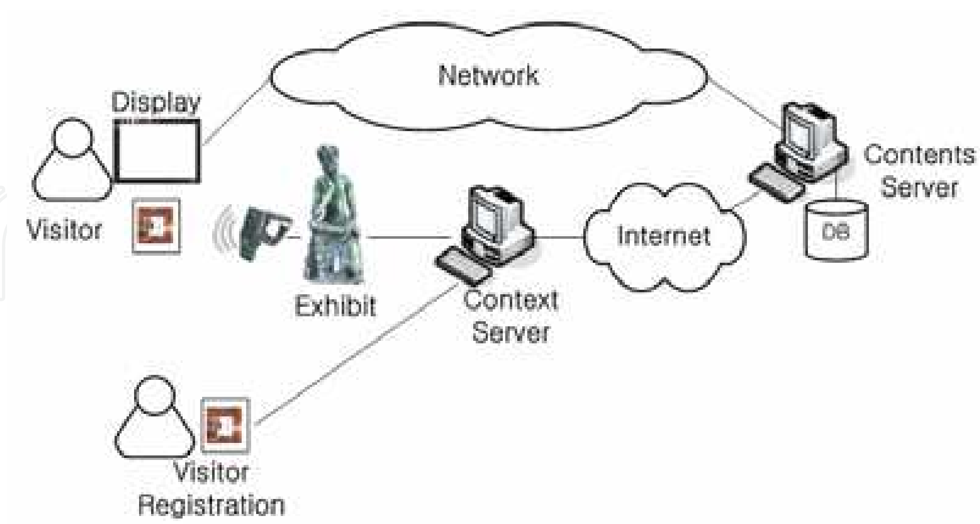

Fig. 14. Overall Architecture of MyGuide System

From the goal and the scenario of MyGuide, we are able to draw the functional and non-functional requirements. There are two basic functional requirements: "register users" and "provide information.

- (F1): register user information. This information is used to determine user-customized guide information.
- (F2): provide a user with information about the exhibition next to the user in user-customized format and with user-customized contents considering user's background knowledge of the exhibition.

Besides the two basic functional reqruiements, there are at least six non-functional requirements as follows. Most of these requriements are related to the usuability issues, but we exclude performance, throughput, and security requirements because we intend to focus into RFID and context-awareness in our system.

- (R1) Customized Services: Existing guide systems provide the same services to all visitors without considering the visitors' background knowledge or age. This can feel unsatisfactory to the visitors. Children and beginners don't understand the explanation with jargon, and experts are not satisfactory with an explanation that contains only general information. Therefore, our system considers the visitors' background and provides appropriate information according to their profiles.
- (R2) Easy Use: The end-system or terminal should be easy to use, and the visitors should feel comfortable using the technology.
- (R3) Personal Usage: The visitor should be able to get and control the services at any time when he/she wants in the museum.
- (R4) Multimedia Services: Our system should be able to deliver multimedia data to the visitors. Multimedia content is very effective to explain information to a visitor whether he/she is a child or an adult, or a beginner or an expert. On the other hand, if the terminal does not support multimedia service, it should be able to provide text data at least.
- (R5) Less Cost: Our system should be implemented with the least costs and efforts because of limited budgets. Therefore, we prefer to use the existing technologies without developing new technologies.
- (R6) Multiple Languages: One of the interesting requirements from stakeholders is for the system to support multiple languages as the number of visitors from other countries increases. Therefore, our system should support guide contents in multiple languages.

In order to meet the basic requirements, we have to determine our fundamental plarform and technologies. Since there are several altanative platforms and technologies, we have to evaluate them and choose one of them. Table 4 shows the altanative technologies and the evaluation result.

●: 5, ○: 3

Technologies / Requirements	Location			Mobile Device		Messaging Protocol	
	RFID	mRFID	Triangulation	Cell Phone	PDA	WAP	J2ME
F2	●	●	○	●	●	●	●
R1	●	●	●	●	○	N/A	N/A
R2	●	●	○	●	○	N/A	N/A
R3	●	●	●	●	●	N/A	N/A
R4	N/A	N/A	N/A	●	●	●	●
R5	○	○	○	●	○	●	●
R6	N/A	N/A	N/A	●	●	N/A	N/A

Table 4. Evaluation of Possible Technologies

After the evaluation, we choose three basic technologies: RFID, mobile phone, and context-awareness. There are several reasons why we choose these technologies as follows;

- RFID: An RF tag is small enough to carry with the visitor and it is natural and easy to use. It is a stable technology because it has been used in various fields for a long time. Furthermore, there are ample system components and software modules for RFID systems, so we can reduce costs and efforts by using the existing components instead of developing them from the scratch.
- Cell Phone: In order to support mobility, we need mobile devices including PDA, cell phone, and other dedicated mobile terminals. From several kinds of devices, we choose mobile phone based on the following merits. First, most of the new products support multimedia display, and even older ones support at least a text message service without adding any hardware components or installing any software modules. Second, we can save cost and effort in developing end-systems for providing guide service. Third, the mobile phones are easy to carry and people tend to keep their phones with them. Fourth, they are user friendly. It is their own phone, and they use it every day. Fifth, a mobile phone is always connected to a CDMA or GSM network at any place. We don’t need to consider a situation in which the end-system is disconnected.

4.2 Context modeling

In the usage scenario of MyGuide, we assume two things. First, the system database has a table that maps the exhibit and the RFID reader’s identifiers. Therefore, we can get the visitor’s location when a reader reads his/her RF tag. Second, the visitor registers his/her information with an RF tag value.

There are countless environmental elements around the context-aware system. However we cannot consider every contextual element, so that we limit the number of contextual elements in our system to the information that is necessary to meet the requirements. We identify four kinds of context attributes that affects the service of our system.

- The visitor's location
- The visitor's background knowledge
- The facilities of the visitor's mobile phone
- The visitor's preferred language

For each context attribute, it has its own value set. The visitor's location is a primary context attribute. The system can determine which exhibit draws a visitor’s attention according to his/her location. We use RF tags and RFID readers to identify the visitor’s location. When the visitor with an RF tag approaches to an exhibit, the reader attached to the exhibit identifies his/her tag value. In the perspective of guide information, the location determines the subject of data to be served.

Considering the visitor's background knowledge is the most valuable feature of our system. The more we divide the level of the background knowledge, the more the service will be satisfactory. However, we should consider development costs and the maintenance problem of contents, so that we group the background knowledge into three levels.

- Beginning: basic information for children and those unfamiliar with the subject
- Normal: general information for ordinary adults and middle/high-school students
- Advanced: advanced information for experts or college students

The facilities of the visitor’s mobile phone determine the media type of information. Some phones support multimedia data display, but others may have the only minimum functions for mobile phone. Therefore our system should consider the facilities of the visitor’s mobile phone. We therefore categorize mobile phones into two types: Text or Multimedia.

The visitor's preferred language is also an important context attribute to be considered. As the world becomes smaller, we need to serve the visitors from other countries in museum. We consider total three languages: Korean, English, and Chinese.

MyGuide adopts four context attributes for its services. The notable feature of them is that they are orthogonal and independent of each other. For example, the visitor’s location has nothing to do with his/her preferred language, and the subject and the language of the contents have nothing in common. The four types of context information can be represented orthogonally in the four-dimensional coordinate as shown in Fig. 15.

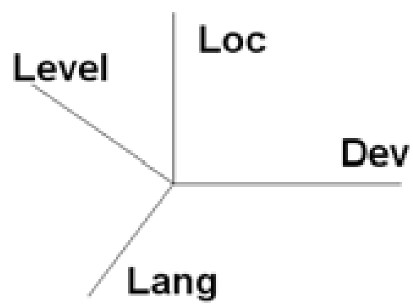

Fig. 15. Context Information Space of MyGuide System

According to our definition, a context may be represented as a dot in the coordinate system in Fig. 15. However, in MyGuide system, we have too many contexts. Let C be the set of all possible contexts in MyGuide system. Then a context c ($\in$ C) can be represented as a tuple: c = <loc, level, dev, lang>, where loc $\in$ Loc, level $\in$ Level, dev $\in$ Dev, and lang $\in$ Lang. We can imagine that Loc = { x | x is an exhibit. }, Dev = {multimedia, text}, Lang = {Korean, English, Chinese}, and Level = {beginning, normal, advanced}. Then we can calculate the number of contexts in MyGuide as follows:

$$| C | = | Loc | \times | Dev | \times | Lang | \times | Level |$$

If we define context as the tuple of context attribute values, this system has too many contexts. Therefore, we need to categorize them according to commonality and variability. In MyGuide system, media type determines the metadata and service server, so we categorize context information according to media type. Furthermore, we choose closed context model because we will not extends our context model later. Considering these issues, we model context as shown in Fig. 16. The notation used in the figure follows our prior studies [20]. The notation is similar to class diagram. According to this notation, a context may be implemented in multiple ways, so that it is represented as an interface. Similar contexts have common features, and the common features can be generated as context type. MyGuide has two contexts: Multimedia and Text and four context attribute type: location, backgounf knowledge level, mobile phone type, and language

The system maps the four context attributes to one of the two contexts after the following context reduce algorith. In the algorithm, retrieve(T, s, v) means that it retrieves data from the database meeting the condition. As you see the algorithm, you will notice that the media type determines the context.

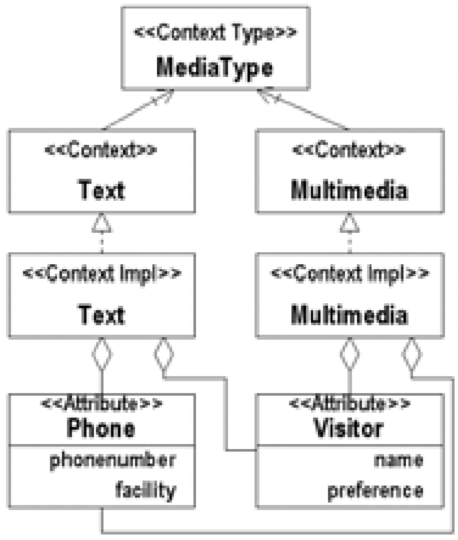

Fig. 16. MyGuide Context Model

```
Algorithm. Determine_context ( TagId, ReaderId )
  v = retrieve (Tag, tagID == TagId, "visitorID")
  (pnum, mcode) = retrieve (Phone, visitorID == v, "phonenumber, mcode")
  type = retrieve ( MediaType, mcode == mcode, "desc")
  if ( type == MULTIMEDIA )
     new MultimediaContext
  else if ( type == TEXT )
     new TextContext
  end-if
```

4.3 Implementation of prototype system

We implemented MyGuide prototype system on the Microsoft Window platform. We chose the Java language for software implementation, and MIDP [25] for mobile phone programming. We used 13.56 MHz RFID readers and card-type RF tags. We used GNU java serial communication library [26] to connect to the RFID readers. For multimedia serv ice, we adopted Darwin Streaming Server from Apple [27]. We managed the context information and the guide data on the exhibits in Oracle 10g DBMS. Fig. 17 shows the architecture of the prototype system and its components.

MyGuide starts services on READ event from RFID reader. The visitor’s profile and his/her mobile phone information are stored in DBMS. Fig. 18 shows the E-R diagram of the prototype system. Contents entity means the guide data for an exhibit. For an exhibit, there are several Contents records to provide visitors with suitable data. Visit entity means the trace data of a visitor’s access to the exhibits.

The class diagram of MyGuide is shown in Fig. 19. TagManager gets the values stored in the RF tag and the reader’s identifier, and generates READ event. On the event, the Controller takes the control and coordinates the cooperation between CntxtReducer and MsgSender. CntxtReducer infers the current context from the values of contextual elements. It uses DbFacade for retrieving contextual data stored in DBMS. Controller determines the contents of service using the context, and asks MsgSender to generate Msg for the service and send it to the visitor. Msg consists of the explanation of the exhibit and the metadata.

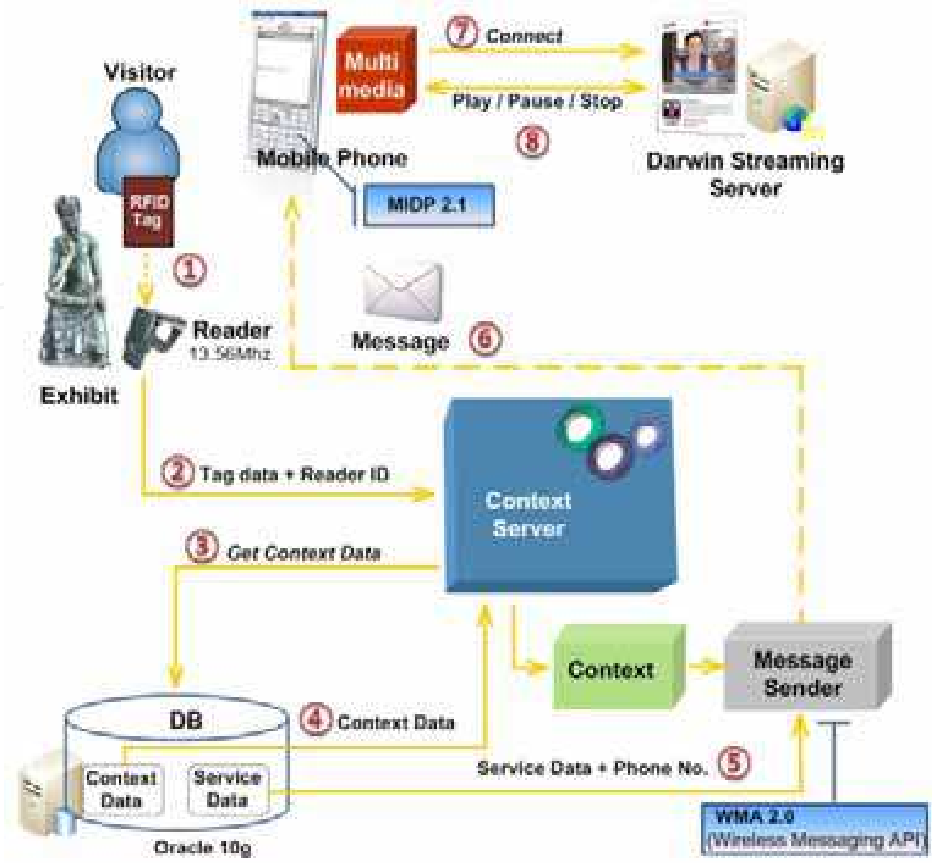

Fig. 17. Architecture of MyGuide Prototype System

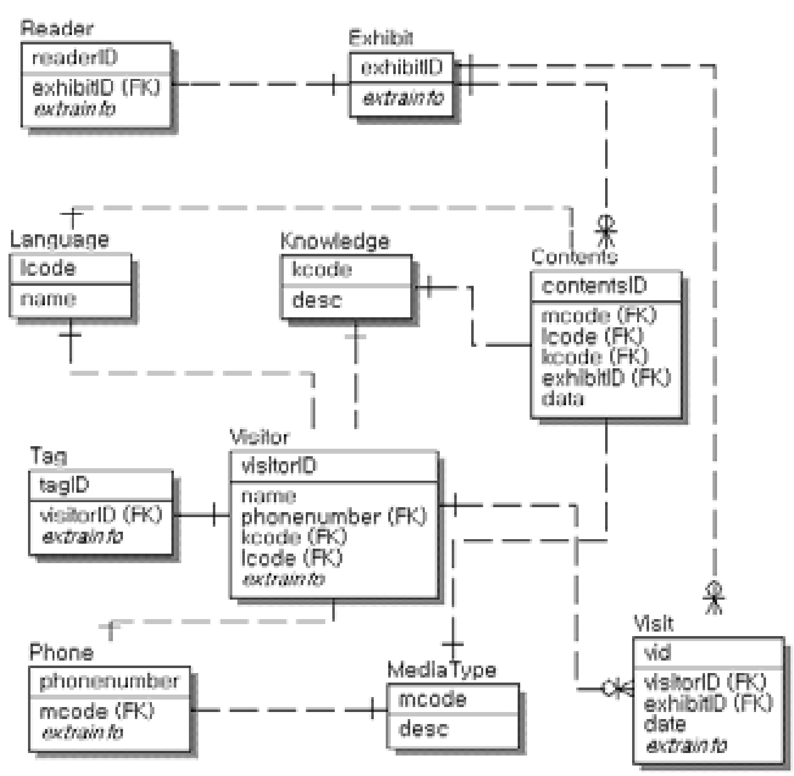

Fig. 18. ER Diagram for MyGuide Prototype System

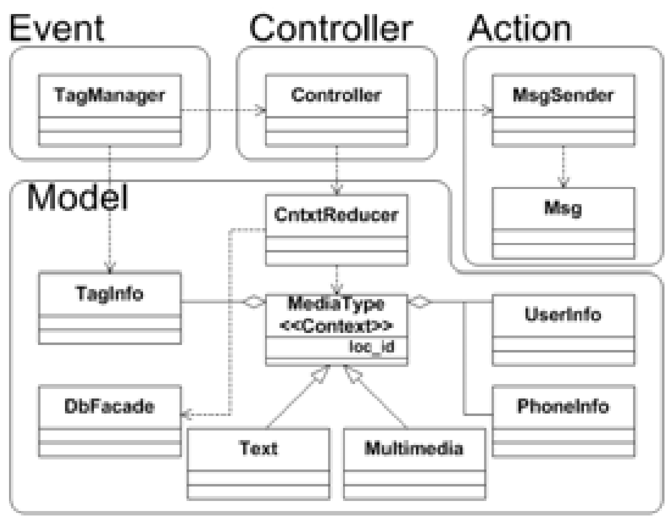

Fig. 19. Class Diagram of MyGuide Prototype System

Fig. 20 show the running examples of MyGuide prototype system. Fig. 20 (a) shows the text service at beginning level in Korean. Fig. 20 (b) shows the multimedia service at beginning level in Korean.

(a)Text Service (b)Multimedia Service

Fig. 20. Running Examples

5. Conclusions

There have been many attempts to adapt RFID technology to context-aware applications. However, until now they have paid little attention to survey of existing systems and to analyze them. In this article, we surveyed the existing context-aware RFID systems, and identified interesting issues: the meaning of tag values, their usage patterns, and context-aware services.

We also introduced a mobile guide system for exhibitions and museums. It considers four kinds of context information: location, background knowledge, media type, and language.

Based on the context information, the system provides the most suitable services to the visitor. For example, when a Korean visitor is an expert on the exhibit, and he/she has a mobile phone that supports multimedia, the system provides the multimedia type expert information in Korean. Furthermore, we introduced some interesting issues when developing context-aware systems: system requirements, context information, context modeling, software architecture, and its prototype system. We drew system requirements from stakeholders during the informal interview. From the requirements, we chose three basic technologies for the system: RFID, mobile phone, and context-awareness. We also found out the four types of context information. Because the context data are independent each other, there are too many possible contexts. Therefore, we model context from context information and determine two kinds of contexts. We chose ECAM architecture pattern for our system because it is event-driven and it can handle multiple kinds of context information. Our prototype system is implemented with 13.56MHz RFID reader, Darwin Streaming server, and MIDP APIs.

6. References

[1] Ron Weinstein, "RFID: A Technical Overview and Its Application to the Enterprise," IT Pro, May/Jun, IEEE, pp. 27-33, 2005.

[2] Peter Harrop, "Rapid adoption of RFID in manufacturing and logistics sectors," Manufacturing and Logistics IT, Aug. 2007, available at http://www.logisticsit.com/absolutenm/templates/article-critical.aspx?articleid=3190&zoneid=31

[3] Bravo J., Hervas R, Chavira G, and Nava S., "Modeling Contexts by RFID-Sensor Fusion," Proc. of Pervasive Computing and Communications Workshops, pp. 30-34, 2006.

[4] Sherry Hsi and Holly Fait, “RFID Enhances Visitors' Museum Experience at the Exploratorium,” Comm. of the ACM, Vol. 48, No. 9, Sep. pp. 60-65, 2005.

[5] Jani Korhonen et al., “mTag - Architecture for Discovering Location Specific Mobile Web Services Using RFID and Its Evaluation with Two Case Studies,” Proc. of AICT-ICIW, pp. 191-200, 2006.

[6] Trevor Pering, Rafael Ballagas, and Roy Want, "Spontaneous Marriages of Mobile Devices and Interactive Spaces," Comm. of the ACM, Vol. 48, No. 9, pp. 53-59, Sep., 2005.

[7] Matthias Lampe and Martin Strassner, "The Potential of RFID for Moveable Asset Management," Proc. of Workshop on Ubiquitous Commerce at Ubicomp, 2003.

[8] Magnus Holmqvist and Gunnar Stefansson, "Mobile RFID, A Case from Volvo on Innovation in SCM," Proc. of Hawaii International Conf. on System Sciences, 2006.

[9] Penttila K., Pere N., Sioni M., Sydanheimo L., and Kivikoski M., "Use and interface definition of mobile RFID reader integrated in a smart phone," Proc. of the 9th International Symposium on Consumer Electronics, pp. 353- 358, 2005.

[10] EPCglobal, http://www.epcglobalinc.org/.

[11] Michael Schneider, "A Smart Shopping Assistant utilising Adaptive Plan Recognition," Workshop on Adaptivity and User Modeling in Interactive Software Systems, pp. 331-334, 2003.
[12] Heikki Ailisto, Johan Plomp, Lauri Pohjanheimo, and Esko Strommer, "A Physical Selection Paradigm for Ubiquitous Computing," LNCS 2875, Springer, pp. 372-383, 2003.
[13] Lauri Pohjanheimo, Heikki Keränen, and Heikki Ailisto, "Implementing TouchMe Paradigm with a Mobile Phone," Proc. of Joint sOc-EUSAI Conf., pp. 87-92, 2005.
[14] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," Wireless Networks, Springer, Vol. 10, No. 6, pp. 701-710, Nov., 2004.
[15] Michael Crawford, et al., "RFID Enabled Awareness of Participant's Context in eMeetings," Proc. of Pervasive Technology Applied Real-World Experiences with RFID and Sensor Networks, 2006.
[16] Joshua R. Smith, et al, "RFID-based Techniques for Human-Activity Detection," Comm. of the ACM, Vol. 48, No. 9, pp. 39-44, Sep., 2005.
[17] Schilit B. and Theimer M, “Disseminating active map information to mobile hosts,” IEEE Network, Vol. 8, No. 5, pp. 22-32, 1994.
[18] Hull R, Neaves P, and Bedford Roberts J., “Towards situated computing,” Proc. Of the International Symposium on Wearable Computers, 1997.
[19] Dey A. K and Abowd G. D, “Towards a better understanding of context and context-awareness,” Proc. Of the Workshop on the What, Who, Where, When and How of Context-Awareness, ACM Press, 2000.
[20] Jongmyung Choi, “Context-Driven Requirements Analysis,” Proc. of ICCSA 2007, LNCS 4707, Springer, pp. 739-748, 2007.
[21] Jongmyung Choi, "Context: From Birth to Design," Proc. of ALPIT, IEEE, 2008.
[22] Jongmyung Choi et al., “Context Model for RFID Systems,” Proc. of 2nd International Conference on Ubiquitous Information Technologies and Applications, 2007.
[23] P. Dockhorn Costa, L. Ferreira Pires, and M. van Sinderen, “Architectural Patterns for Context-Aware Services Platforms,” Proc. of the Second International Workshop on Ubiquitous Computing, 2005.
[24] Jongmyung Choi and Jong-bae Moon, "MyGuide: A Mobile Context-aware Exhibit Guide System," Proc. of ICCSA, Springer, LNCS, 2008.
[25] Mobile Information Device Profile (MIDP), available at http://java.sun.com/products /midp/.
[26] RXTX, available at http://www.rxtx.org/
[27] Open Source Streaming Server, available at http://developer.apple.com/opensource/ server/streaming/.
[28] Matthai Philipose, et. al., “Inferring Activities from Interactions with Objects”, IEEE Pervasive Computing, pp. 50-57, 2004.
[29] M. Levinson, “All-in-One Appliance - THE REFRIGERATOR”, CIO Magazine, Feb., 2003.

[30] Christian Floerkemeier and Friedmann Mattern, "Smart Playing Cards - Enhancing the Gaming Experience with RFID," Proc. of the Third International Workshop on Pervasive Gaming Applications, 2006.

[31] Yun-Maw Cheng, Wai Yu, and Tzu-Chuan Chou, "Life is Shareable: Blogging Life Experience with RFID Embedded Mobile Phones," MobileHCI, pp. 295-298, 2005.

Sustainable Radio Frequency Identification Solutions

Edited by Cristina Turcu

ISBN 978-953-7619-74-9

Hard cover, 356 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

Radio frequency identification (RFID) is a fascinating, fast developing and multidisciplinary domain with emerging technologies and applications. It is characterized by a variety of research topics, analytical methods, models, protocols, design principles and processing software. With a relatively large range of applications, RFID enjoys extensive investor confidence and is poised for growth. A number of RFID applications proposed or already used in technical and scientific fields are described in this book. Sustainable Radio Frequency Identification Solutions comprises 19 chapters written by RFID experts from all over the world. In investigating RFID solutions experts reveal some of the real-life issues and challenges in implementing RFID.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821