# The Modeling Scheme of RFID Tags for Processing Regional Queries

**Dong Hyun Kim, Bong Hee Hong**, *Member, KIMICS*

***Abstract*—** A RFID is an automatic data collection system based on the radio frequency and is the key technology of ubiquitous computing environments. Since the locations of objects attached by RFID tags can be acquired by readers, it is possible to query the locations of tags. To query tags efficiently, the data of RFID tags should be modeled and indexed. However, since the location information of tags, the predicates of queries, are differ from coordinates of moving objects, it is difficult to model tags under the concept of moving objects, In this paper, we propose the location model of tags to represents the trajectories of tags. The location model is composed of the set and graph based approaches.

***Index Terms*—**Perceptual Space, Location Modeling, Tags, RFID

## I. Introduction

A RFID is an automatic data collection system based on a radio frequency and is the key technology of ubiquitous computing environments. The system composed of tags, readers and host computers records the information of tags in the hosts using the wireless radio signal. Since the RFID system provides valuable figures, multiple tags identification, non-visible data collection, etc, it is supposed to make an epochal change in all industrial application domains.

A tag is the small unit composed of an antenna, chips and is attached to the each object such as boxes, pallets. When a tag enters the interrogation zone of a reader, the reader identifies the tag and reports what the tag is, when the reader identifies, who the identified reader is, as shown in Figure 1. If the user wants to find out where the specific product is, he/she simply issues the queries predicated by tags or times in the system.

Since the products attached by tags are large in the industrial application domains, the information generated by tags should be large. Moreover, as a time goes on, these data accumulated over times are massive. To process the user's queries efficiently, it is required to build the index about the data of tags. However, since the locations of tags reported by readers differ from the geometric coordinates of moving objects, it is difficult to model the data of tags and build the index.

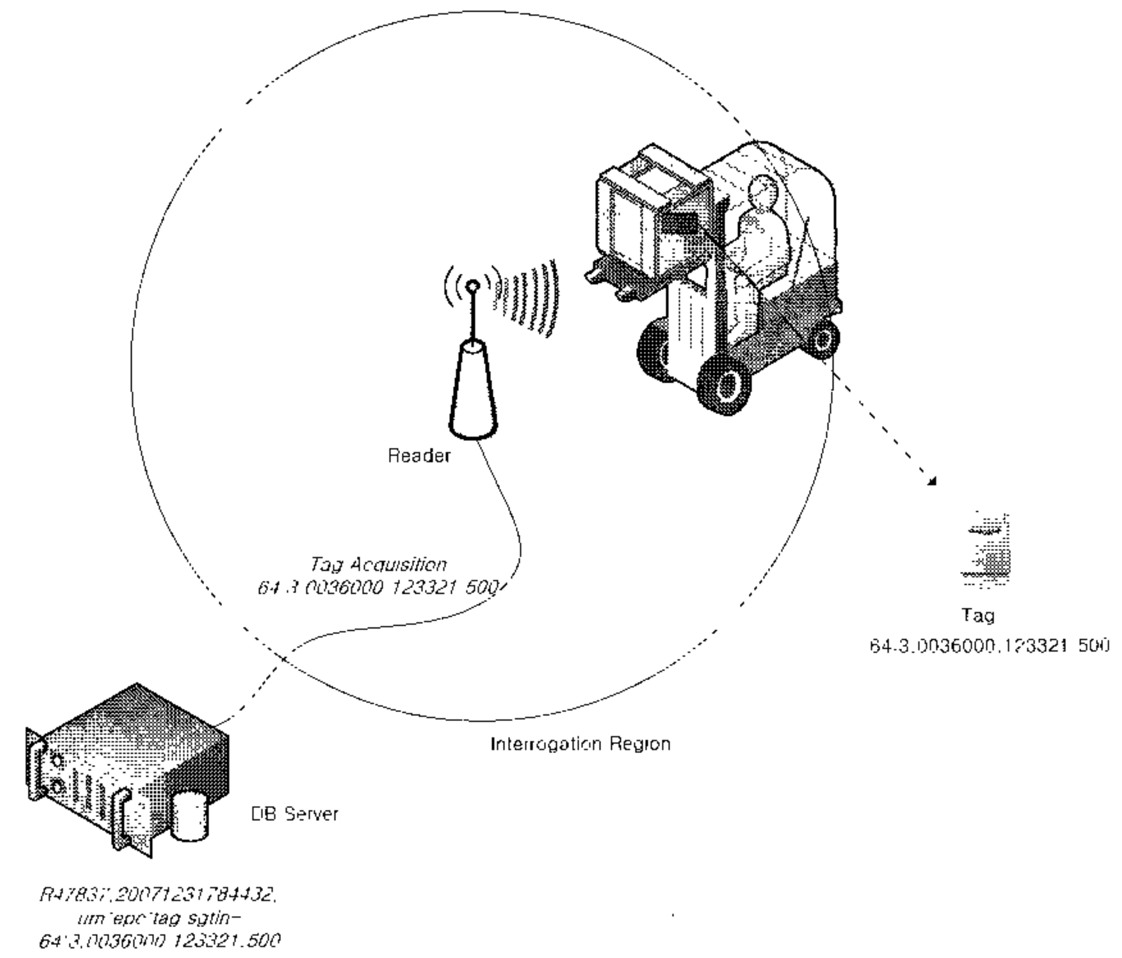



Figure 1 RFID System

In this paper, we propose the perceptual space model for a tag to represent the location information. The perceptual space is the space where the user identifies as a unit based on the set theory. We also present a tag flow graph to define the relationship among perceptual spaces. The tag flow graph is the hybrid structure based on the directed graph and a hierarchy. By using the proposed perceptual spaces and the tag flow graph, it is possible to model trajectories generated by tags and build an index based on the proposed the perceptual space.

In section 2, we will introduce related works on the location models. The EPCIS which is the de-facto standards will be explained in section 3. Section 4 describes the figures of a tag and the proposed modeling scheme for a tag will be introduced in section 5. Section 6 shows the model of tag movements and examples to process queries for tags. Section 7 gives a summary.

## II. Related Works

In [1][2][3], they map object IDs to location IDs in location services. The Guide project[1] defines the

Manuscript received March 7, 2008.

Dong Hyun Kim is with the Division of Computer & Information Engineering, Dongseo University, Busan, 617-716, Korea (Tel: +82-51-320-1801, Fax: +82-51-327-8955, Email: pusrover@dongseo.ac.kr)

Bong Hee Hong is with the Department of Computer Engineering, Pusan Nat'l University, Busan, 609-735, Korea (Tel: +82-51-510-2424, Fax: +82-51-518-6990, Email: bhhong@pusan.ac.kr)

locations which are interested by the tourists using the WaveLAN AP. They contain two distinct objects: navigation point objects and location objects. The location objects represent attractions in the city and have states representing its physical location within in the city. The Active Badge system[2] captures and stores the identifier of the user's badge using the symbolic location where the badge is identified. In this system, they can only process point queries with minimum modeling efforts. But, allowing sets of locations to overlap, the range queries may be used. In QoSDREAM[3], they define sets of locations and record the set identifier where a mobile object is observed. In these systems, they support range queries but have a problem causing considerable efforts to process them.

In [4][5], spatial regions, such as floors and rooms, are modeled by the location users and connected to each other for representing connectivity and distance. They mainly provide navigational services with the positions of individual objects. However, since they do not have the direct notion of ranges, it is difficult to process range queries without extensions.

In [6][7], they structure the hierarchy of locations. EasyLiving[4] and MOOsburg[7] models the spatial inclusion of locations and an organizational structure. For example, a company may organize its locations into development, marketing, research and production. Also, ranges and their spatial relations can be reflected in a hierarchy. However, it is difficult to define a distance concept and proximity between locations.

In [8][9], they combine a set based approach with a hierarchical approach. The Active Map[8] and the Semantic Space[9] model the inclusion relation as well as the connected-to relation between locations. Also, they can maintain two different location models where one reflects the distance and the other reflects the ranges. These approaches show rich expressiveness power but suffer from the massive modeling efforts.

## III. EPCIS

The EPC Network Architecture is the de-facto standard for a middleware system of RFID published by the EPC Global[10]. It is composed of Tag, Reader, ALE, EPCIS and ONS, etc and defines interfaces for the upper layer. Among the layers, the EPCIS aims to make it possible for various applications to find the EPC related data using the EPC code beyond the enterprise application areas. To do this, the EPCIS provides the standard interfaces for sharing data to the application programs.

The EPCIS shows the figures which differ from those of lower layers. The ALE aims to capture the raw data received by a reader and filter unnecessary data. However, when an application requests EPC related services, the EPCIS placed upon the ALE generates the data by processing the raw data with the business context as shown in Figure 2. The raw data are captured in the ALE. As a result, it returns the XML typed data to the requested application. To do this, the EPCIS stores internally EPC related historical data.

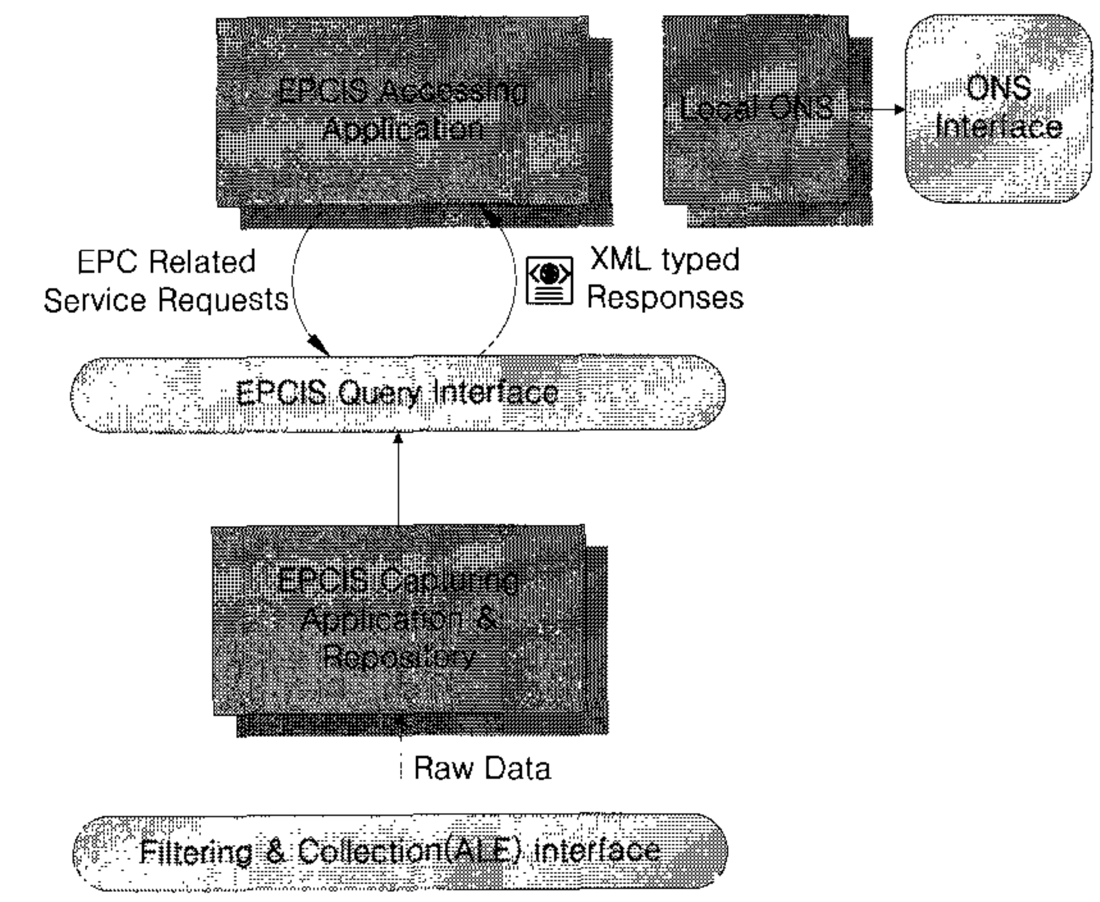



Figure 2 The Role of EPCIS

The EPCIS has a three layered structure as shown in Figure 3. The lowest layer is the abstract data model layer which defines the abstract data structures of EPCIS data. The middle layer is the data definition layer that defines which data can be exchanged via the EPCIS and describes the definition of abstract data structures and meanings. The highest layer is the service layer which defines the service interfaces that the EPCIS's client can access.

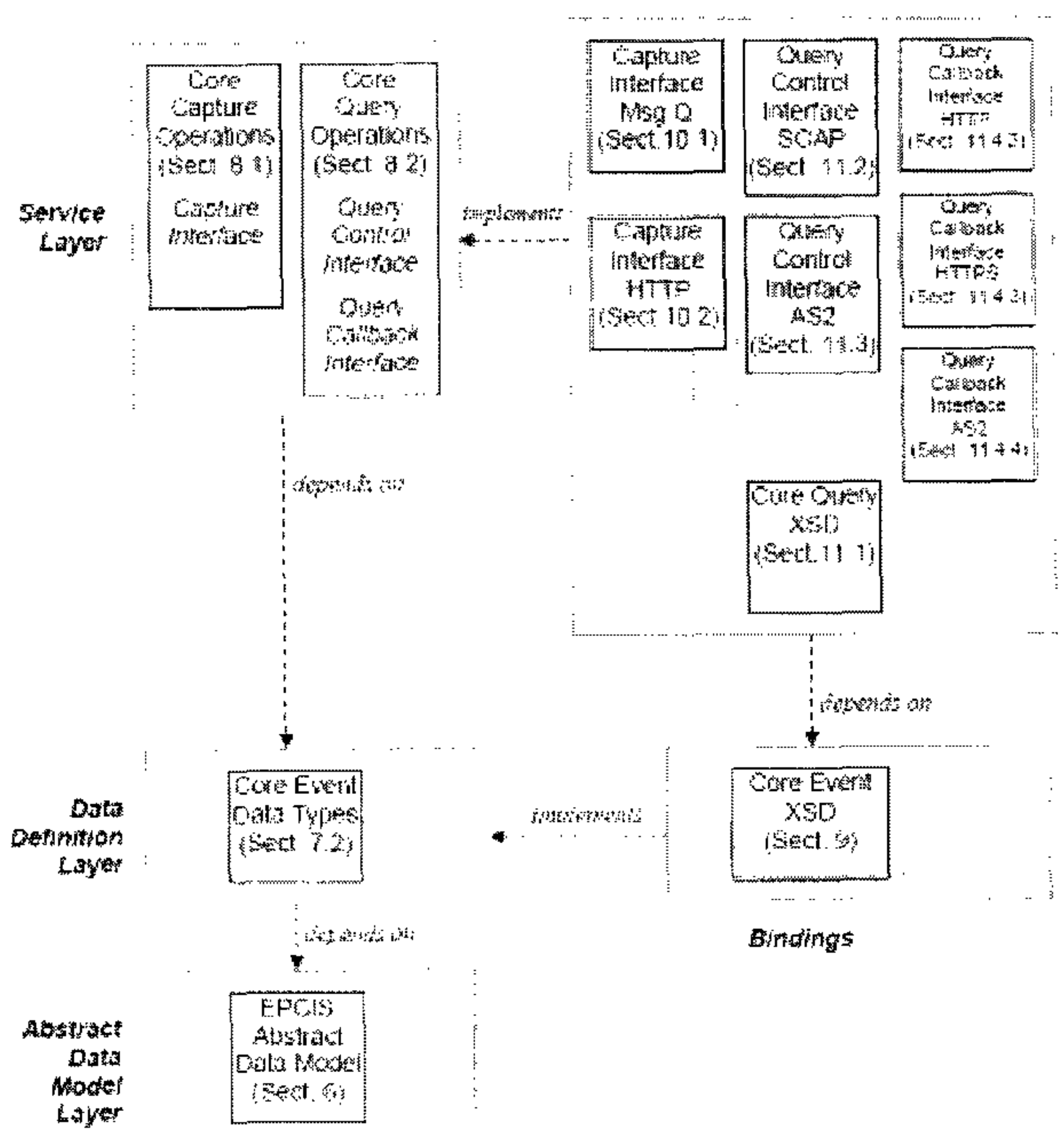



Figure 3 The Layer Structure of the EPCIS

Among these layers, the data definition layer presents the types of events which should be handled and defines four location types which can be used as location information in the EPCIS. The PhysicalReaderID and the LogicalReaderID are the physical and logical EPC code of a reader respectively and are the primitive reader types. However, these values are used in the ALE, not the EPCIS. The ReadPointID and the BizLocationID are the types to represent a location attribute of a tag. The ReaderPointID shows the location where the event occurs whenever a tag is identified. The BizLocationID is the location to be assumed that a tag is located until the following event occurs. The values of these two types are determined by the administrator under the specific business context.

## IV. Figures of a Tag

### *A. Data of Tags*

The data of tags are collected to the EPCIS via the ALE as shown in Figure 4.

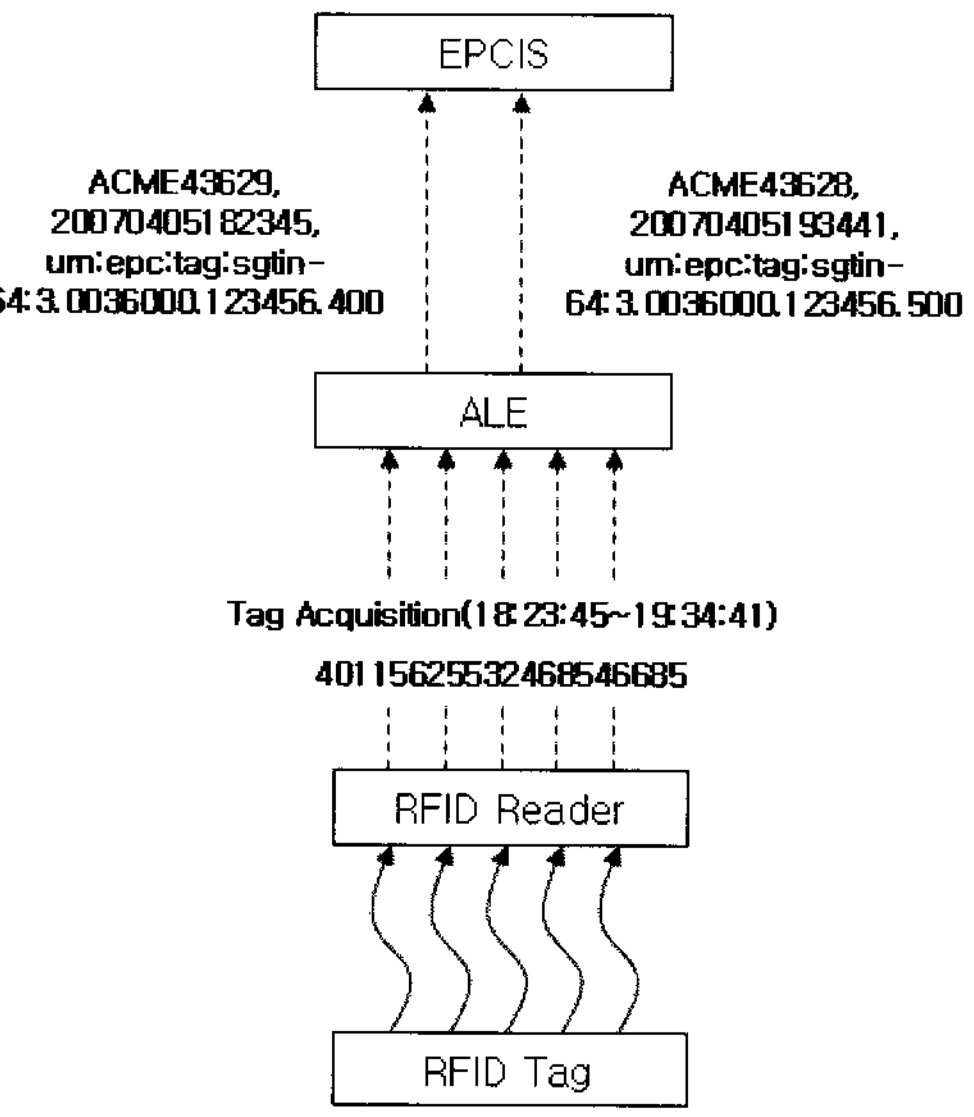



Figure 4 The Flow of Data Collection for Tags

The collected data are classified into 3 types. The first is an EPC code that is the unique identifier of a tag. The second is the identifier of a reader which identifies a tag. The last is the time when a reader identifies a tag.

These data have following characteristics. The first is that the status of data is dynamic over times. Since a reader frequently identifies a tag placed in its interrogation zone, the location of the tag is changed continuously over times when it moves. The second is that the location of a tag is related to the business context. A tag cannot figure out its location for itself. To define the location of a tag, it is possible to exploit the identifier of the reader which detects the tag as the location of the tag. However, since the identifier of a physical reader or a logical reader is not handled as the real location data in EPCIS, the identifier of a reader should be related to the identifier of the logical location defined by a business context. Thus, the location data of tags are symbolic location affected by the business context unlike moving objects.

The third is that the numbers of tags which are detected in the reader are pretty dynamic along the time. The numbers of tags in a reader can be large during the specific time intervals. Meanwhile, the tags detected in the same reader can be small during the remaining time periods. The last is that tags may not be detected by the readers placed in a specific space although the tags are really located in that space.

### *B. Queries*

The primitive queries for a location have 4 types. The position query is the query which searches for the location of a specific object. The nearest neighbor query finds the $n^{th}$ objects which are near from the specific object. The navigation query finds the path between two specific locations. The last is the region query which searches all objects included spatially in the boundary of a specific region. Since the case which finds the path between two tags does not exist, we deal with only the position query and the region query.

In Table 1, the query for the tag are classified into 4 types[11].

TABLE 1 THE TYPES OF QUERIES FOR TAGS

| | predicates | results | Location query |
|---|---|---|---|
| Find | tids, times | locations | position query |
| Look | locations, times | tids | Region query |
| History | tids | locations | |
| With | tid, times | tids | |

The Find query is the query which searches the locations where a tag is located during a time period. For example, "where is the location in which $tid_i$ is placed for $t_1 \sim t_2$" is one of the Find queries. In SQL, "select * form tag_data where tag_id = $tid_i$ and TS > $t_1$ and TS < $t_2$ ". The Look query finds the tags through the specific set of locations during a time period. For example, "who pass through $rid_k$ for $t_1 \sim t_2$ " is the Look query. In SQL, "select * from tag_data where reader_id = $rid_k$ and TS > $t_1$ and TS < $t_2$".

The History query is the query which finds the path a tag passes through during a time period. The With query looks for a set of tags accompanied by a specific tag during a time period. However, since the History and the With query are the types of a complex query, it is possible to process both queries using the Find and the Look query. For example, to process the With query, the Find query finds the locations where $tid_i$ is detected during the time periods at first. And, the Look query finds tags which stay at the result location of the previous find query during the time periods.

## V. the Locations of Tags

### *A. a Perceptual Space*

In spatio-temporal or moving objects databases, a space is represented as x, y and z values of the Euclidean coordinate system. For example, the GPS uses the coordinates of the WGS84 to measure the position of a moving object. However, since a tag does not locate its position for itself, the usage of a reader's location is the only way to abstract the location of the tag.

In this paper, we assumed that a tag always move along the flow of a business where the system is applied because a tag cannot move freely. To represent the physical space of a business application as the logical space of the system, a global space and a perceptual space are defined as follows.

- Definition 1. a global space, S
  - A global space is the whole space of a business flow which a system manages.
  - In a system, the one global space should be existed.

- Definition 2. a perceptual space, PS
  - A perceptual space is the space which a system user identifies as one unit.

A perceptual space is defined initially by a system administrator along the business flow and is determined logically regardless of a physical space. One physical space can be defined as one perceptual space or more than two perceptual spaces according to the work part of a business flow.

For example, as show in Figure 5, the whole space of a warehouse including a building, a warehousing space and a delivery space is defined as a global space, S, in the system which manages a distribution center. Each work part is defined from $PS_1$ to $PS_5$ along the goods delivery flow.

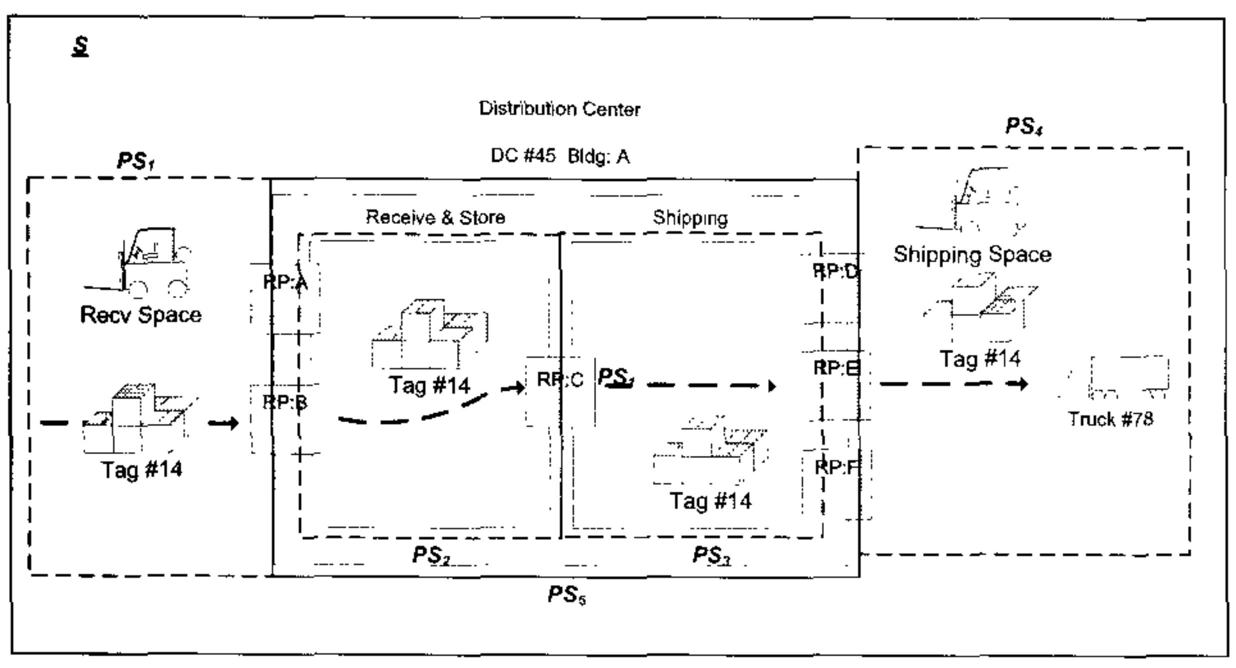



Figure 5 The Example of Perceptual Spaces.

These perceptual spaces have a lattice typed hierarchical structure. A parent perceptual space can be defined as a set of child perceptual spaces. As shown in Figure 5, $PS_5$ includes $PS_2$ and $PS_3$. Thus, a terminal perceptual space and a non-terminal perceptual space can be defined as follows

- Definition 3. Terminal Perceptual Space, TPS
  - A terminal perceptual space is a space of a lowest level in the hierarchical structure of perceptual spaces.

- Definition 4. Non-Terminal Perceptual Space, NTPS
  - A non-terminal perceptual space is a remaining space excluding terminal perceptual spaces in the hierarchical structure of perceptual spaces.

One child perceptual space can have more than two parent perceptual spaces. But the logical spaces of terminal perceptual spaces cannot be intersected.

- Definition 5.
  - $NTPS \cap NTPS = \varnothing$ or $\{NTPS, TPS\}$

- Definition 6.
  - $TPS \cap TPS = \varnothing$

### *B. a Tag Flow Graph*

Since a reader cannot know the coordinate of a tag, the location of the tag should be acquired by using the identification events which are occurs when the tag enters in the interrogation zone. A Read Point is the logical location where the identification event occurs. To model movements of a tag, we assume that a tag should move through a read point related to a perceptual space.

- Definition 7. Read Point, RP
  - A read point is the logical point where a reader identifies a tag and represents the enter/leave of a tag in a perceptual space.

A read point may provide the basic location data to determine the existence of a tag in a perceptual space when the event occurs. However, it is difficult to determine the direction by using only the location data of a read point. To decide the direction of a tag, a business flow which is the business context of the system is exploited.

To represent the flow of tag movements based on a perceptual space and a read point, a tag flow graph is defined as follows.

- Definition 8. Tag Flow Graph, TFG
  - A tag flow graph is a directed graph which follows the predefined business flow.
  - TFG = {V, E}
    . $V=\{TPS_i | TPS_i \text{ is a Terminal Perceptual Space}\}$
    . $E=\{RP^{<i,j>}_k | RP^{<i,j>}_k = k^{th} <PS_i, PS_j> \text{ where } PS_i \in V, i \neq j\}$

In a tag flow graph, a terminal perceptual space is defined as a node of the graph and a read point is defined as a edge. For example, the movements of a tag #14 of Figure 5 is represented as shown in Figure 6.

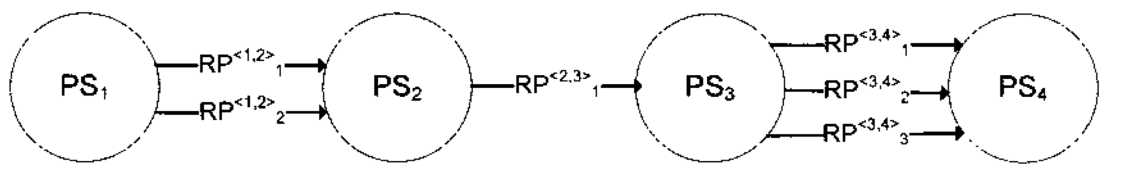



Figure 6 A Tag Flow Graph of Tag #14.

The type of a node is classified into 8 types in a tag flow graph as shown in Figure 7. For example, type (1), (2), (3) and (4) are the types which should be shown in the start and the end of a tag flow graph because they are the perceptual spaces where the movements of tags start and end.

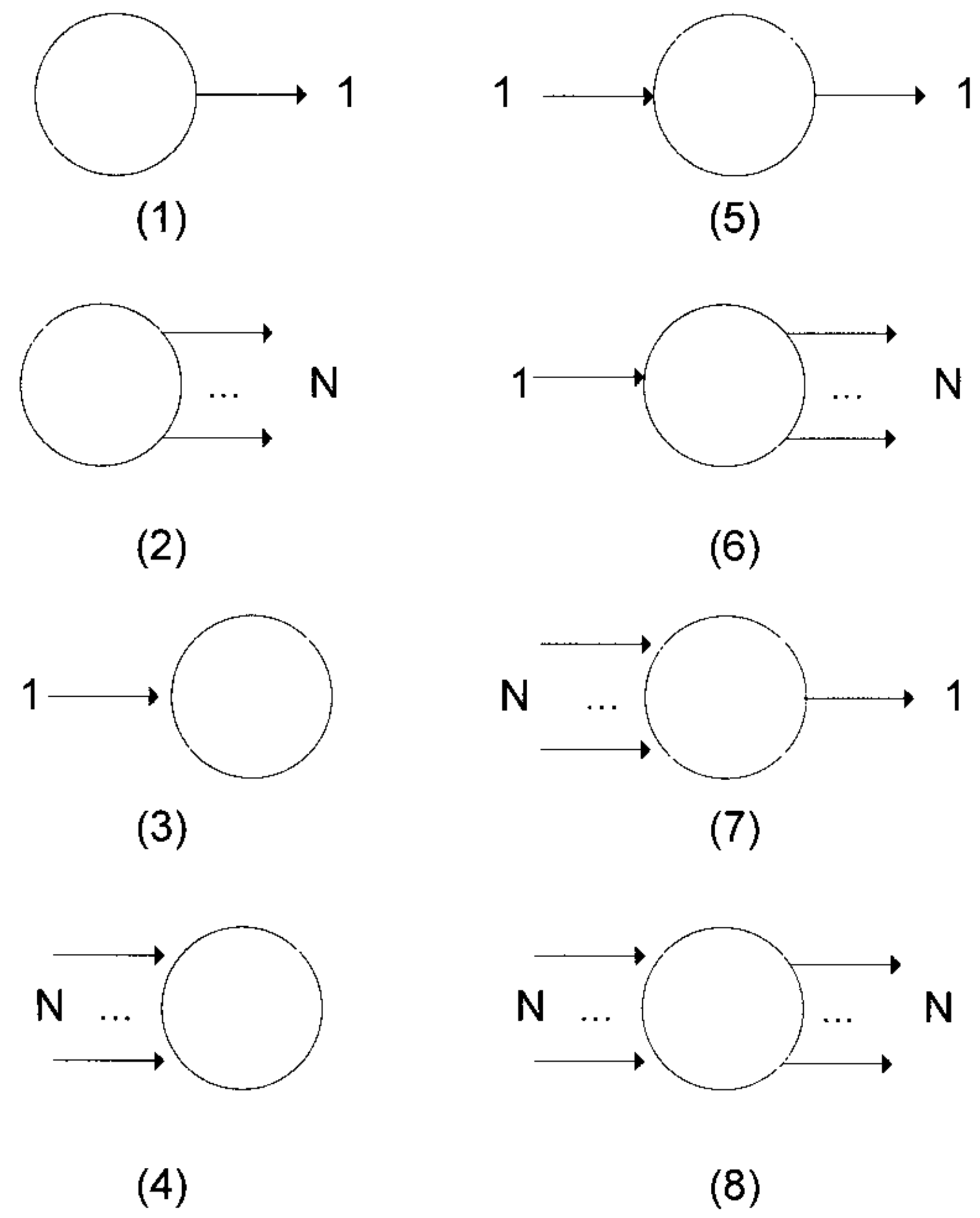


Figure 7 A Node Types of a Tag Flow Graph.

## VI a Tag Movements

### *A. Intrevals*

The connection between two perceptual spaces should be made by a read point and a tag should move through the read point from one perceptual space to the other perceptual space. In a tag flow graph, if the event of tag $tid_k$ occurs in the enter edge of a perceptual space $PS_i$, then $tid_k$ enters into $PS_i$. To know the existence of $tid_k$, it is possible to check continuously the $tid_k$ at a physical reader of $PS_i$. However, even though $PS_i$ does not check $tid_k$ physically, $PS_i$ contains $tid_k$ until the other events of $tid_k$ occur.

There is three ways to model the trajectories of a tag movement. One way is to model the trajectory as a point whenever the events occur at a enter edge of a perceptual space. The benefit of this way is to process a time-slice query at a time of the event. However, it suffers from the exhaustive search of the index in order to process a query which finds the tags in the middle of two event times. The second is to model the trajectory as a line which connects two times when the enter events occur. This scheme is exploited to represent the trajectory of a moving object. The benefit is to process a query which finds the tags in the middle of two event times. However, if we find a tag when the tag leaves, we should wait and trace the trajectory of the tag until the next enter event occurs.

The last is the interval scheme[11] to model a trajectory as a line which connects between the time at which a tag enters into a perceptual space and the time at which the tag leaves from the perceptual space. The benefit is to process the query in the middle of two event times as well as the query after a tag leaves. To process query efficiently, we adopt the interval scheme.

To represent the trajectories of a tag, three domains are exploited: id of a tag(tid), id of a perceptual space(pid) and time(t). Based on three domains, the trajectory of a tag is defined as follows.

- Definition 9. $j^{th}$ Trajectory of a tag $tid_k$, $tr_j$
  - $tr_j = \{(tid, pid, t) \in R^3 \mid tid = tid_k, pid = pid_i, t_{enter} \leq t \leq t_{leave}\}$

In Definition 9, $t_{enter}$ is the time at which $tid_k$ enters into $pid_i$ and $t_{leave}$ is the time at which $tid_k$ leaves from $pid_i$.

### *B. Examples*

In this section, we show the example based on the warehouse of Figure 5. The warehouse is composed of 6 perceptual spaces including a global space and a tag #14 moves from $PS_1$ to $PS_4$. The whole tag flows is constructed including a hierarchy as shown in Figure 8.

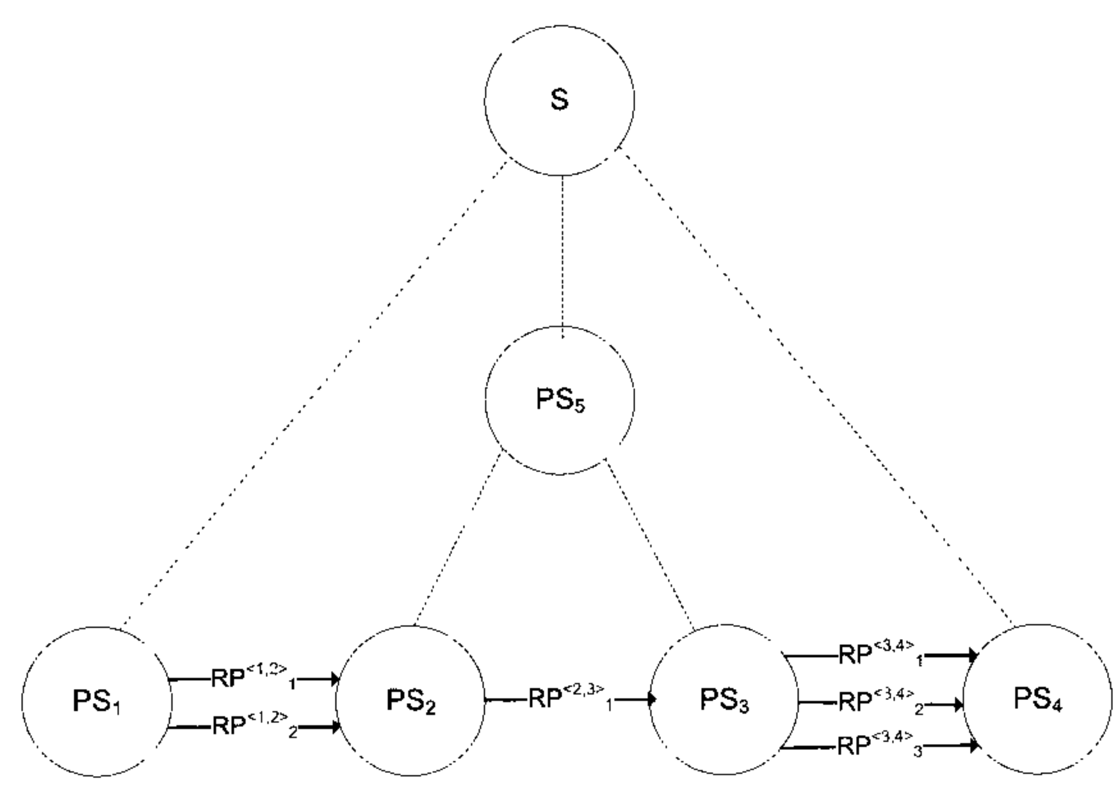



Figure 8 The Example of Whole Tag Flow Graph.

In Figure 8, a dotted line represents the include relationship between two perceptual spaces. As shown in Figure 5, $PS_5$ includes $PS_2$ and $PS_3$. A global space includes directly $PS_1$, $PS_4$ and $PS_5$.

If a tag #14, $tid_{14}$, moves through read points $RP^{<1,2>}_2$, $RP^{<2,3>}_1$and $RP^{<3,4>}_2$ at each time $t_i$, $t_j$ and $t_k$, then the trajectories of $tid_{14}$ are stored as shown in Table 2 and represented in an index as shown in Figure 9.

TABLE 2 THE EXAMPLE OF $TID_{14}$ IN TAGS

| tr | tid | pid | $t_{enter}$ | $t_{leave}$ |
|---|---|---|---|---|
| .... | .... | .... | .... | .... |
| $tr_9$ | $tid_{14}$ | $PS_1$ | $t_{start}$ | $t_i$ |
| $tr_{10}$ | $tid_{14}$ | $PS_2$ | $t_i$ | $t_j$ |
| $tr_{11}$ | $tid_{14}$ | $PS_3$ | $t_j$ | $t_k$ |
| $tr_{12}$ | $tid_{14}$ | $PS_4$ | $t_k$ | $t_{end}$ |
| .... | .... | .... | .... | .... |

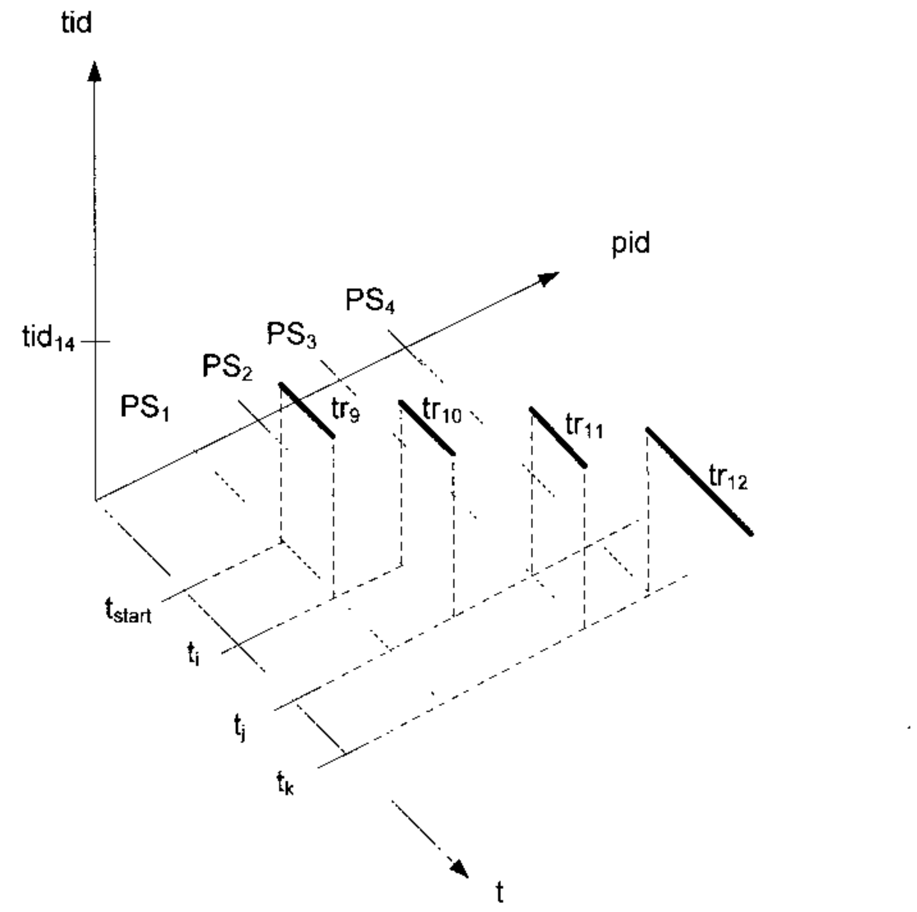



Figure 9 The Example Index of $tid_{14}$.

A query classified in Table 2 can be processed efficiently by using a tag flow graph and the index for tags. For example, a Look query Q1 to finds tags lives at $PS_5$ during $t_i$ ~$t_j$ can be expressed as follows.

• Q1: $\sigma_{\,ti \le t \le tj \wedge pid = PS5}$ (TAGS)

Since PS5 is not a terminal perceptual space, the query Q1 is divided into Q1` and Q1`` whose predicates are terminal perceptual spaces,PS2 and PS3, according to the tag flow graph of Figure 8.

• Q1`: $\sigma_{\,ti \le t \le tj \wedge pid = PS2}$ (TAGS)
• Q1``: $\sigma_{\,ti \le t \le tj \wedge pid = PS3}$ (TAGS)

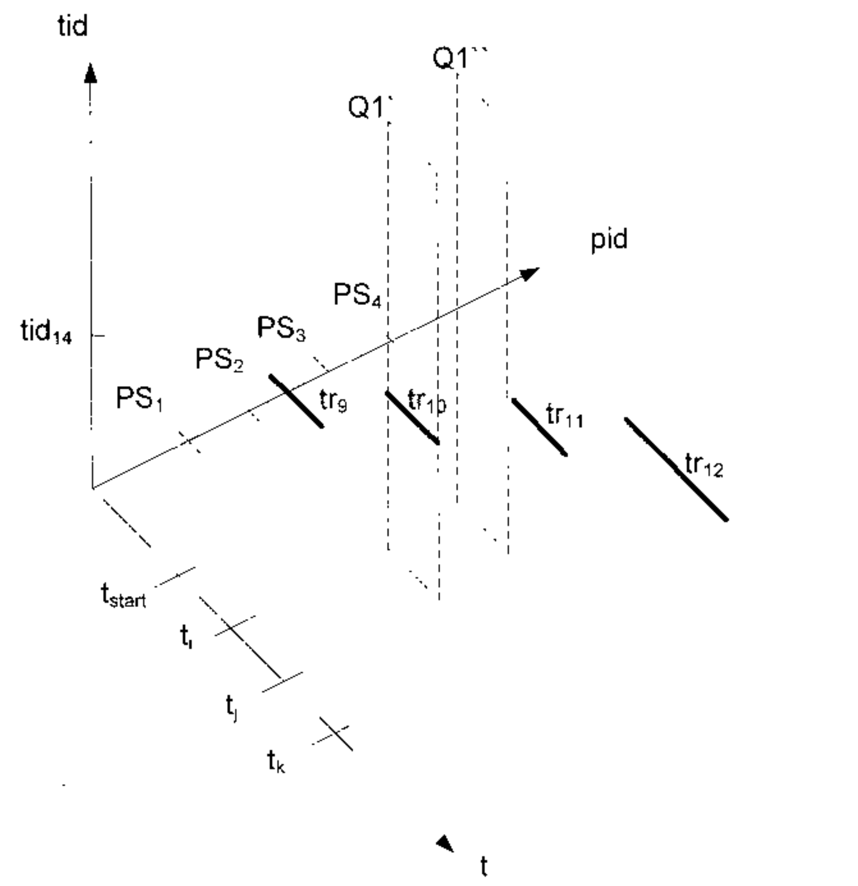



Figure 10 The Example of Query Processing.

When Q1` and Q1`` are processed on the index of Figure 9, a trajectory $tr_{10}$ can be founded as shown in Figure 10. Finally, the result set of Q1 is the union of the results of Q1` and Q1``

• Q1 = Q1` ∪ Q1`` = {$tid_{14}$}

## VII Conclusion

A RFID system is a useful method to track products or identify animals easily because it provides valuable figures, multiple tags identification, non-visible data collection, etc. Since the number of products attached by tags is usually large, the data of tags accumulated over times becomes massive as a time goes on. To process the user queries efficiently, it is required to build an index about the data of tags. However, since the locations of tags is symbolic values and shows dynamic characteristics which differ from the geometric coordinates of moving objects, it is difficult to model the data of tags and build the index.

In this paper, we propose the perceptual space model for a tag to represent the location information. The perceptual space is the space where the user identifies as a unit based on the set theory. We also presented a tag flow graph to define a relationship among perceptual spaces and represent the movements of tags. Based on the tag flow graph and the intervals, it is possible to build an index for the trajectories of tags and process the queries for the tags.

However, since a perceptual space is a logical location, the proximity between ids of perceptual spaces is not clear. To process queries more efficiently, it is required to define the proximity of values which are domains of the index for the trajectories of tags.

## REFERENCES


[1] Cheverst K, Davies N, Mitchell K, Friday A, "Experiences of developing and deploying a context-aware tourist guide:the GUIDE project," *Proc. of the 6th annual international Conf. on mobile computing and networking*, 2000, pp.20-31.

[2] Want R, Hopper A, Gibbons J, "A new location technique for the active office," *IEEE personal commun*, 4(5), pp.42-47.

[3] Naguib H, Coulouris G, "Location information management," *Proc. of the 3rd International Conf. on Ubiquitous Computing*, 2001.

[4] Rizzo M, Linington P, Utting I, "Integration of location services in the open distributed office," *Technical Report* 14/94, University of Kent, UK, 1994.

[5] O` Connel T, Jensen P, Dey A, Abowd G, "Location in aware home," *Proc. of the workshop on location modeling for ubiquitous computing*, 2001.

[6] Roy A, Bhaumik SKD, Bhattacharya A, Basu K, Cook DJ, Das SK, "Location aware resource management in smart

homes," *Proc. of the 1st IEEE International Conf. on pervasive computing and communications*, 2003.

[7] Ganoe CH, Schafer WA., Farooq U, Carroll JM, "An Analysis of location models for MOOsburg," *Proc. of the workshop on location modeling for ubiquitous computing*, September, 2001.

[8] Schilit WN, "A System Architecture for Context-Aware Mobile Computing," *PhD thesis*, Columbia University.

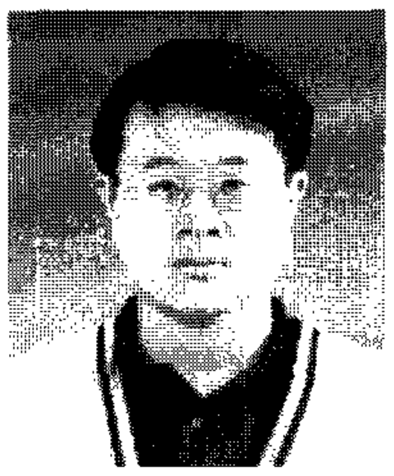
[9] Brumitt B, Shafer S, "Topological World Modeling using Semantic Spaces," *Proc. of the workshop on Location Modeling for Ubiquitous Computing*, 2001.

[10] EPC Information Services(EPCIS) Ver. 1.0 Specification, Proposed version of 29, November, 2006

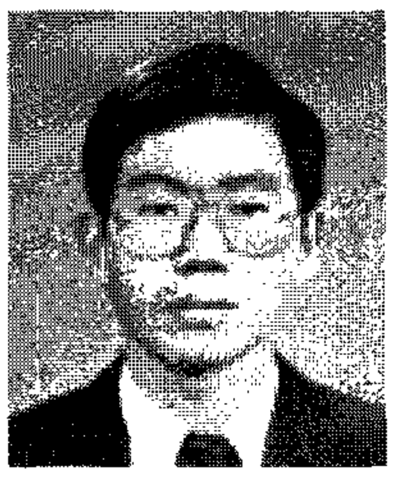
[11] ChaeHoon Ban, BongHee Hong, DongHyun Kim, "Time parameterized Interval R-tree for Tracing Tags in RFID systems," *Proc of 16th International Conf on DEXA*, 2005, pp.503-513.

## ACKNOWLEDGMENT

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD)(KRF-2006-331-D00460).



**Dong Hyun Kim**
received the B.S., M.S., and Ph.D. degrees in Computer Engineering from Pusan National University, Busan, Korea, in 1995, 1997 and 2003, respectively. His research interests include databases, spatial databases, RFID tag tracking system and indexes



**Bong Hee Hong**
received the B.S., M.S., and Ph.D. degrees in Computer Science & Engineering from Seoul National University, Seoul, Korea, in 1982, 1984 and 1988, respectively. He is now a Professor in Department of Computer Engineering, Pusan National University, Busan, Korea. His research interests include database systems, moving objects database, RFID and RFID tag tracking system.