RFID 태그의 위치추적을 위한 색인 기법에 대한 연구[1]

A Study of Indexing Scheme for Tracing of RFID Tags

안성우*, 홍봉희
(Sungwoo Ahn, Bonghee Hong)
부산대학교 컴퓨터공학과
{swan, bhhong}@pusan.ac.kr

요 약 RFID 태그 객체의 위치정보는 시간에 따라 궤적 정보가 누적되는 이동체와 유사한 특성을 가지지만 태그의 위치는 논리적인 리더의 위치로 인식되며 위치보고가 리더의 인식영역 안에서만 이루어지므로 시간축에 평행한 이산적인 시간 간격 형태로 나타나는 차이점이 있다. 기존 이동체의 위치 추적 색인에서는 이동체의 위치를 연결된 다중선으로 표현하여 색인에 저장을 하기 때문에 시공간적으로 연결되지 않은 태그의 위치 정보를 저장하면 궤적 검색 비용이 매우 높아지는 문제가 발생한다.

이 논문에서는 이동체와는 다른 태그의 위치 특성을 반영하여 태그의 궤적 검색을 효율적으로 수행하는 색인 기법을 제안한다. 제안된 색인에서는 시간적으로 연결되지 않은 태그의 궤적 정보를 검색하기 위하여 동일 태그의 위치 간의 연결 정보를 유지하는 기법을 제시하고 있다. 또한, 부모 태그와 자식 태그 간의 포함관계를 유지하는 기법을 제시함으로써 상품의 역학조사와 같이 물품에 부착된 태그 간의 포함관계를 이용한 순방향 및 역방향 궤적 검색을 효율적으로 수행할 수 있도록 하고 있다.

1. 서 론

RFID(Radio Frequency Identification)는 무선 주파수를 이용하여 태그(tag)를 장착한 객체를 리더(reader)가 자동으로 인식하고 정보를 읽는 기술로서 객체의 위치 추적이나 객체의 상태를 감시하는 자동화 생산, 재고 관리, 공급망 관리 등과 같은 다양한 응용분야에서 사용된다[1].

최근 RFID를 적용한 많은 응용분야에서 태그를 부착한 사물의 위치 정보 관리가 중요하게 부각되고 있다. 특히, 태그를 부착한 상품에 대한 유통경로 파악, 수입 식품에 문제가 발생했을 때 관련 상품들에 대한 역학조사를 위한 원산지 추적 등 태그의 이력을 효과적으로 추적할 수 있는 방법이 요구되고 있다.

태그 객체는 이동체와 유사하게 시간에 따라 연속적으로 이동하는 특징을 가지고 있으나 리더의 인식반경 내에서만 위치를 결정할 수 있으므로 태그의 궤적은 리더의 위치에 들어올 때와 나갈 때 보고하는 두 개의 시공간 위치를

[1] 이 논문은 교육인적자원부 지방연구중심대학육성사업(차세대물류IT기술연구사업단)의 지원에 의하여 연구되었음.

연결한 선분으로 표현이 된다. 리더는 이산적으로 분포되어 있으므로 태그가 리더기 간을 이동하게 되면 시간축에 평행한 이산적 간격(discrete time interval) 형태로 표현된다.

기존 이동체의 위치 추적 색인[2][3]에서는 이동체의 위치를 연결된 다중선(polyline)으로 표현하여 저장을 하기 때문에 시공간적으로 연결되지 않은 태그의 궤적을 검색 할 때 매우 높은 검색 비용을 가지는 문제가 발생한다.

최근 태그의 위치 특성을 고려하여 저장 및 질의 처리를 효과적으로 처리하기 위한 색인인 TPIR-tree(Time Parameterized Interval R-tree)[4]가 제안되었다. 그러나, 시간적으로 이산적인 간격에 대한 연결정보를 저장하지 않기 때문에 궤적 검색 시 검색 영역이 급격히 증가하여 검색 성능이 떨어지는 문제가 있다.

이 논문에서는 태그의 위치 특성을 이용하여 태그의 궤적 검색을 효율적으로 수행하는 TPIR-tree 기반 색인 기법을 제안한다. 시간적으로 연결되지 않은 태그의 궤적정보를 검색하기 위하여 동일 태그 간격 간의 연결 정보를 유지하는 구조를 제시하고 있다. 또한, 태그를 부착한 객체가 계층 구조를 가지고 이동하는 특성을 이용하여 태그 간의 포함관계를 유지하는 기법을 제시함으로써 상품의 역학조사와 같이 서로 연관된 상품의 이력정보를 포함관계를 이용하여 효과적으로 수행할 수 있는 방법을 제시하고 있다.

이 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고, 3장에서는 대상환경과 문제정의를 기술한다. 4장에서는 태그 객체에 대한 데이터 모델 및 질의를 분류한다. 5장에서는 동일 태그 위치정보 연결 및 태그 간의 계층정보를 유지하는 방법과 이를 이용한 궤적 검색 방법을 제시한다. 마지막으로 6장에서 결론 및 향후 연구를 기술한다.

2. 관련 연구

이동체의 위치 이력 정보를 저장하기 위한 색인으로는 3DR-tree[2], TB-tree[3]가 있다.

3DR-tree[2]는 기존 2차원 공간 데이터를 저장하는 R-tree에 시간을 또 다른 하나의 축으로 처리하여 객체의 위치를 3차원 시공간에서 다중선으로 표시하고 있다. 3DR-tree는 이동체의 위치를 공간적인 근접성을 고려하여 그룹화하였기 때문에 시공간 영역질의에 대해서는 좋은 성능을 보여주고 있으나 두 끝점이 알려져 있는 선분만 저장할 수 있으므로 현재 위치를 알 수 없으며 궤적 질의 시 이동체의 전체 선분에 대해서 점질의를 수행하므로 검색 성능이 좋지 않은 단점이 있다.

TB-tree[3]는 이동체의 복합 질의의 성능 향상을 위한 색인으로 엄격한 궤적 보존 정책을 따른다. 단말 노드에는 동일 이동체 위치에 대한 선분만을 저장하고 동일 이동체의 궤적이 여러 단말 노드에 걸쳐 분포할 때 각 단말 노드를 양방향 연결 리스트로 연결하여 이동체의 궤적 중 하나의 선분만 찾으면 상위 노드로의 재탐색 없이 모든 궤적을 추출할 수 있다. TB-tree는 복합 질의에 대해서는 우수한 성능을 보이나 공간 근접성을 전혀 고려하지 않기 때문에 영역질의 성능이 상당히 떨어지며 시간 영역이 주어지지 않은 이동체의 궤적 검색 시 전체 색인 영역을 검색해야 하는 문제가 있다.

RFID 태그의 위치를 추적하기 위한 색인으로 TPIR-tree[4]가 있다. TPIR-tree는 태그 객체 위치가 시간축에 평행한 이산적 간격의 특징을 가지는 것을 고려하여 태그 객체의 위치 정보의 모델링 기법과 질의 타입을 정의하고 있다. 태그의 위치 정보를 리더에서의 인식 시점에 따라 성장간격(*now interval*), 고정간격(*fixed interval*)으로 표현하여 데이터의 삽입과 노드의 분할 시 이를 부모 노드의 시간 경계까지 시간

값을 확장해 처리하여 효과적인 데이터 저장과 현재 질의를 가능하게 하였다. 그러나 동일 태그에 대한 위치 정보 간에 연결 정보를 저장하지 않기 때문에 궤적 검색 시 시간과 공간 도메인 전체 영역에 대해서 영역 질의 후 결과 셋을 시간값으로 정렬하므로 검색 비용이 높은 단점이 있다.

3. 대상 환경 및 문제 정의

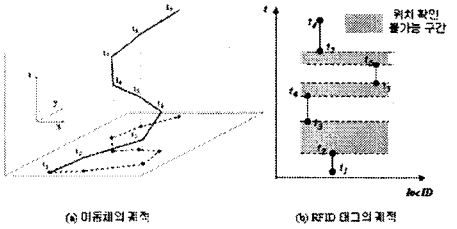
3.1 대상 환경

RFID 시스템은 리더와 태그, 서버로 구성된다. 리더는 자신의 태그 인식영역 안으로 태그가 들어오면 *Enter* 이벤트를 발생하고 인식영역을 나갈 때는 *Leave* 이벤트를 발생하여 서버에 전송한다. 리더는 실제 좌표를 가지고 인식반경을 가지는 물리적인 리더와 여러 개의 리더의 구성을 통하여 논리적인 공간(창고, 빌딩 등)을 표현하는 논리적인 리더로 구분될 수 있다[5]. RFID 시스템에서 대부분의 질의가 논리적인 리더를 대상으로 하며 물리적 리더의 구성에 독립적으로 색인을 구성할 수 있으므로 이 논문에서는 논리적인 리더를 가정한다.

3.2 문제 정의

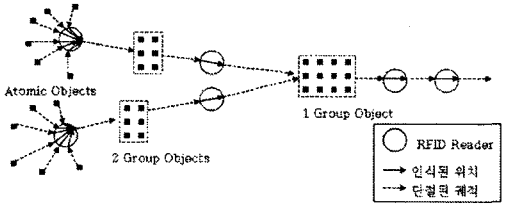
태그 객체는 이동체와 같이 시간에 따라 연속적으로 이동하는 특징을 가지고 있지만 위치 표현 방식의 차이점이 있다.

<<그림 1>(a)와 같이 이동체의 궤적은 각 보고 시점을 잇는 선분으로 표현되어 다중선의 형태로 나타난다[2]. 각 선분의 시작점과 끝점이 서로 연결되어 있기 때문에 궤적 검색 시 각 연결 지점에 대한 점 질의를 수행하여 궤적을 찾아낼 수 있다. 그러나, <<그림 1>(b)와 같이 태그의 궤적은 $tr = \{(locID, tid, t) \in R^3 \mid locID = L_n, tid = tid_o, t_{enter} \le t \le t_{leave}\}$인 시간축에 평행한 이산적인 시간 간격 집합으로 표현되기 때문에 이동체 색인에서와 같은 방법으로 궤적을 얻어내는 것이 어렵다. 또한, 태그정보가 유효한 기간 중 시간도메인 상에 위치정보를 표현하지 못하는 구간이 생기기 때문에 이동체 색인처럼 시간 단면(time slice) 질의 시 원하는 태그 데이터의 검색을 보장할 수 없다. 즉, 동일 태그에 대한 간격 정보가 시공간적으로 어떠한 연결 정보를 가지지 않기 때문에 궤적 검색 시 검색 영역 안의 모든 데이터를 검색해야 한다.



<그림 1> 이동체와 RFID 태그의 위치 표현

태그를 부착한 상품은 객체 간의 계층 관계를 가지는 특징이 있다. <그림 2>와 같이 태그를 부착한 상품은 보통 단위 상품으로 이동하지 않고 박스, 팔레트, 컨테이너 등과 같은 그룹을 이루어서 이동을 한다. 그룹은 같은 종류, 또는 비슷한 종류의 객체의 집합으로 표현될 수 있고 자동차와 부품의 관계에서와 같이 서로 다른 종류의 객체 결합으로 하나의 상위 객체를 생성할 수 있다. 이와 같은 계층 관계에서 상위 (parent)객체와 하위(child) 객체는 항상 같이 이동을 하게 되며 하위 객체의 추가, 삭제가 일어나면서 그룹의 결합 및 해체가 발생한다.



<그림 2> 태그를 부착한 상품의 이동패턴

태그의 다른 계층 특징으로 상위 객체의

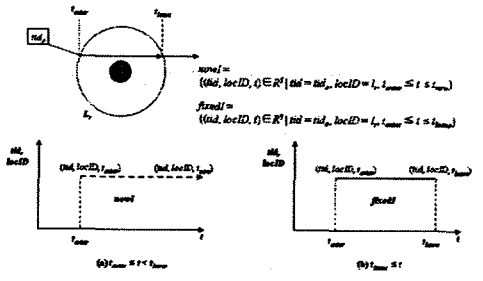
해체를 통하여 새로운 하위 객체가 생성될 수 있다. 이 경우 하위 객체가 생성되면 상위 객체는 소멸되어 동일 시간대에 존재할 수 없게 된다. 소, 돼지와 같은 축산물은 도축된 후 여러 부위로 나누어져 각 부위에 태그가 부여되므로 이러한 계층 구조의 예가 될 수 있다.

상품의 유통 경로 추적 및 역학 조사와 같은 순방향, 역방향 궤적 추적을 하기 위해서는 색인에서 이와 같은 계층 구조를 검색할 수 있는 방법을 제공해 줘야 한다. 그러나 기존의 이동체 색인과 태그 위치추적 색인에 저장되는 객체는 원자적 객체이기 때문에 계층 구조를 검색하기 위해서는 추가 저장소 및 검색 방법을 제공해야 하는 문제가 있다.

4. 데이터 모델 및 질의

4.1 데이터 모델

RFID 태그 객체에 대한 데이터 모델은 [2]에서 제시하고 있다. 그러나, 3.1절에서 살펴본 바와 같이 물리적인 리더의 좌표보다 논리적인 리더가 색인의 도메인이 되어야 한다. 이 논문에서는 색인의 도메인으로 논리리더의 번호(*locID*), 태그(*tag*), 시간(*t*)을 사용한다.



<그림 3> 태그 객체의 간격 모델링

태그의 간격은 <<그림 3>과 같이 태그가 리더의 인식영역을 벗어나지 않은 경우 간격의 끝을 결정할 수 없기 때문에 계속 확장되는 성장 간격(*now interval, nowI*)과 리더의 인식영역을 벗어나서 위치 정보가 고정된 고정간격(*fixed interval, fixedI*)으로 구분할 수 있다[2].

태그 간의 연결 정보를 유지하기 위해서 간격 정보 삽입 시 직전의 간격 정보를 찾아내야 한다. 색인에서 삽입 시 링크의 연결에 관여하지 않는 고정간격과 직전의 간격을 구분하기 위하여 이 논문에서는 직전 간격을 최종고정간격(*last fixed interval, lfixedI*)으로 정의한다.

4.2 태그의 위치추적을 위한 질의

RFID 시스템에서 태그의 위치를 추적하기 위해서는 다양한 질의가 필요하다. 질의는 Q = (*tid*, *locID*, $[t^{\vdash}, t^{\dashv}]$)로 정의될 수 있다. [1][4]에서는 태그의 위치 추적을 위해 *Find*, *Look*, *With*, *History* 연산자를 정의하고 있다. 질의에서 *History* 질의는 시간 범위가 시간 축 전체인 *Find* 질의이며 *With* 질의는 *Find* 질의를 수행하고 난 후 *Look* 질의를 수행하여 처리할 수 있다. 따라서, RFID 시스템에서는 *Find* 질의와 *Look* 질의가 기본 질의이다[4].

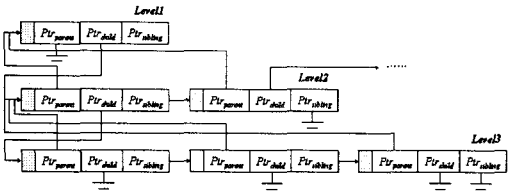
3장에서 언급하였듯이 태그 간의 계층 관계가 존재한다. 상품의 역학조사와 같은 질의를 수행하려면 해당 상품의 유통 경로를 역추적하면서 동시에 그룹 단위로 이동한 다른 상품도 추적을 해야 한다. 또한, 최종 원산지를 찾기 위해서는 어떤 상위 객체에서 분리되어 나왔는지 알 수 있어야 한다. 따라서, 태그의 위치 추적을 위해서는 포함(Containment) 질의를 지원해야 한다[6]. 상위 태그 객체, 하위 태그 객체를 찾아내기 위한 연산자로 *GetParent*, *GetChild*를 정의할 수 있다.

- $Q_{GetParent}$ = (tid_o, $[t^{\vdash}, t^{\dashv}]$) : 시간 $[t^{\vdash}, t^{\dashv}]$에서 태그 식별자 tid_o의 상위 태그 객체를 반환
- $Q_{GetChild}$ = (tid_o, $[t^{\vdash}, t^{\dashv}]$) : 시간 $[t^{\vdash}, t^{\dashv}]$에서 태그 식별자 tid_o의 하위 태그 객체 집합을 반환

5. 태그의 위치추적을 위한 색인 기법

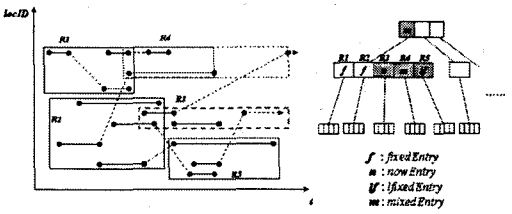
5.1 노드 구조

단말 노드는 태그의 간격 엔트리(*nowI, fixedI, lfixedI*)들을 포함하는 MBB로 구성되어 있다. 간격 엔트리는 I = <ptr_{prev}, ptr_{next}, ptr_{parent}, ptr_{child}, $ptr_{sibling}$, *tid*, *locID*, [$t^{\vdash}$, $t^{\dashv}$]> 형태로 구성된다. ptr_{prev}, ptr_{next} 는 동일 태그의 궤적을 연결하기 위해서 간격 정보간에 양방향 연결 리스트를 구성할 때 사용된다. ptr_{parent}, ptr_{child}, $ptr_{sibling}$ 는 태그 간의 계층 정보를 구성할 때 부모 태그와 자식 태그를 연결하기 위해서 사용된다. 하나의 부모 태그에 다수의 자식 태그가 포함될 수 있기 때문에 첫번째 자식 태그의 간격이 삽입될 때 ptr_{child}를 자식 태그에 연결하고 이후 자식 태그는 자식 태그 간에 $ptr_{sibling}$를 이용해서 연결한다. 태그 간 계층 구조 연결은 <<그림 4>와 같다.



<그림 4> 태그 간의 계층 구조 연결

비단말 노드는 <*cp*, *state*, MBB> 형태의 엔트리를 가진다. *cp*는 자식 노드의 주소를 가리키고 MBB는 자식 노드의 모든 엔트리를 포함하는 최소 경계 박스를 의미한다. *state*는 포함하고 있는 엔트리의 종류를 표시하며 TPIR-tree의 *nowEntry*, *fixedEntry* 이외에 태그의 삽입 시 연결정보를 구성하기 위한 직전 간격 정보를 효과적으로 검색하기 위하여 *lfixedEntry*, *mixedEntry*로 엔트리를 구분한다. *lfixedEntry*는 자식 노드에 *nowI*가 없으며 *lfixedI*이 존재할 경우이며 *mixedEntry*는 *nowI*, *fixedI*, *lfixedI*가 모두 존재하는 경우이다. 엔트리에 따른 노드의 구성은 <<그림 5>와 같다.



<그림 5> 엔트리에 따른 색인 노드의 구성

5.2 간격 데이터의 삽입

색인으로의 간격 삽입은 태그가 리더의 인식영역 안으로 들어갈 때 발생하는 *nowI*의 삽입과 리더의 인식영역 밖으로 나갈 때 발생하는 *fixedI*의 삽입이 있다. *nowI*의 삽입 시 *lfixedI*를 검색해 두 간격 간의 연결정보를 생성해야 한다. *lfixedI*의 검색을 위해 현재 질의 수행 시 질의 영역과 겹치는 비단말 노드 중에서 *lfixedEntry*, *mixedEntry*만 검색하면 된다.

*fixedI*의 삽입은 리더의 인식영역을 벗어날 때 발생하므로 간격의 끝 시간만을 알고 있다. 따라서, 이전의 *nowI*를 검색하여 간격의 시작시간을 알아야 한다. 현재 질의를 통하여 비단말 노드 중 *nowEntry*, *mixedEntry*를 검색하여 해당 태그의 *nowI*를 찾은 후 *nowI*를 삭제하고 *fixedI*를 삽입하게 된다.

```
Find_lfixedI(root, tid_new)

F1  Set N to be the root
F2  IF N is a leaf
        FOR each entry E of N
            IF E is the same tid as tid_new
                RETURN E
F3  ELSE
        FOR each entry E of N
            IF E is lfixedEntry OR mixedEntry
                invoke Find_lfixedI(N, tid_new) where N is
                the childnode of N pointed to by E
```

Insert(*root*, *I*, tid_{parent})

I1 Set *N* to be the *root*

I2 IF *N* is a leaf

Invoke Find_lfixedI(*N*, *I.tid*) for search the last fixed interval

Connect each interval and change the node state that contains previous interval if the node organization is changed

Invoke Search(*N*, *I.locID*, tid_{parent}, t_{now}) for search the parent interval

IF the parent entry E_{parent} exists

Connect E_{parent} and E_{child} with link

RETURN *N*

I3 ELSE

Choose the entry *E* by least new area enlargement.

Resolve ties by least overlap enlargement. Change the node state if the node organization is changed

I4 Set *N* to be the childnode pointed to by the childpointer of *E* and repeat from *I2*

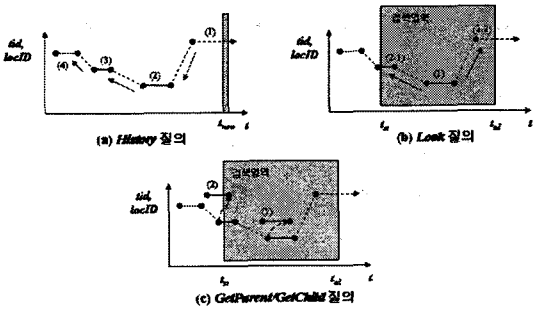
<그림 6> 삽입 알고리즘

간격 정보를 삽입할 때 태그의 계층 구조가 있다면 부모 태그와 자식 태그 간의 연결을 해야 한다. *nowI*를 삽입할 때 계층 관계 연결을 하게 되면 *fixedI* 삽입 시 *nowI*의 계층 속성을 그대로 사용할 수 있기 때문에 *nowI*의 삽입 시에만 부모 태그를 검색하여 연결을 한다. 부모 태그는 같은 리더 영역에서 *nowI*로 존재하기 때문에 $Q = (tid_{parent}, locID, t_{now})$의 영역질의를 통하여 부모 태그를 검색할 수 있다.

5.3 태그의 위치 검색

이 논문에서 제시한 구조를 사용하여 <<그림 7>(a)과 같이 해당 태그의 마지막 간격(*lfixedI or nowI*)를 검색한 후 연결 정보를 이용하여 전체 궤적 정보를 바로 추출 할 수 있다. 시간 범위가 제한된 *Find* 질의는 <<그림 7>(b)처럼 시간 영역에 대해서 시간 단면 질의를 수행하여 해당 태그의 간격 정보가 발견되면 간격 간의 연결 정보를 이용하여 나머지 간격 정보를 추출할 수 있다.

태그 간의 계층 구조를 이용하여 <그림 8(c)와 같이 시간 영역 질의를 통하여 태그 객체의 간격을 구한 후 *GetParent* 질의 시 간격 *I*의 Ptr_{parent}를 이용하여 부모 객체를 찾을 수 있다. *GetChild* 질의도 마찬가지 방법으로 수행된다.



<그림 7> 색인에서의 질의 처리

6. 결론 및 향후 연구

이 논문에서는 기존 이동체와 RFID 시스템의 태그 환경의 특징에 대해서 분석하고 기존 색인의 높은 궤적 검색 비용문제와 객체 간의 계층 관계 표현 부재를 해결하기 위해서 간격(interval) 간의 연결정보를 유지하는 기법과 태그 객체 간의 계층 구조를 유지하는 기법을 제시하였다. 제시된 연결정보 유지 기법으로 *History, Look* 질의 시 노드 방문 횟수를 줄임으로써 TPIR-tree에서 시간 영역 안의 전체 데이터 셋의 검색으로 인한 높은 검색 비용을 해결하고 있다. 또한, 태그 이동의 특징인 계층 관계를 색인에 표현하여 포함(Containment) 질의가 가능하도록 하고 있다.

향후 연구로써 그룹 단위로 이동하는 태그 객체의 특징으로 인한 동일 위치의 중복 삽입을 줄이는 연구가 필요하다. 그리고 긴 시간 간격이 저장될 때 심한 노드 겹침과 질의 성능 저하를 가져오므로 효율적으로 저장할 수 있는 방법이 필요하다.

<참고 문헌>

[1]K. Romer, T. Schoch, " Infrastructure Concepts for Tag-Based Ubiquitous Computing Applications," Workshop on Concepts and Models for Ubiquitous Computing, 2002.

[2]Y. Theodoridis, M. Vassilakopoulos, and T. Sellis, " Spatio-Temporal Indexing for Large Multimedia Applications," In Proc. of the 3rd IEEE Conf. on Multimedia Computing and Systems, pp. 441-448, 1996.

[3]D. Pfoser, S. Jensen, and Y. Theodoridis, " Novel Approaches in Query Processing for Moving Object Trajectories," In Proc. of VLDB, pp. 395-406, 2000.

[4]C. H. Ban, B. H. Hong, and D. H. Kim, " Time Paramterized Interval R-tree for Tracing Tags in RFID Systems," 16th Int. Conf. on DEXA, pp. 503-513, 2005.

[5]EPCglobal, " The Application Level Events (ALE) Specification, Version 1.0," 2005.

[6]EPCglobal, " EPC™ Information Service-Data Model and Queries," 2003.