RFID 태그 메모리 접근의 일관성을 위한 태그 연산의 동시성 제어

(Concurrency Control of RFID Tag Operations for Consistent Tag Memory Accesses)

류 우 석 † 홍 봉 희 ††
(Wooseok Ryu) (Bonghee Hong)

요 약 본 논문에서는 RFID 전자태그에 부착된 메모리의 정보를 접근할 때 발생하는 태그 연산 실행의 불완전성에 따른 태그 데이터의 불일치 문제를 분석하고, 이를 해결하기 위한 프로토콜을 제안한다. 수동형 RFID 태그는 통신의 불확실성과 단절성으로 인해 태그 메모리 접근연산의 완전한 실행을 보장하지 못하므로, 불완전하게 실행된 연산으로 인해 태그 데이터의 비일관성을 초래하는 문제가 발생한다. 본 논문에서는 태그 접근의 일관성을 유지하면서 불완전 연산의 실행을 완료시키기 위한 동시성 제어 프로토콜을 제안한다. 이 프로토콜은 불완전 실행된 연산의 대상태그를 연속질의로 정의하고 태그의 인식을 모니터링 함으로써 다른 연산들에 의한 불확실 데이타의 접근을 차단하고, 재수행을 통해 불완전하게 실행된 연산의 수행을 완료시킨다. 또한, 증명을 통해 제안한 프로토콜의 정확성, 일관성을 입증하였으며, 실험을 통해 본 프로토콜이 기존의 일관성 유지기법보다 좋은 성능을 나타냄을 보였다.

키워드 : 전자태그 메모리, 태그 접근 연산, 동시성 제어 프로토콜, RFID 미들웨어

Abstract This paper analyzes the tag data inconsistency problem caused by incomplete execution of the tag access operation to the RFID tag's memory and proposes a protocol to control consistent tag data accesses with finalizing the incomplete operation. Passive RFID tag cannot guarantee complete execution of the tag access operations because of uncertainty and unexpected disconnection of RF communications. This leads to the tag data inconsistency problem. To handle this, we propose a concurrency control protocol which defines incomplete tag operations as continuous queries and monitors the tags' re-observation continuously. The protocol finalizes the incomplete operation when the tag is re-observed while it blocks inconsistent data accesses from other operations. We justify the proposed protocol by analyzing the completeness and consistency. The experiments show that the protocol shows better performance than the traditional lock-based concurrency control protocol.

Key words : RFID Tag Memory, Tag Access Operation, Concurrency Control Protocol, RFID Middleware

· 이 논문은 2010년 교육과학기술부로부터 지원받아 수행된 연구임 (지역거점연구단육성사업/차세대물류IT기술연구사업단)

† 학생회원 : 부산대학교 컴퓨터공학과
wsryu@pusan.ac.kr
†† 종신회원 : 부산대학교 컴퓨터공학과 교수
bhhong@pusan.ac.kr
(Corresponding author임)

논문접수 : 2009년 11월 30일
심사완료 : 2010년 2월 8일

정보과학회논문지: 데이타베이스 제37권 제3호(2010.6)

1. 서 론

비접촉식 인식기술인 RFID 기술은 전자태그를 부착한 제품의 실시간 인식에서 시작하여 제품의 부가정보를 저장, 활용하는 형태로 발전하고 있다. 제품의 생산 및 유통과정에서 발생하는 다양한 정보를 태그 메모리에 기록하고 조회함으로써 제품의 이력검사, 손상된 제품의 리콜 및 폐기처리 등의 의사결정을 중앙서버 접속 없이 실시간으로 조치할 수 있다. 이러한 태그 메모리의 활용은 공급망 관리, 생산공정 관리, 자산관리 등 다양한 응용으로 활용범위를 넓혀가고 있다[1].

태그 메모리로의 접근은 무선통신(Radio Frequency Communication)을 통해 이루어진다. 수동형 태그의 경

우 자체적인 전원을 갖고 있지 않으므로, 태그 메모리는 태그가 리더와 가까운 영역에 있을 때만 접근이 가능하다. 하지만, 이때 무선 통신의 특성으로 인해 태그연산이 불완전하게 실행되는 현상이 발생한다. 무선 통신은 통신방해, 간섭, 태그위치 이동 등으로 인하여 비정상적으로 두절될 수 있는 특징이 있는데, 연산의 실행 도중 통신이 두절되는 경우 데이터의 일부만 기록되거나 때로는 연산 실행의 결과를 확인조차 할 수 없는 경우가 발생한다[2]. 이로 인해 연산 실행의 불완전성(incompleteness) 및 태그 데이터의 불일치(inconsistency) 문제가 발생한다.

이 문제를 해결하기 위해 제안된 관련연구들은 RFID 태그 데이터를 물리적 저장소로 보는 관점과 분산 데이터베이스로 보는 관점으로 나눌 수 있다. 물리적 저장소 관점에서 볼 때 [3]은 통신 두절에 따른 태그연산의 불확실성(uncertainty)을 언급하고 이를 위해 태그 메모리에 대한 백업 저장소를 제안하였다. 하지만, 대량의 태그가 인식되는 경우 해당 태그들에 대한 모든 메모리 데이터 및 연산의 결과를 별도로 유지해야 하는 문제가 발생한다. 분산 데이터베이스 관점에서는 태그 연산 실행이 모바일 트랜잭션 수행과 유사한 측면이 있다. 모바일 단말은 데이터 캐시[4] 또는 복제[5]를 통해 통신 단절상태에서도 트랜잭션을 수행하며, 통신이 재개될 시 트랜잭션을 완료한다. 하지만, 전자태그의 경우 통신 두절 상태에서는 자체적으로 연산을 처리하지 못하므로, 모바일 트랜잭션의 기존 기법은 적용할 수 없다.

본 논문에서는 전자태그 메모리에 대한 여러 태그 연산이 동시에 실행되는 환경에서 태그의 일관성을 유지하면서 불완전하게 실행된 태그 연산을 재수행하기 위한 동시성 제어 프로토콜을 제안한다. 이 프로토콜은 잠금을 통해 불확실한 태그의 접근을 차단하는 대신, 연속질의 기법을 도입하여 미들웨어에 인식되는 모든 태그 식별 이벤트에 대한 모니터링을 수행하고, 불완전하게 실행된 연산을 재수행 함으로써 태그 접근의 일관성을 유지한다. 이 프로토콜의 가장 큰 특징은 불확실한 통신 환경에서 태그 메모리 접근의 정확성 및 직렬성을 보장함으로써 태그 메모리를 단순한 저장공간으로 인식하는 것에서 벗어나 분산환경에서의 태그 데이터베이스로 확장하는 것에 의의가 있다.

본 논문의 기술순서는 다음과 같다. 제2장에서는 전자태그의 메모리에 접근하기 위한 태그 데이터 접근 모델을 제시한다. 제3장에서는 태그 연산 실행의 동시성 제어를 위한 IOR프로토콜을 제안하고, 제4장에서는 정리를 통해 제안한 기법의 정확성과 일관성을 증명한다. 제5장에서는 실험을 통해 본 연구의 우수성을 입증하고, 마지막으로 제6장에서 결론을 기술한다.

2. RFID 태그 데이터 접근 모델

RFID 시스템은 데이터를 저장하고 있는 태그, 무선 주파수를 통해 주위의 태그를 인식하고 명령을 처리하기 위한 리더, 그리고, 다수의 리더들에 연결되어 사용자의 요청에 의해 데이터 접근을 처리하는 미들웨어로 구성된다. 이 논문에서 태그는 Gen2 태그[6], 즉 메모리를 가지고 있는 수동형 전자태그로 정의한다. 그리고, 태그 연산은 태그 메모리에 저장된 데이터를 읽거나 쓰기 위한 읽기(Read), 쓰기(Wrirte)연산과, 태그 조작을 위한 잠금(Lock), 소멸(Kill)등의 연산으로 구분되는데, 이 논문에서 대상으로 하는 태그 연산은 태그 메모리 접근 연산이다. 이때, 태그 연산 o의 표현은 RFID 미들웨어 표준인 Application Level Events[7]에 기반하여 읽기의 경우 *read*(*tid*, *rid*), 쓰기의 경우 *write*(*tid*, *rid*)로 정의한다. 이때 *tid*는 태그의 식별자로서 연산이 실행될 데이터 아이템을 의미하고, *rid*는 리더의 식별자로서 태그를 인식할 리더를 의미한다.

태그 연산의 실행은 다음과 같은 순서를 가진다. 사용자로부터 요청받은 태그 연산은 미들웨어에 등록되고, 리더는 무선 주파수를 이용하여 리더에서 인식되는 태그 목록을 보고한다[6]. 미들웨어는 태그 목록을 검사하여 태그 연산의 대상이 되는 태그의 인식을 확인하고, 해당 태그 연산을 실행을 리더에 요청한다. 리더들에 의해 인식되는 태그들은 지속적으로 미들웨어에게 보고되므로 아래와 같이 태그 스트림 및 태그 연산의 실행을 정의한다.

정의 1. RFID 태그 이벤트 스트림 $E=\{e_1, e_2, \cdots\}$는 태그 인식 이벤트 e_i의 연속적인 스트림으로, 이때 e_i는 태그를 인식한 리더, 인식된 태그의 식별자 및 이벤트 발생시간, 즉 $<rid, tid, t>$로 정의한다.

정의 2. 태그 연산 o_i의 실행 $Ex(o_i)$는 o_i와 태그 인식 이벤트 e_j의 맵핑, 즉 $M(o_i, e_j)$로 정의한다. 이때, e_j는 $o_i.tid = e_j.tid$, $o_i.rid = e_j.rid$를 만족해야 한다.

태그 연산은 연산이 요청되었을 때 즉시 실행되지 않고 대상 태그가 지정한 리더에서 인식되었을 때 실행된다. 그러므로 태그 연산의 실행 순서는 태그 인식 이벤트의 순서에 따라 달라진다. 예를 들어 그림 1과 같이 두 연산 o_1, o_2가 미들웨어에 등록되었다고 가정할 때, $E_1 = \{e_1 = <Tag\#1, Reader\#A>, e_2 = <Tag\#1, Reader\#B>, \cdots\}$의 경우 연산의 실행은 $Ex(o_1) \rightarrow Ex(o_2)$, 즉 $M(o_1, e_1) \rightarrow M(o_2, e_2)$의 순서를 가진다. 만일 태그 이벤트 스트림이 $E_2=\{e_1=<Tag\#1, Reader\#B>, e_2 = <Tag\#1, Reader\#A>, \cdots\}$와 같다면 순서는 $M(o_2, e_1) \rightarrow M(o_1, e_2)$로 바뀌게 된다. 즉, 연산들은 서로 독립적이며, 실행의 순서는 태그 인식 이벤트의 순서에 영향을

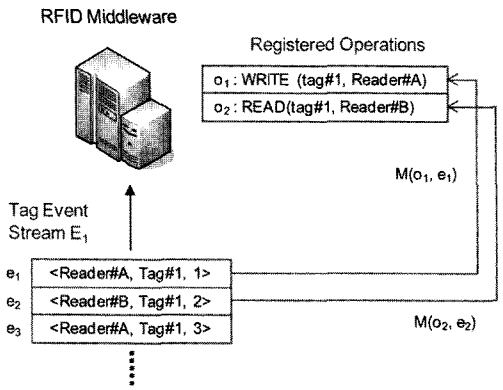


그림 1 태그 데이타 접근 모델 예시

받는다. 하지만, 태그 스트림 E_1의 경우 $M(o_1, e_1)$의 결과가 불완전한 경우 tag#1에 쓰기가 완료되지 않았으므로 tag#1이 불확실 태그가 되며, 이어지는 $M(o_2, e_2)$에서 o_2는 부정확한 데이터를 읽게 된다. 즉, 태그 연산이 불완전 실행된 경우에는 tag#1에 저장된 데이타가 불확실하고 비일관적이게 되므로, 연산 순서 제어를 통해 o_2 접근을 차단하고 재수행을 통해 o_1의 실행을 완료시키는 것이 필요하다.

3. 태그 연산의 동시성 제어 프로토콜

이 장에서는 태그연산의 부정확성에 따른 태그연산의 비일관성 문제를 해결하기 위한 동시성제어 프로토콜인 IOR(Incomplete Operation Reprocessing) 프로토콜을 제안한다. 이 프로토콜은 불완전 실행된 연산을 연속 질의(Continuous Query)로 정의하고 연속 질의 처리를 통해 태그연산의 정확성 및 일관성을 보장한다.

불완전 연산에 대한 연속 질의 처리를 위해 IOR 프로토콜은 불완전 연산 관리자 (Incomplete Operation Manager, IOM)를 포함한다. IOM은 불완전 연산에 대한 정보를 저장하고, 태그 인식 이벤트들과의 비교를 통해 해당 태그가 재인식되면 불완전 연산을 재수행하게 하는 역할을 수행한다. 불완전 연산의 정보를 관리하기 위해 IOM은 불완전 연산 테이블(IOT)를 포함하고 있는데, 이는 <*operation_id, tag_id, status*>의 형식으로 정의된다. 이때 *operation_id*는 불완전 연산의 식별자, *tag_id*는 불완전 연산의 대상 태그의 식별자이다. *Status*는 연산이 현재 실행 중인지를 저장하는 상태 플래그로써, 불확실 태그가 여러 리더에서 동시에 인식되는 경우 불완전 연산이 동시에 재수행되는 것을 막고 한 리더에서만 재수행을 하는 역할을 가지고 있다.

IOR 프로토콜의 수행순서는 다음과 같다. 먼저, 태그 연산 o_i가 연산의 실행 후 불완전하게 실행된 것으로 판정되면, REGISTER(o_i) 메시지를 통해 IOM에 등록한다. 이후 o_i는 대기상태에 머물러 있으면서 대상 태그인 $o_i.tid$의 재인식을 기다린다. 이때 IOM은 <o_i, $o_i.tid$, *false*> 레코드를 IOT에 추가한 후 $o_i.tid$를 연속 질의의 조건자로 정의하여 해당 태그의 재인식을 탐지한다. 이때 불완전 연산에 대한 연속 질의는 아래와 같이 정의된다.

정의 3. 불완전 연산 o_i에 대한 연속 질의 Q_i는 태그 스트림 E에 대하여 $o_i.tid = e_j.tid$를 만족하는 태그 인식 이벤트 e_j를 반환하는 질의이다.

IOM의 수행 메커니즘은 알고리즘 1에 기술되어 있다. *CQ_Processing* 함수는 연속질의를 처리하는 함수로서, *Search_IOT*를 통해 태그 인식 이벤트 e_j에 해당하는 연속질의 Q_i가 검색된 경우, OBSERVE(o_i, e_j) 메시지를 연산 처리기에 전달하여 대기상태의 o_i를 재수행하게 한다. e_j에 해당하는 연속 질의가 없는 경우에만 *SearchRegisteredOperation*을 통해 미들웨어에 등록된 다른 태그 연산들을 검색하고 이를 실행하는 방법으로 연산 실행 순서를 제어한다. *IO_Managing*은 불완전 연산을 등록하고 관리하는 함수로서, SUSPEND(o_i) 메시지는 불완전 연산의 재수행 결과가 여전히 불완전한 것으로 나올 때 태그의 재인식을 다시 기다리기 위해 대기 상태로 재진입하기 위한 메시지이다.

사용자가 요청한 태그 연산에 수행 시간이 포함된 경우, 태그 연산이 불완전하게 처리된 상태로 제한된 시간을 경과할 수 있다. 하지만, 시간이 경과하더라도 태그는 여전히 불확실한 정보를 저장하고 있으므로, IOR프

알고리즘 1 IOR 프로토콜의 수행 알고리즘

```
Algorithm IO_Managing
Begin
    while true do
        msg = receive_message();
        if msg is not null
            if msg is REGISTER then
                Add_IOT(msg.o.oid, msg.o.tid, false);
            Else
                if msg is SUSPEND then
                    r = Search_IOT(msg.o.tid)
                    r.status = false;
                else if msg is UNREGISTER then
                    remove_IOT (msg.o.tid)
            end if
        end if
    wend
end.

Algorithm CQ_Processing
Input E   /* tag event stream */
Begin
    foreach e in E do
        r = Search_IOT(e.tid)
        if r is found then
            if r.status is true then
                continue;
            else
                r.status = true
                Observe(r.oid, e);
            end if
        else
            SearchRegisteredOperation(e.rid, e.tid);
        end if
    end
end.
```

로토콜은 이 경우 불완전한 상태를 사용자에게 보고한 후 지속적으로 재수행을 진행한다. 그러므로 IOR 프로토콜은 실행 시간 제한이 있더라도 불완전 실행 시 제한에 상관없이 연산을 완료하는 특징이 있다.

4. 프로토콜의 정확성 및 일관성

불확실 태그의 일관성은 불완전 연산의 재수행시 재수행의 직렬성을 만족함으로써 유지될 수 있다. 이때 재수행의 직렬성은 정의 4와 같이 정의되며, IOR 프로토콜은 정리 1과 같이 연산의 직렬성과 태그의 일관성을 만족한다.

정의 4. 불완전 연산의 o_i의 직렬성은 $M(o_i, e_k)$가 불완전하게 실행된 후 이후 $M(o_i, e_m)$를 통해 재수행될때까지 그 사이에 실행된 모든 연산 c_j에 대하여 $o_i.tid \neq c_j.tid$를 만족하는 것으로 정의한다.

정리 1. IOR 프로토콜은 불완전 연산 o_i에 대한 재수행의 직렬성을 만족한다.

증명. $o_i.tid=c_j.tid$를 만족하는 c_j가 o_i의 재수행 도중에 실행되는 것은, $e_k.t < e_l.t < e_m.t$를 만족하는 태그 인식 이벤트 e_l에 대해 $M(o_j, e_l)$이 존재하는 것을 의미한다. 하지만, $e_l.tid = o_i.tid$를 만족하는 경우에는 IOR 프로토콜에 따라 o_i가 c_j에 앞서 수행되며 이때 $e_l = e_m$이 된다. 그렇게 되면 위 조건을 만족하는 $M(c_j, e_l)$은 존재할 수 없으므로, IOR프로토콜은 o_i의 재수행에 대한 직렬성을 만족한다. □

다음은 재수행을 통한 태그 연산 실행의 정확성이다. 연산 실행의 정확성은 불완전 실행된 연산이 재수행을 통해 완전하게 실행되는지의 여부를 통해 확인이 가능하다. 하지만, 실제 환경에서 전자태그는 물리적인 고장 및 훼손, 다른 사이트로의 이동 등의 이유로 미들웨어에 영구적으로 재인식되지 않을 가능성이 있다. 이때는 완전성 여부를 확인할 수 없으므로, 연산 실행의 완전성은 정의 5와 같이 정의한다. 이때 IOR 프로토콜이 이를 만족하는지는 정리 2를 통해 증명한다.

정의 5. 태그 연산 o_i는 태그 $o_i.tid$의 상태를 V_a에서 V_b로 바꾸는 것으로 가정한다. 태그 연산의 실행 $M(o_i, e_k)$ 이후 $c_j.tid = o_i.tid$인 c_j에 대해 $M(c_j, e_l)$이 실행될 때의 태그 상태가 V_b이면 o_i는 완전히 실행된 것으로 정의한다.

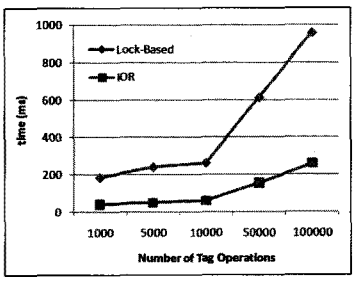
정리 2. IPT 프로토콜은 불완전 연산에 대한 완전한 수행을 보장한다.

증명. $M(o_i, e_k)$가 불완전하게 실행된 이후 IOR 프로토콜은 $o_i.tid = e_l.tid$를 만족하는 모든 이벤트 e_l에 대해 $M(o_i, e_l)$을 통해 o_i의 재수행을 시도한다. c_j는 o_i가 재수행된 시점 이후에 실행이 가능하므로, c_j가 실행되는 시점에서는 o_i의 재수행이 완료되어 태그 상태가 V_b가 됨을 보장한다. □

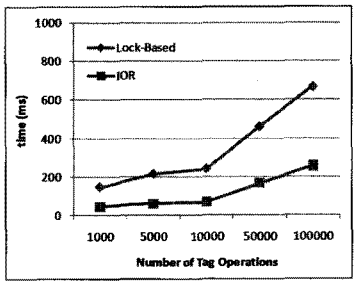
5. 실험 및 성능평가

IOR 프로토콜의 검증을 위해 본 논문에서는 RFID 미들웨어 시뮬레이션 모델을 통해 프로토콜의 성능을 평가하였다. 이 모델은 가상 리더로부터 인식되는 태그 인식 이벤트들에 대해 여과(Filtering)를 수행하여 등록된 태그 연산의 대상 태그로 확인된 경우 해당 연산을 실행하는 구조를 가지고 있다. 실험을 위해 본 논문에서는 두가지 데이터셋을 생성하였다. 첫번째는 시뮬레이션 모델에 등록되는 태그 연산 집합으로 균등분포를 가지도록 연산의 리더조건과 태그조건을 설정하였다. 두번째는 태그 이벤트 스트림으로 100k개의 태그 인식 이벤트로 설정하였다. 또한, 태그 연산의 불완전성을 모델링하기 위해 각 연산 실행시 연산이 불완전하게 실행될 가능성을 확률 변수 PI로 정의하였다. 본 시뮬레이션 모델은 자바 언어를 사용하여 구현하였으며, 실험은 펜티엄 Core2 2.1GHz, 2GB 메인 메모리를 가지는 윈도우 시스템에서 수행되었다.

그림 2는 태그 연산이 불완전하게 실행된 후 해당 연산이 재수행을 통해 완료될 때까지의 평균 시간을 측정한 결과이다. 비교 평가를 위해 동일한 환경에서 잠금기반(Lock-Based) 프로토콜의 성능을 함께 측정하였다. 이 프로토콜은 기본적인 동시성 제어 프로토콜로써 불완전 연산의 대상 태그에 대해 잠금 테이블을 구축함으로써, 불완전 연산이 아닌 다른 연산이 불확실 태그를 접근하는 것을 막고, 해당 리더에 태그가 재진입할 때



(a) Set PI to 1%



(b) Set PI to 10%

그림 2 불완전 연산의 재수행 시간 비교

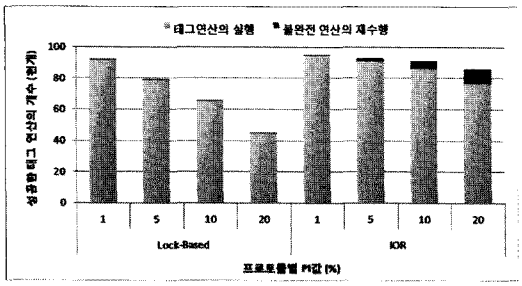


그림 3 성공적으로 실행된 태그연산의 개수 비교

재수행을 처리한다. 하지만, 이 프로토콜은 다른 리더에서 불확실 태그가 인식될 때 해당 태그로의 접근을 거부하는 문제가 있는데, 실험 결과 제안한 프로토콜이 기존의 잠금기반(Lock-Based) 프로토콜에 비해 70% 이상 처리시간을 단축시키는 것으로 나타났다. 이는 제안한 프로토콜은 어떤 리더가 재인식하더라도 불완전 연산을 재수행하기 때문이다. 또한 그림 2(a)와 그림 2(b)를 비교해볼 때 *PI* 값에 따라 잠금기반 기법은 수행속도의 차이가 나타났지만 IOR 프로토콜은 *PI* 값에 상관없이 유사한 재수행 시간을 보이고 있다.

그림 3은 태그 이벤트 스트림에 대해 성공적으로 수행된 태그 연산의 개수를 측정한 결과이다. 그림에서 보면 잠금기반 기법은 불완전 연산의 재수행을 통해 수행을 완료시킨 회수가 상당히 미미하며, 또한 불확실 태그의 잠금으로 인하여, *PI*가 커짐에 따라 전체적인 태그 연산의 수행 회수가 현저하게 감소함을 알 수 있다. 하지만, IOR 프로토콜은 *PI* 값이 커지는 만큼 태그 연산의 수행 성공 회수는 다소 줄어들지만 그만큼 재수행을 통해 보상하고 있음을 확인할 수 있다.

6. 결론 및 향후연구

수동형 전자태그는 태그에 부착된 메모리를 통해 태그를 부착한 제품에 대한 부가적인 데이터를 기록할 수 있다. 하지만, 데이터 접근시 발생하는 무선통신의 단절성, 불확실성 등으로 인해 태그 연산 실행의 완전성을 보장하지 못하며 이로 인해 연산의 부정확성 및 태그의 불일치 문제가 발생한다. 본 논문에서는 이를 해결하기 위하여 태그 연산의 실행 모델을 제시하고 태그 일관성을 유지하면서 불완전하게 수행된 연산을 재수행하기 위한 기법인 IOR 프로토콜을 제안하였다. 또한, 증명과 실험을 통해 제안한 프로토콜의 완전성 및 일관성을 증명하였다.

제안한 프로토콜의 가장 큰 특징은 불확실한 통신 환경에서 접근되는 태그 메모리에 대한 연산의 정확성 및 일관성, 직렬성을 보장함으로써 태그 메모리를 단순한 물리적 저장공간으로 인식하는 것에서 벗어나 분산 태그 데이터베이스로 확장하는 것에 의의가 있다. 향후 연구로는 본 연구의 확장을 통해 복수 개의 태그 및 연산으로 이루어지는 태그 트랜잭션의 정의 및 수행 모델을 제시하는 것이 필요하다.

참 고 문 헌

[1] J. Banks, D. Hanny, M. A. Pachano, L. Thompson, "RFID Applied," Wiley, Chichester, March, 2007, pp.328-329.

[2] EPCglobal Inc., "Low Level Reader Protocol (LLRP) Version 1.0.1," http://www.epcglobalinc.org/standards/llrp/llrp_1_0_1-standard-20070813.pdf, 2007.

[3] C. Floerkemeier, C. Roduner, M. Lampe, "RFID Application Development with the Accada Middleware Platform," IEEE Systems Journal, Special Issue on RFID, vol.1, no.2, pp.82-89, 2008.

[4] S. K. Madria, B Bhargava, "A Transaction Model for Mobile Computing," Int. Database Engineering and Application Symposium, pp.92-102, 1998.

[5] A. Rasheed, A. Zaslavsky, "Ensuring Database Availability in Dynamically Changing Mobile Computing Environments," In Proceedings of the 7th Australian Database Conference, Melbourne, Australia, 1996.

[6] EPCglobal Inc., "EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960 MHz. Version1.1.0," http://www.epcglobalinc.org/standards/uhfc1g2/uhfc1g2_1_1_0-standard-20071017.pdf, 2007.

[7] EPCglobal Inc. "The Application Level Events (ALE) Specification. Version 1.1," http://www.epcglobalinc.org/standards/ale/ale_1_1-standard-core-20080227.pdf, 2008.

류 우 석

1997년 부산대학교 컴퓨터공학과(공학사). 1999년 부산대학교 컴퓨터공학과(공학석사). 2002년~현재 부산대학교 컴퓨터공학과 박사과정. 관심분야는 RFID 미들웨어, ALE, 태그 데이터 모델, RFID 트랜잭션, RTLS

홍 봉 희

1982년 서울대학교 전자계산기공학과(공학사). 1984년 서울대학교 전자계산기공학과(공학석사). 1988년 서울대학교 전자계산기공학과(공학박사). 1987년~현재 부산대학교 컴퓨터공학과 교수. 관심분야는 RFID 미들웨어 데이터베이스, 실시간 위치정보 시스템, 유비쿼터스 미들웨어