

유비쿼터스 환경에서 EPCIS를 이용한 ID 추적 시스템 설계

신명숙[○] 송기범 홍성표 이정기 이철승 허진 이준
조선대학교 대학원 컴퓨터공학과
조선대학교 전자정보공과대학 컴퓨터공학과

{msshin[○], gbsong, goodman, jjang, cheolseung, Jhu, jlee}@chosun.ac.kr

Design of ID Tracking System using EPCIS in Ubiquitous Enviroment

Myeongsook Shin[○] Gibeom Song Seongpyo Hong Jeonggi Lee Cheolseung Lee Jin Hu Joon Lee
Dept. of Computer Engineering Graduate School, Chosun University
Dept. of Computer Engineering, Chosun University

요 약

유비쿼터스 환경에서 RFID는 제품을 인식하기 위해 물류분야를 중심으로 RFID 시스템을 도입하거나 적극 검토 중에 있다. EPC 네트워크 시스템은 EPC로 구분되는 모든 이벤트를 캡처하는 EPCIS를 가지고 있으며, 제품 정보를 포함한다. EPC 태그가 다른 곳으로 이동될 때마다, 리더는 태그를 읽으며, 태그 데이터가 EPCIS에 저장된다. EPCIS는 데이터 저장 시스템으로써 클라이언트가 서비스를 요구할 때 원하는 데이터를 제공한다. 모든 서비스가 웹 서비스를 통하여 EPCIS로 접근되며 EPCIS는 클라이언트로 하여금 제품의 실시간 상태를 얻을 수 있게 한다. 본 논문에서는 네트워크에서 실시간으로 제품 상태를 파악하기 위해서 EPCIS를 이용한 ID 추적 시스템을 설계하였다.

1. 서 론

유비쿼터스 컴퓨팅을 선도하는 차세대 핵심요소로 RFID(Radio Frequency IDentification)[1] 기술이 주목을 받고 있다. 특히 RFID 태그의 크기가 작아지고 가격이 저렴해지면서 모든 물건에 태그를 부착하여 물류 관리 분야의 큰 관심을 받고 있다. 이는 기존의 바코드 방식이 제품의 종류만을 인식할 수 있었던 것과 달리, RFID 방식은 제품을 개별적으로 인식할 수 있기 때문이다. 게다가 바코드는 가까운 거리에서만 인식이 가능했던 것에 비해 RFID는 먼 거리에서도 인식이 가능하다. 또한 바코드는 제품마다 개별적으로 인식한 반면 RFID는 동시에 여러 제품을 한꺼번에 인식할 수 있다.

본 논문에서는 네트워크에서 효율적으로 실시간 제품 추적을 위해 EPCIS(EPC Information Service)[2]를 이용한 ID 추적 시스템을 설계하였다. 이를 위해 EPC(Electronic Product Code)[3] 네트워크에 있는 각각의 EPCIS는 EPC 이벤트 데이터를 포함하며 이벤트 데이터는 주로 리더 ID, EPC, 확보된 각각 이벤트 타임을 가지고 있으며 리더 ID는 제품의 위치를 인식한다.

ID 추적 프로세스는 모든 위치에서 실시간으로 모든 제품의 상태를 유지하며, 주어진 EPC의 소유권을 가지고 있는 EPCIS를 찾기 위해 ONS(Object Naming System)와 접속한다. 소유권은 특정한 EPCIS에 주어진 EPC로써 최근 이벤트 확보를 의미한다. EPC를 리더가 읽을 때, 이벤트는 EPCIS에 저장된다. 그리고 EPC 소유권은 EPCIS로 교환되기 때문에 확보한 EPC의 마지막 이벤트를 가지고 있다. 그래서 ID 추적 프로세스가 ONS와 접속할 때, EPCIS 주소를 리턴 한다. 또한 ID 추적 시스템이 실시간으로 제품 검색을

위해서 로컬 데이터베이스를 조사하여 실시간으로 원하는 제품의 현재 위치를 얻는다.

본 논문의 구성은 다음과 같다. 2장은 관련 연구로서 RFID 시스템에 대한 전반적인 소개를 하고 3장에서는 ID 추적 시스템을 설계한다. 마지막으로 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. EPC 네트워크 시스템

모든 물건에 태그를 부착하여 관리하고자 하는 연구는 MIT, Cambridge를 중심으로 한 Auto-ID Center[4]와 일본의 Ubiquitous ID Center를 비롯해 이미 여러 곳에서 활발히 이루어지고 있다. 특히 Auto-ID Center의 연구는 최근 EPCglobal[5]로 이어져, Wal-Mart, Gillette, Hewlett-Packard 등 세계 유명 기업들과 연계되어 RFID 기술을 실제 물류에 적용하고 있다.

본 논문에서 제안한 ID 추적 시스템을 설계하기 위해 RFID, EPCIS에 대하여 알아본다.

2.1 RFID

RFID는 RF(Radio Frequency)를 이용하며, RFID 시스템은 리더, 즉 태그라고 불리는 트랜스폰더, 데이터를 처리하고 응용프로그램을 관리할 수 있는 로컬 서버로 구성된다. RFID 태그는 크게 액티브(Active) RFID와 패시브(Passive) RFID로 나눌 수 있으며 액티브 RFID는 태그 안에서 자체 전원장치를 설치하여 읽고 쓰기가 가능하며 재 작성 및 수정이 가능하다. 또한 메모리를 가지고 있어서 수신범위가 넓다. 하지만 사이즈가 커지고 비용이 많이 들며 태그 자체의 수명으로 인한 사용시간의

제한이 발생한다. 패시브 RFID는 가볍고 저렴하며, 수명이 길고 리더기로부터 에너지를 받아서 사용되므로 가시거리가 짧아지고 리더기에 상당히 많은 전력이 공급되어야 하는 점이 있다. 대부분의 읽기전용 태그는 패시브 태그를 사용하며, 32비트-128비트의 수정할 수 없는 정보가 프로그램 되어 있다.

이러한 기본적인 시스템 구성을 바탕으로 글로벌한 RFID 시스템을 구축하기 위해서 ONS, 검색 서비스(Discovery Service), EPCIS, Servant와 같은 추가적인 시스템이 필요하며 네트워크 시스템은 아래 (그림 1)과 같이 구성된다.

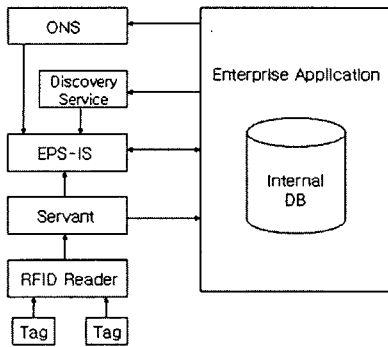


그림 1. EPC 네트워크 시스템 구성

EPC 네트워크 시스템은 주로 ONS, EPCIS, Savant, RFID 리더, 태그 등 5개의 구성요소로 구성된다. EPC 네트워크의 흐름은 RFID 태그에 저장된 EPC 정보를 RFID 리더로 읽어서 Savant가 EPC에 해당하는 관련 정보를 제공하는 PML(Physical Markup Language) 서버의 위치를 알려주는 ONS 서버에 요청하여 PML 서버의 IP 주소를 제공 받아 EPCIS에 저장한다. 그리고 정보를 검색하기 위해 ID 추적 시스템에 접속한다. 검색 서비스는 특정 EPC에 대한 EPCIS 포인터 리스트를 가지고 있으며 EPC 정보를 가지고 있다.

2.2 EPCIS

RFID 시스템은 트랜스폰더를 식별하기 위한 유일한 값으로 EPC를 사용한다. EPC 코드는 Auto-ID 시스템에서 기본이 되는 코드 체계로서 거의 무한한 자원이다.

EPC 코드는 아래 (그림 2)에서와 같이 네 부분으로 나누어지며, 각 부분의 역할은 다음과 같다. 'Header'는 버전을 나타내고 'EPC Manager'는 제품 생산자의 정보를 나타낸다. 'Object Class'는 품목, 즉 재고 보관 단위를 나타낸다. 'Serial Number'는 품목 중 제품의 고유 번호를 표시한다.

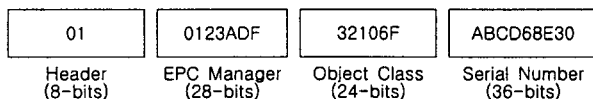


그림 2. EPC 코드 체계

리더가 전송 가능한 범위의 트랜스폰더에게 에너지와 정보를 주게 되면 트랜스폰더는 자신의 EPC 코드를 리더에게 전송하므로 자신의 정보를 전송하게 된다.

리더는 EPC 코드와 부가 정보를 읽어 들여 레거시 시스템의 자료로 저장된다. EPC 코드의 값이 유일한 값이기 때문에 읽혀졌던 데이터를 추적하면 태그가 부착된 제품의 이동 경로를 파악할 수 있다.

EPC 태그가 다른 위치로 이동할 때마다 리더가 태그를 읽으며 태그 데이터를 저장하기 위해서 사용자들에게 EPC에 대한 이벤트를 저장하고 있다가 클라이언트의 요청이 있으면 원하는 데이터를 검색하여 제공할 수 있는 EPCIS가 필요하다. 또한 EPCIS로 제공된 데이터는 웹 서비스를 통하여 사용 가능하다.

3. ID 추적 시스템 설계

3.1 ID 추적 시스템 시나리오

ID 추적 시스템은 리더에 의해 읽혀진 EPC 코드를 저장하여 실시간으로 제품 상태 파악과 재고 관리를 제공한다.

본 논문에서는 ID 추적 시스템을 위해 제품 ID로 재고 상태를 검색하는 시나리오를 가지고 EPCIS를 이용한 ID 추적 시스템을 설계한다. 또한 제품 ID로 재고 상태를 검색하는 ID 추적 시스템은 아래의 상황을 가정한다.

시스템 컨트롤러(System Controller)는 사용자가 선택한 제품을 모으기 위해 재고 관리자(Inventory Manager)에게 메시지를 보내며, 재고 관리자는 사용자가 선택한 저장소에 제품 양을 모으기 위해 EPC 관리자(EPC Manager)에게 메시지를 보낸다. 그리고 재고 관리자는 선택한 저장소에서 일치한 리더 리스트를 모으기 위해 데이터 컨넥터(Data Connector)로 메시지를 보내며, EPC 관리자는 선택된 저장소의 현재 EPC 리스트를 모으기 위해 EPCIS 컨넥터(EPCIS Connector)를 요청한다. 또한 재고 관리자는 제품과 저장소에서 타겟 양을 수집하기 위해 데이터 컨넥터를 요청하며, 저장소 위치로 제품 양을 모은다. 마지막으로 시스템 컨트롤러는 스크린에 보고서를 디스플레이 한다.

3.2 시스템 설계

본 논문에서 (그림 3)은 EPCIS를 이용한 ID 추적 시스템 설계도를 나타내며, ID 추적 시스템을 위한 함수를 정의한다.

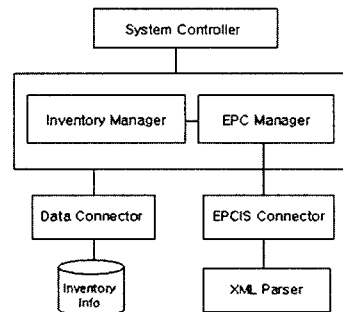


그림 3. ID 추적 시스템 설계도

시스템 컨트롤러는 GUI와 다른 구성요소 사이의 미들레이어를 생성하는 모든 기능성을 캡슐화하며 특별한 사용자 인터페이스로부터 독립적인 시스템을 만든다. 호출되는 `GetInventoryByStore(Store)` 함수는 현재 양과 타겟 양을 리턴하며 `GetInventoryByProductID(Product)`는 제품 ID, 제품 이름을 포함하는 보고서로써 각각의 저장소에서 현재 양과 타겟 양을 리턴한다. `AlertByProduct(Product)`는 요청한 제품에 대한 타겟 양의 위치와 리스트로 사용자에게 경고하며, `AlertByStore(Store)`는 현재의 양이 타겟 양으로 요청된 저장소에서 제품 리스트를 사용자에게 경고한다. 또한 `ViewProductLocation(EPC)`는 제품이 위치한 곳에서 디스플레이한다.

재고 관리자는 재고 비즈니스 논리를 캡슐화함으로써 타겟 양을 모으며 타겟 항목에 대하여 경고한다. `GetProductQtyByStore(Store)`는 현재 제품 양, 제품 ID, 위치를 리턴하며 `GetInventoryDetail(DataTable)`는 데이터를 확보하고 현재 양과 타겟 양 사이의 차이를 계산한다. `GetProductQty(Product)`는 요청된 제품의 현재 제품 양을 리턴한다.

EPC 관리자는 EPCIS로부터 리턴된 데이터를 조작하며, 현재 재고 양과 저장소에서의 총 제품을 모은다. `GetProductQty(Store)` 함수는 저장된 현재 제품 양을 모으며 `GetProductLocation(EPC)`는 요청된 제품의 현재 위치를 얻는다.

또한 EPCIS 컨넥터는 EPCIS에 대하여 웹 서비스 호출을 요청하는 모든 함수를 캡슐화하며 XML 형식에서 모든 입출력은 스트링으로 한다. `GetEventsByLocation(Reader)`는 리더의 위치를 확보하며 확보된 모든 이벤트를 리턴한다. 그리고 `GetEventsByEPC(EPC)`는 EPC를 확보하며 확보된 최근 이벤트를 리턴한다.

데이터 컨넥터는 데이터베이스 래퍼이며 데이터베이스와 직접적으로 상호 작용하는 모든 함수를 캡슐화한다. 이 함수는 데이터베이스에서 정보를 검색하는 저장된 프로시저를 호출한다. 그리고 리턴된 결과는 데이터 셋 형태이다. 함수로 `GetReaderList(Store)`는 저장소에 대해 표시된 리더 리스트를 리턴하며 `GetProductID(EPC)`는 제품 ID를 검색한다. 그리고 `GetStoreLocation(Reader)`는 리더 위치 명세를 리턴하고 `GetProductByEPC(EPC)`는 EPC의 제품 ID를 리턴한다. `GetEPCListByProduct(Product)`는 제품 ID로 EPC 리스트를 리턴하며 `GetTargetQty(Product, Reader)`는 리더가 제품의 타겟 양을 리턴한다. `GetModelByProduct(Product)`는 관련된 제품의 모델 리스트를 리턴한다. 또한 EPCIS 컨넥터가 EPC 정보 시스템에 접속하여 함수를 호출할 때, EPCIS에서 제공한 웹 서비스를 호출하여 EPC, 리더 ID, 확보한 이벤트 시간을 리턴한다. 그 다음에 확보한 리더 ID와 일치한 저장 위치를 검색하여 출력해주는 데 다음에 나오는 (그림 4)는 EPCIS에서 출력 스트링을 문법적으로 분석할 샘플 소스를 보여준다.

```
<observations>
<logEvent>
<location>
urn:epc:id:gl:900100.7296</location>
<observation>
<DateTime>2005-7-17T09:30:47</DateTime>
<Tag>
<ID>urn:epc:id:sgtin:900100.456.876</ID>
</Tag>
</observation>
</logEvent>
<logEvent>
<location>urn:epc:id:gl:900100.7297</location>
<observation>
<DateTime>2005-7-15T09:30:47</DateTime>
<Tag>
<ID>urn:epc:id:sgtin:900100.456.876</ID>
</Tag>
</observation>
</logEvent>
<logEvent>
<location>urn:epc:id:gl:900100.7298</location>
<observation>
<DateTime>2005-7-16T09:30:47</DateTime>
<Tag>
<ID>urn:epc:id:sgtin:900100.456.876</ID>
</Tag>
</observation>
</logEvent>
</observations>
```

그림 4. `GetEventsByEPC(XMLstring)` 소스

4. 결 론

본 논문에서는 네트워크에서 실시간으로 제품 상태를 파악하기 위해 EPCIS를 이용한 ID 추적 시스템을 설계하였다.

RFID는 제품을 인식하는 새로운 방법이 되고 있으며 RFID 기반 시스템은 바코드 시스템 문제에 대한 솔루션을 제공한다. 각각의 유일한 EPC는 각 제품에 대한 정보를 가지고 전송되며 리더가 코드를 판독할 때, 디지털 형식으로 전환되어 EPCIS에 저장된다. EPCIS는 ONS, 검색 서비스, 클라이언트 응용프로그램을 통해 찾을 수 있다. 이를 태먼 ID 추적 시스템은 EPCIS가 정보를 검색하기 위해 접속한다. EPCIS에서 리턴된 결과로는 EPC, 시간, 위치를 출력함으로써 실시간으로 제품 상태 파악을 제공한다.

이와 같이 ID 추적 시스템 설계를 통하여 사용자에게 제품에 대한 신뢰성을 제공하고 업무의 효율성을 향상시킬 수 있도록 하였다. 향후 연구 방향으로 ID 추적 시스템을 구현하고자 한다.

참고문헌

- [1] William P. Walsh, Research and application of RFID Technology to enhance aviation security, IEEE, 2000.
- [2] Mark Harrison, *EPC Information Service*, Cambridge Auto-ID Lab, Institute for Manufacturing, University of Cambridge
- [3] *Standard Specification (2004) EPC Tag Data Standards Version 1.1 Rev. 1.24*, EPC Global
- [4] David L. Brock, "The Virtual Electronic Product Code", Auto-ID Center Whitepaper, 2002.
- [5] Michael Mealling (2004), *EPCglobal Object Name Services (ONS) 1.0*, EPCGlobal