

# 유비쿼터스 환경에서의 물류관리를 위한 정보 서비스

김종욱<sup>0</sup>    정유나    황인준\*  
 아주대학교 정보통신전문대학원 데이터베이스 연구실, \*고려대학교 전자공학과  
 (kju, serazade}@ajou.ac.kr, ehwang04@korea.ac.kr

## Information Service for Logistics in Ubiquitous Environment

Jonguk Kim<sup>0</sup>    Youna Jung    Eenjun Hwang\*  
 Graduate School of Information and Communication, Ajou University  
 \*Department of Electronics and Computer Engineering, Korea University

### 요 약

RFID(Radio Frequency Identification) 기술이 발전하면서 다양한 분야에서 이를 활용하기 위한 연구들이 진행되고 있다. 특히 물류 관리 분야에서 실용화를 눈앞에 두고 있다. 본 논문에서는 물류 관리를 위한 정보 서비스 시스템을 제안하고자 한다. 물류 정보의 기록과 관리를 위한 정보 서비스 서버를 제안하고, 이질 서버간의 정보 교환을 위해 XML 기반의 언어 OML(Object Markup Language)을 제안하겠다.

### 1. 서 론

최근 RFID 태그(tag)가 크기가 작아지고 가격이 저렴해지면서 모든 물건에 태그를 부착하는 것이 가능하게 되었다. 모든 물건에 RFID 태그가 부착되면 이를 통해 물건 각각을 구분할 수 있다. 이 새로운 기술은 다양한 응용이 가능하지만 특히 물류 관리 분야의 큰 관심을 받고 있다. 기존의 바코드(bar-code) 방식이 제품의 종류만을 인식할 수 있었던 것과 달리, RFID 방식은 제품을 개별적으로 인식할 수 있기 때문이다. 게다가 바코드는 가까운 거리에서만 인식이 가능했던 것이 비해 RFID는 먼 거리에서도 인식이 가능하며, 물건마다 개별적으로 인식했던 것과 달리 RFID는 동시에 여러 물건을 한꺼번에 인식할 수 있다.

본 논문에서는 RFID를 이용한 물류관리 시스템을 구현하고 시스템을 위한 정보 서비스 서버를 제안하고자 한다. 이를 위해 정보의 표현과 교환을 위한 언어(OML)를 제안하고, 정보 서비스 서버를 설계하였다. 정보 서비스 서버는 생산자로부터 소비자까지 이어지는 유통 과정에 대한 정보를 저장하는 역할을 한다. 정보 서비스 서버는 기본적인 정보의 저장 외에도 정보 흐름의 오류를 감지하고 정보의 재전송을 요구한다. 이러한 정보 서비스를 통하여 제품을 구입한 소비자는 제품의 생산자 정보와 유통 과정을 언제 어디서든 열람할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 소개하고, 3절에서는 정보 서비스 서버의 구조와 동작 방식을 설명한다. 4절에서는 서버의 구현에 대해 설명한다. 마지막으로, 5절에서는 결론 및 향후 과제에 대해 논의한다.

### 2. 관련 연구

모든 물건에 태그를 부착하여 관리하고자 하는 연구는 MIT, Cambridge를 중심으로 한 Auto-ID Center[1]와 일본의 Ubiquitous ID Center[3]를 비롯해 이미 여러 곳에서 활발히 이루어지고 있다. 특히 Auto-ID Center의 연구는 최근 EPCglobal[4]로 이어져, Wal-Mart, Gillette, Hewlett-Packard 등 세계 유명 기업들과 연계되어 RFID기술을 실제 물류에 적용하고 있다.

본 논문에서 제안한 정보 서비스 서버는 Auto-ID Center

의 EPC와 PML을 기반으로 하고 있다. 2.1절에서는 각 사물을 구별하기 위한 식별자(ID)인 EPC(Electronic Product Code)[5]에 대해 살펴보고, 2.2절에서는 XML기반으로 사물을 표현하는 PML(Physical Markup Language)[2, 6]에 대하여 알아본다.

#### 2.1. EPC (Electronic Product Code)

사물을 구별할 수 있는 RFID 태그는 각각 다른 식별자를 가지고 있어야 한다. Auto-ID Center에서는 이 '유일한 식별자' 역할을 위해 EPC를 사용했다. EPC는 96-bit 식별자이다. 그림 1은 EPC Type 1이라고 불리는 초기 모델(model)이며 현재는 GID-96(General Identifier-96)으로 불리고 있다.

01	0123ADF	32106F	ABCD68E30
Header (8-bits)	EPC Manager (28-bits)	Object Class (24-bits)	Serial Number (36-bits)

그림 1. EPC Type 1

EPC Type 1은 4가지 부분으로 나누어지며, 각 부분의 역할은 다음과 같다.

- 헤더(Header): EPC는 기존에 정의 되어 있던 다른 코드들과의 호환도 고려하고 있으며[10], 헤더를 x 통해 코드들을 구분
- EPC Manager: 제품 생산자의(벤더, vendor) 정보
- Object Class: 한 생산자의 제품 종류를 구분하기 위한 부분
- Serial Number: 같은 종류의 개별 제품을 구분하기 위한 부분

#### 2.2. PML (Physical Markup Language)

Auto-ID Center에서 사물을 표현하기 위해 제안한 언어인 PML은 XML형식을 취하고 있으며, 작성 초기에는 모든 물질을 표현하기 위해 상당히 범용적인 스키마를 구상했다[11]. 그러나 현재는 RFID를 비롯한 각종 센서(sensor)를 통해 얻은 데이터를 교환하는데 이용하는 PML Core 1.0만이

표준으로 제안되어있는 상태이다.

### 3. 정보 서비스

#### 3.1. 정보 서비스 시스템

정보 서비스란 글자 그대로 사용자가 정보를 요구했을 때 알맞은 정보를 제공해주는 것을 말한다. 본 논문에서 제안하는 정보 서비스 시스템은 물류 관리를 위한 시스템이다. 즉, 물건이 생산되어 소비자까지 연결되는 과정에 관련된 정보를 유통 관계자와 소비자에게 제공하는 것이다.

물류 관리를 위한 정보 서비스 시스템의 구조는 그림 2와 같다. 여기에서 점선 화살표는 물류의 진행방향을 의미하며 실선 화살표는 데이터 전송을 의미한다. 시스템의 구조를 보면 정보 서비스 서버와 서버에게 물류 정보를 제공하는 기업(클라이언트), 그리고 서버에 질의하는 일반 사용자로 나눌 수 있다. 정보 서비스 서버가 중앙에 위치하고 있으며 생산자부터 소매상에 이르기까지의 모든 기업들이 클라이언트가 된다. 각각의 클라이언트는 자체적으로 정보를 데이터베이스에 저장하고 있다고 가정한다.

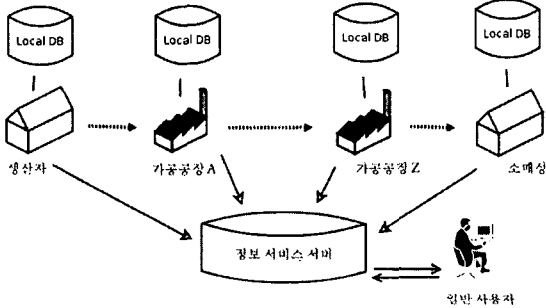


그림 2. 정보 서비스 시스템

클라이언트는 서버에게 자신이 처리한 물류 정보를 보내주는데 이러한 서로 이질적인 클라이언트의 정보 교환을 위한 통일적인 정보 서버를 구현하는 것은 어려운 일이다. 클라이언트들은 이미 자체적인 시스템을 가지고 있을 가능성이 높다(그림 3의 (a)). 그래서 본 논문에서는 OML (Object Markup Language)이라고 하는 새로운 언어를 제안하였다. OML은 XML 기반의 언어로서, 정보 서비스 시스템에서 데이터의 교환을 위해 이용된다. 클라이언트는 자신이 보유하고 있는 기존의 시스템에서 정보를 얻어내어 이를 OML로 표현하고, 이질 시스템간의 정보 교환을 수행한다. OML은 검증과 정보 전달의 목적을 가지고 있다.

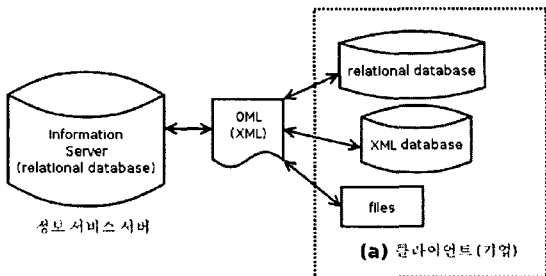


그림 3. 정보 서비스 서버 & 클라이언트의 데이터 교환

1) 정보 흐름 검증: 물류 유통에 대한 정보가 정상인지를 판단하기 위한 것이다. 클라이언트는 자체 생산된 정보를 보내기 이전에 검증 정보를 보낸다. 이후 서버로부터

흐름이 정상적이라는 메시지를 받으면 해당 정보를 OML 형태로 전송한다.

2) 정보 전송: 해당 유통 과정에서 생산된 추가적인 제품 정보와 유통 정보를 중앙의 정보 서비스 서버에게 전송한다. 서버는 이러한 추가 정보를 받아 해당 물건의 정보를 갱신한다.

유통 과정에 속해 있는 사업자뿐만 아니라 일반 사용자도 웹을 통해 물류 정보를 얻을 수 있다. 사용자가 정보 서비스 서버에 접속하여 물건의 EPC를 입력하면 해당 물건에 대한 생산 정보나 유통 정보를 볼 수 있다.

#### 3.2. OML (Object Markup Language)

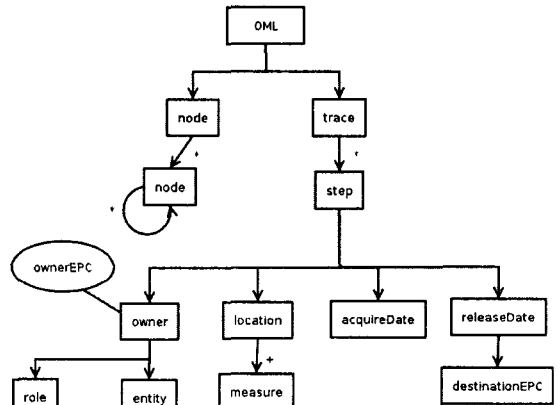


그림 4. OML 스키마 (핵심 요소)

Auto-ID Center에서 제안된 PML은 Wal-Mart와 같은 대형 유통 기업과 협약을 맺으면서 물류에 초점을 맞추었다. 그러나 이는 유통 과정이 정상적으로 이루어지는 경우에만 오류 없이 작동 된다. 하지만 유통 과정의 실수는 빈번하며 과정이 올바르게 하더라도 정보 전송 시의 네트워크 오류도 발생할 수 있다. 그러므로 본 논문에서는 이러한 상황을 감안하여 기존의 PML을 수정, 보완한 OML을 제안한다. 그림 4에서 OML 스키마의 핵심 엘리먼트 몇 가지를 표현해 보았다.

OML이 표현할 수 있는 정보는 다음과 같다.

- 제품의 포함관계를 표현한다. 예를 들어 A라는 물건이 B라는 용기에 포장된 상태.
- 제품의 유통 경로(사업자의 주소, 연락처, 위치등)를 표현한다.
- 제품의 입고(入庫) 시각과 출고(出庫) 시각을 표현한다.
- 출발지와 목적지의 정보를 표현한다. 목적지 정보는 제품이 목적지에 도달했을 때 올바르게 유통된 것인지 확인을 위해 사용된다.

### 4. 정보 서비스 서버

#### 4.1. 구현 환경 및 도구

구현된 정보 서비스 서버는 FreeBSD[7] 운영 체제를 기반으로 동작하며, 데이터베이스로 PostgreSQL[8]을 사용하였다. 사용된 언어는 Java이며, OML을 Java로 다루기 위해 Castor XML[9]을 사용하였고, PostgreSQL과의 연동은 JDBC와 Castor JDO[9]가 이용되었다. Castor JDO는 자바 클래스의 내용을 관계형 데이터베이스에 저장하고 추출할 수 있도록 한다.

정보 서비스 서버는 관계형 데이터베이스에 정보를 저장

한다. 그러나 외부의 이질 서버와 정보를 주고 받을 경우에는 범용적인 정보 표현 형식인 XML을 기반으로 하는 OML로 이루어진다. 이를 수행하기 위해서는 테이블 형태의 관계형 데이터와 XML간의 형태 변환이 자유롭게 수행되어야 한다. 구현된 정보 서비스 서버의 구조는 아래 그림 5와 같다.

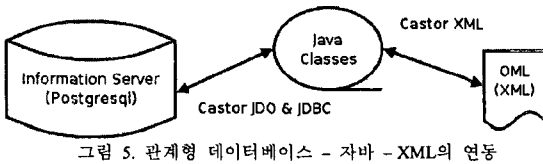


그림 5. 관계형 데이터베이스 - 자바 - XML의 연동

4.2. 시나리오

구현된 정보 서비스 서버는 아래의 상황을 가정한다.

- 1) 정보 서비스 시스템에서는 모든 제품이 EPC Type 1로 표현된다고 가정한다.
- 2) 클라이언트 ID 역시 EPC Type 1로 표현하며, Object Class와 Serial Number가 모두 1로 세팅된 것을 이용한다. 예를 들어 01.1234567.FFFFFFFF.FFFFFFFF가 가능하다.
- 3) 클라이언트의 인증 방법은 ID/Password 방식을 이용한다.

정보 서비스 서버는 다음과 같은 과정으로 하나의 클라이언트를 처리한다.

- 1) 클라이언트의 서버 접속 후 인증
  - 2) 정보 흐름 검증
  - 3) 2번 과정을 통과한 경우 클라이언트로부터 OML 형태의 정보를 받아 저장
- 다음은 OML의 한 인스턴스(instance)이다.

```
<OML>
  <node label="box" EPC="01.000142F.001C0F.000031928">
    <node label="apple" EPC="01.0016CA2.000104.000000591"/>
    <node label="apple" EPC="01.0016CA2.000104.000000592"/>
  </node>
  <trace>
</OML>
```

크게 물품정보(node)와 물류정보(trace)로 나뉘며, node정보는 반드시 하나 있어야하고 trace정보는 없어도 된다. trace정보가 없는 위와 같은 인스턴스가 3.1절에서 언급한 검증을 위한 OML이다. 위 OML은 상자에 두 개의 사과가 담겨있는 것을 표현하고 있다. 상자과 두 개의 사과가 가진 EPC를 서버에서 검색해본 후 모두 존재하지 않는다면 생산자라고 볼 수 있으며 정보 서버에 등록한다. 존재한다면 이 제품은 유통과정에 있다는 뜻이며 이 제품이 가지고 있는 가장 최근의 유통 정보를 얻어온다.

위 인스턴스가 생산자가 보내는 검증을 위한 OML이었다고 가정하고 상자를 트럭에 싣고서 다음 목적지로 보낸다고 하자. 생산자가 보낸 정보 OML의 예이다.

```
<OML>
  <node label="Truck" EPC="01.0053EF2.003FFC.000000A3">
    <node label="box" EPC="01.000142F.001C0F.000031928">
      <node label="apple" EPC="01.0016CA2.000104.000000591"/>
      <node label="apple" EPC="01.0016CA2.000104.000000592"/>
    </node>
  </node>
  <trace>
  <step>
    <owner ownerEPC="01.000A571.FFFFFFFF.FFFFFFFF">
      ...
    </owner>
  </step>
  <acquireDate>393840614</acquireDate>
```

```
<releaseDate
  destinationEPC="01.C4A984E.FFFFFFFF.FFFFFFFF">
  3956197748</releaseDate>
</step>
</trace>
</OML>
```

생산자의 EPC는 01.000A571.FFFFFFFF.FFFFFFFF이며 가공공장 A의 EPC는 01.C4A984E.FFFFFFFF.FFFFFFFF라는 뜻이다. 후에 트럭이 가공공장 A에 도착하면 가공공장 A는 트럭의 EPC와 트럭에 실려온 물품들의 EPC를 얻고 정보 흐름을 검증하기 위해 이를 정보 서비스 서버로 보낸다. 정보 서비스 서버는 해당 EPC에 대해 저장하고 있는 마지막 단계의 releaseDate 엘리먼트의 destinationEPC값과 현재 접속한 가공공장 A의 EPC가 일치하는지 확인함으로써 정상적으로 물류가 진행되고 있는지를 판단한다. 만약 생산자가 보낸 물건이 가공공장 A가 아닌 가공공장 Z로 전해졌다면, 혹은 가공공장 A가 보낸 정보가 유실된 상태로 가공공장 Z로 물건이 전해졌다면, 저장된 destinationEPC값과 일치되지 않으므로 서버가 해당 제품의 정보 흐름이 잘못되었다는 것을 알 수 있다. 서버가 잘못된 정보 흐름을 인식하면 해당 공장에 정보 갱신이 불가능함을 알립과 동시에 destinationEPC로 지정된 공장에 데이터를 요구하는 메시지를 보내어 잘못된 정보 흐름을 바로 잡게 한다. 이러한 수행 과정을 통하여 잘못된 유통 과정에도 비교적 단단한 정보 서비스를 구현할 수 있다.

5. 결론 및 향후 계획

본 논문에서는 물류 정보 관리를 위한 정보 서비스 서버를 구현하였다. 정보 서비스 서버는 유통 과정의 기록, 유통 과정에서 발생하는 오류 상황 감지, 그리고 일반 사용자에 유통 정보를 제공하는 역할을 한다.

하지만 텍스트(text)정보인 OML의 교환 시 내용을 어떻게 암호화할 것인가의 문제가 남아있다. 또한 일반 사용자들에게 물류 관련 정보를 모두 보여주는 것은 보안상 문제점이 있다. 기업 사이에서만 공유되어야 하는 내용도 얼마든지 OML에 포함 될 수 있기 때문이다. 그러므로 사용자에게는 정보의 일부만 보여주도록 정보를 숨기는 기능도 추가해야할 사항이다.

6. 참고 문헌

- [1] <http://www.autoidcenter.org/>
- [2] David L. Brock, "The Physical Markup Language," MIT Auto-ID Center Whitepaper, 2001.
- [3] <http://www.uidcenter.org/>
- [4] <http://www.epcglobalinc.org/>
- [5] David L. Brock, "The Virtual Electronic Product Code," Auto-ID Center Whitepaper, 2002.
- [6] David. L. Brock, Timothy P. Milne, Yun Y. Kang & Brendon Lewis, "The Physical Markup Language Core Components: Time and Place," Auto-ID Center Whitepaper, 2001
- [7] The FreeBSD Project: <http://www.freebsd.org>
- [8] PostgreSQL: <http://www.postgresql.org>
- [9] The Castor Project: <http://castor.exolab.org>
- [10] EPC Tag Data Specification Version 1.1 [http://www.epcglobalinc.org/standards\\_technology/EPCTagDataSpecification11rev124.pdf](http://www.epcglobalinc.org/standards_technology/EPCTagDataSpecification11rev124.pdf)
- [11] PML: The Physical Markup Language <http://web.mit.edu/mecheng/pml/>