

# 콜드체인에서 입증성 향상을 위한 Pedigree 시스템 설계†

엄지인\*, 김기홍\*\*, 홍봉희\*\*

\*부산대학교 물류 IT 협동과정

\*\*부산대학교 컴퓨터공학과

e-mail : jiin.eom@gmail.com, {buglist, bhhong}@pusan.ac.kr

## Pedigree System Design for Improving Authenticity in Cold-chain

Ji-in Eom\*, Gihong Kim\*\*, Bonghee Hong\*\*

\*Dept. of Logistics Information Technology, Pusan National University

\*\*Dept. of Computer Engineering, Pusan National University

### 요 약

콜드체인은 생산지로부터 소비자까지 저온상태로 상품을 신선하게 유지하여 안전하게 소비자에게 공급하는 것을 목적으로 한다. 그러나 유통 중 저온 유지가 안될 경우가 발생하고 이에 따라 변질된 상품을 소비자가 구매할 수 있다. 그러므로 소비자는 냉장 유통과정에 대해 입증할 수 있는 정보를 제공받기를 원한다. 따라서 본 논문에서 제안하는 Pedigree 시스템은 콜드체인 상품을 주고 받을 때 거래에 관한 정보인 Pedigree 데이터를 교환하는 시스템으로, 매 분배 단계에서 냉장 온도 및 물품의 상태를 체크하여 이상이 없을 경우에만 확인자의 서명정보를 추가하여 다음 분배지로 유통시키고 이상이 있을 경우 유통을 중지하여 변질된 상품이 유통되는 것을 막는다. 본 시스템의 아키텍처는 기존의 EPCIS(EPC Information Services)를 확장하여 사용하도록 설계하였고, Pedigree 처리를 위한 API 를 제공하는 Pedigree Library 를 제안하여 EPCIS 환경에 대하여 확장이 용이하다.

### 1. 서론

최근 소비자는 삶의 질이 향상됨에 따라 가격이 조금 높더라도 양질의 신선한 상품을 구입하길 원하고 있다. 이에 식품 유통을 추축으로 하여 콜드체인이 활성화 되어있다. 콜드체인은 생산지부터 소비자까지 냉장, 냉동의 저온상태로 상품을 신선하게 유지하여 양질의 상품을 안전하게 소비자에게 공급하는 것을 목적으로 하고 있다.

현재 콜드체인 시스템의 경우, 소비자는 상품 구매 시 저온상태로 신선하게 유지되어 유통되었는지 확인할 수가 없고, 믿고 구매할 수 밖에 없는 상태이다. 최근 소비자가 신선식품으로 믿고 구매하였으나 상품이 변질된 경우가 있었다. 따라서 콜드체인에 대하여 소비자의 불만은 커져가고 있고, 신뢰성은 떨어지고 있다. 이에 소비자는 냉장 유통과정에 대한 입증할 수 있는 데이터를 제공받기를 원하고 있다.

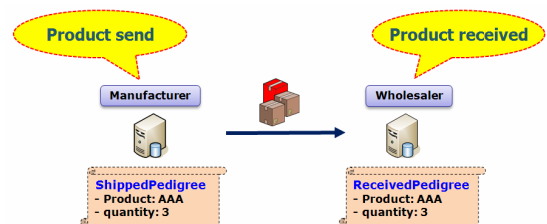
본 논문에서는 콜드체인 상품을 주고 받을 때, 거래에 관한 정보인 Pedigree 데이터를 교환하는 Pedigree 시스템을 제안하고자 한다. Pedigree 시스템은 매 분배 단계에서 냉장 온도 및 물품의 상태를 체크하여 이상이 없을 경우에만 확인자의 서명정보를 추가하여 다음 분배지로 유통하고, 이상이 있을 경우는 즉시 유통을 중지한다. 그러므로 본 시스템은 콜드체인 유통 시 Pedigree 정보를 가지는 인증 받은 안전한 상품만 유통이 가능하게 한다.

본 논문의 구성은 다음과 같다. 2 장은 관련연구로 Pedigree 와 EPCIS 연구를 기술하고, 3 장에서는 본 시스템의 대상환경 및 문제를 정의한다. 4 장은 문제를 해결하기 위한 Pedigree 시스템을 제안하고, 마지막으로 5 장은 결론 및 향후 연구에 대해 기술한다.

### 2. 관련연구

#### 2.1 Pedigree 연구

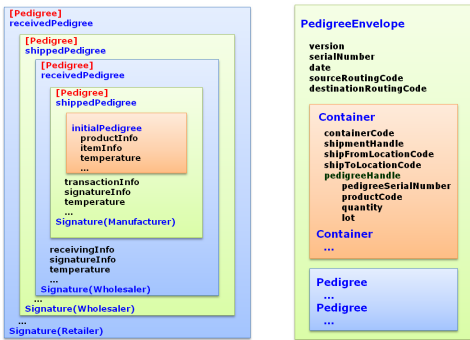
Pedigree 의 사전적 의미는 혈통 또는 가계라는 뜻이다. 본 시스템에서의 Pedigree 는 상품의 혈통, 즉 어느 곳에서 누구에 의해 생산되어 어디를 거쳐서 유통되었는지를 나타내는 상품 거래 정보이다. (그림 1) 과 같이 상품을 보낼 때 Pedigree 정보를 함께 보내고, 물건을 받을 때 받은 것을 확인하는 Pedigree 정보를 남김으로써 상품의 유통 경로를 파악할 수 있고, Pedigree 를 사용함으로써 인증된 상품만 유통된다.



(그림 1) 상품 거래 시 생성되는 Pedigree 정보

† “이 논문은 2009 년 교육과학기술부로부터 지원받아 수행된 연구임” (지역거점연구단육성사업/차세대물류 IT 기술연구사업단)

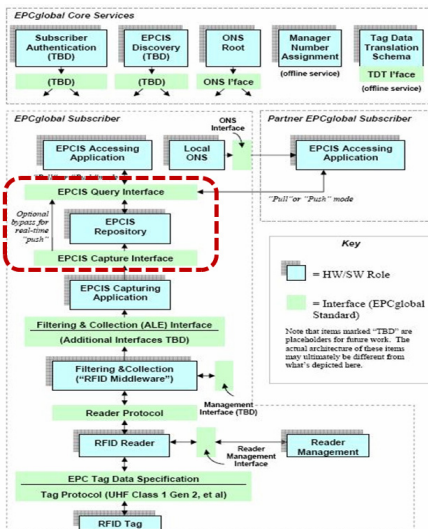
Pedigree 정보는 (그림 2)와 같이 Pedigree 와 Pedigree Envelope 의 두 가지의 형태로 나타낼 수 있다. Pedigree 는 상품, 트랜잭션, 분배자, 수령자, 서명 등의 정보를 포함한다. 전 단계에 생성된 Pedigree 정보는 다음 유통단계에서 그대로 유지되고, 현재단계의 트랜잭션 정보와 서명정보를 추가한다. Pedigree Envelope 은 하나 또는 그 이상의 Pedigree 를 감싸서 거래 파트너에게 보낼 때 사용한다. EPCglobal 에서는 데이터 구조의 표준으로 Pedigree 를 제시한 바 있다.[1] 하지만 이 Pedigree 는 현재 미국 법안에 맞추어진 Drug Pedigree 로 의약품에 맞추어진 데이터 구조를 가진다.



(그림 2) Pedigree 및 Pedigree Envelope 포맷

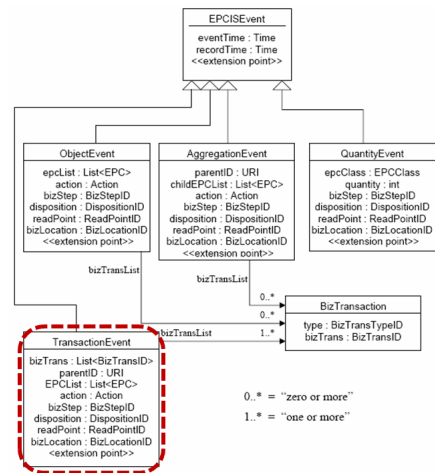
Pedigree 의 요구사항은 크게 세 가지 이다. 첫째, 상품에 대한 추적이 가능해야 한다. 공급망에서 상품을 주고 받을 때 정보를 기록함으로써 상품의 이동경로 추적을 가능하게 할 수 있다. 둘째, Pedigree 데이터를 저장하고 관리할 수 있어야 한다. 각 분배지에서 Pedigree 정보를 정확하고 안전하게 기록이 되도록 유지하고 또한 높은 신뢰도를 가진 Pedigree 정보를 관리하고 리포팅 할 수 있어야 한다. 셋째, 파트너사 간에 Pedigree 데이터 교환이 가능해야 한다. 파트너사는 Pedigree 정보를 공유함으로써 안전한 공급망을 유지할 수 있게 된다.

2.2 EPCIS 연구



(그림 3) EPCglobal 네트워크 아키텍처

EPCglobal 은 (그림 3)과 같이 표준 네트워크 아키텍처를 제시하고 있다.[2] RFID 태그를 읽는 RFID 리더, RFID 데이터를 필터링하고 모으는 ALE(Application Level Event), 비즈니스 이벤트를 생성하는 EPCIS Capturing Application, EPCIS 레벨의 이벤트를 저장하고 검색하는 EPCIS 등으로 구성된다. 이 중 EPCIS 는 EPC Information Services 로 EPC 정보 서비스의 표준이다. EPC 정보를 나타내는 이벤트는 (그림 4)와 같이 크게 4 가지로 분류된다. 기본적인 EPC 상품의 발견을 나타내는 Object 이벤트, 상품의 포함관계에 관한 정보를 나타내는 Aggregation 이벤트, 상품의 수량 정보는 나타내는 Quantity 이벤트, 상품의 거래 정보를 나타내는 Transaction 이벤트가 있다.



(그림 4) EPCIS 이벤트

EPCIS 이벤트에 Drug Pedigree 정보를 저장하는 방식은 SupplyScape[3]와 IBM[4]에서 제안한 바 있다. Pedigree 정보를 나타내기 적합한 이벤트는 위의 4 가지 이벤트 중 상품 거래에 관한 정보라는 공통된 특징을 가지는 Transaction 이벤트이다. 따라서 Transaction 이벤트의 Extension 포인트를 활용하여 Pedigree 정보를 저장할 수 있다.

3. 대상환경 및 문제정의

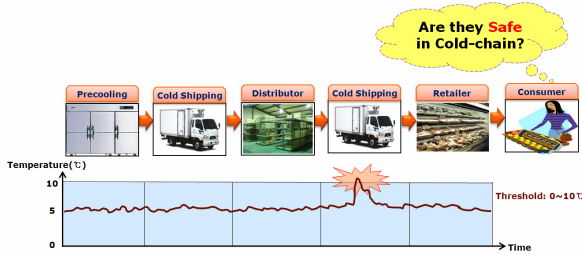
3.1 대상환경

본 연구의 대상환경은 콜드체인 시스템이다. 콜드체인은 상할 수 있는 상품, 예를 들어 식품,약품, 혈액 등에 대하여 생산지부터 소비지까지 일정한 냉장 온도로 빈틈없이 이동하여 생산품의 질과 안전성을 유지하는 시스템이다. 그리고 콜드체인 상품의 생산지와 각각의 분배지는 EPC 정보 서비스를 하기 위하여 각각의 EPCIS 를 가지고 있다.

3.2 문제정의

생산지에서 소비지까지 이동하는 동안 (그림 5)와 같이 콜드체인의 저온 상태가 유지되지 않는 상황이 발생할 수 있다. 예를 들면, 냉장 트럭의 고장, 상온에서의 상품 이관 작업 및 상품 방치, 또는 냉장 진

열대의 허술한 온도관리로 인해서 저온 상태가 유지되지 못할 수 있다. 소비자는 진열되어 있는 냉장 상품을 보고도 과연 콜드체인 유통과정에서 아무 문제가 없었는지에 대한 의문을 가질 수 있다.



(그림 5) 저온상태가 파괴된 콜드체인

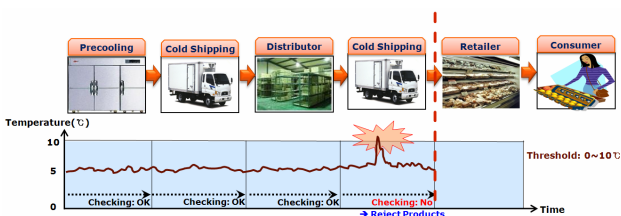
콜드체인을 거쳐온 신선식품이라고 소비자가 믿고 구매하였으나 (그림 6)의 기사에서처럼 변질된 식품으로 발견된 경우가 있다. 또한 신선하지 못한 상품을 덩이나 떨이로 판매하는 경우도 허다하다. 이로 인하여 신선식품에 대한 신뢰성이 상실되었고, 소비자들의 불만도 증대되었다. 소비자는 냉장유통 과정에서 안전하게 잘 운송해 왔다는 것을 확인하기 위한 믿을 만한 정보를 제공받기 원한다. 냉장유통 과정에서 분배되는 동안 누가 확인을 했는지, 어느 지점을 어느 시간에 지나왔는지, 유통 도중의 온도 변화는 어떠한지 등의 정보를 제공받는다면 콜드체인에 대한 신뢰성이 회복될 수 있을 것이다.



(그림 6) 신선식품 불신 관련 기사(경향일보-2008.3.27)

#### 4. Pedigree 시스템

##### 4.1 Pedigree 시스템 개요

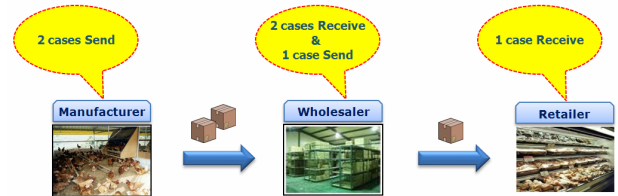


(그림 7) Pedigree 시스템이 적용된 콜드체인

Pedigree 시스템은 (그림 7)과 같이 콜드체인에서 매 분배 단계에서의 냉장 온도를 체크하여 제한 온도를 넘어선 경우 다음 단계에서 상품을 받는 것을 거절하고, 제한 온도를 넘어서지 않고 상품에 이상이 없을 경우 상품을 받는다. 이 때 상품과 온도에 대해 안전하다는 것을 확인한 후 확인자의 서명정보를 남

긴다. 이렇게 매 단계마다 상품을 안전하게 주고 받음을 확인하는 데이터를 Pedigree 라고 한다. 이러한 Pedigree 정보가 없는 상품의 경우 믿을만한 상품이 아니고 유통 또한 거부된다.

##### 4.2 Pedigree 시스템 시나리오



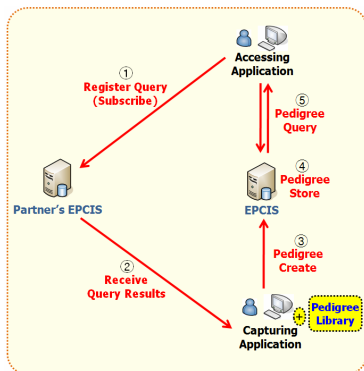
(그림 8) Pedigree 시스템 시나리오

(그림 8)과 같이 생산지에서 도매상을 거쳐 소매상으로 상품이 이동할 경우 EPCIS 이벤트와 연계하여 Pedigree 생성 시나리오를 살펴보면 다음과 같다.

- Pre-Step: 파트너 사에 EPCIS 연속질의를 등록하여 상품의 Shipping 및 Receiving 발생 시 Transaction 이벤트 데이터를 자동으로 교환하도록 한다.
- Step 1: 생산지에서 냉장상품에 태그를 부착하면 각 상품에 대한 Object 이벤트가 발생하고, 상품의 기본 정보와 생산 정보를 포함한 Initial Pedigree 가 생성된다. 이 정보는 생산지의 EPCIS 에 저장된다.
- Step 2: 생산지에서 냉장케이스에 태그를 부착하면 각 케이스에 대한 Object 이벤트가 발생하고 이 정보는 생산지의 EPCIS 에 저장된다.
- Step 3: 생산지에서 냉장상품을 케이스에 Packing 하면 Aggregation 이벤트가 발생하고 이 정보는 생산지의 EPCIS 에 저장된다.
- Step 4: 생산지에서 냉장 팔렛에 태그를 부착하면 각 팔렛에 대한 Object 이벤트가 발생하고 이 정보는 생산지의 EPCIS 에 저장된다.
- Step 5: 생산지에서 냉장케이스를 팔렛에 Packing 하면 Aggregation 이벤트가 발생하고 이 정보는 생산지의 EPCIS 에 저장된다.
- Step 6: 냉장 DC 에 팔렛이 도착하면 팔렛에 대한 Object 이벤트가 발생하고, 이 정보는 생산지의 EPCIS 에 저장된다.
- Step 7: 냉장 DC 에서 팔렛에 대한 Unpacking 작업을 하면 케이스에 대한 Object 이벤트가 발생하고 이 정보는 생산지의 EPCIS 에 저장된다.
- Step 8: 생산자는 도매상의 주문에 따라 냉장트럭에 물건을 싣고 도매상으로 운송한다. 이 때 Transaction 이벤트가 발생하고 Transaction 이벤트의 Extension Point 에 Pedigree Envelope 을 포함한다. Pedigree Envelope 은 Step 1 에서 생성된 Initial Pedigree 에 현재 단계의 Shipped Pedigree 와 전자서명 정보를 포함하여 생산지의 EPCIS 에 저장된다.
- Step 9: 생산지의 Step 8 에서 생성된 Transaction 이벤트는 Pre-step 단계에서 등록해 놓은 연속질의에 따라 해당 도매상으로 전송되어 물건을 보냈음을 알려준다.

- Step 10: 도매상은 생산자로부터 물품을 받고 확인한다. 이 때 Transaction 이벤트가 발생하고 Transaction 이벤트의 Extension Point 에 Pedigree Envelope 을 포함한다. Pedigree Envelope 은 Step 1 에서 생성된 Initial Pedigree, Step 8 에서 생성된 Shipped Pedigree 에 현재 단계의 Received Pedigree 와 전자서명 정보를 포함하여 도매상의 EPCIS 에 저장된다.
- Step 11: 도매상의 Step 10 에서 생성된 Transaction 이벤트는 Pre-step 단계에서 등록해 놓은 연속질의에 따라 물건을 보내준 생산지로 전송되어 물건을 안전하게 받았음을 알려준다.
- Step 12: 도매상은 소매상의 주문에 따라 냉장트럭에 물건을 싣고 소매상으로 운송한다. 이 때 Transaction 이벤트가 발생하고 Transaction 이벤트의 Extension Point 에 Pedigree Envelope 을 포함한다. Pedigree Envelope 은 Step 1 에서 생성된 Initial Pedigree, Step 8 에서 생성된 Shipped Pedigree, Step 10 에서 생성된 Received Pedigree 에 현재 단계의 Shipped Pedigree 와 전자서명 정보를 포함하여 도매상의 EPCIS 에 저장된다.
- Step 13: 도매상의 Step 12 에서 생성된 Transaction 이벤트는 Pre-step 단계에서 등록해 놓은 연속질의에 따라 해당 소매상으로 전송되어 물건을 보냈음을 알려준다.
- Step 14: 소매상은 도매상으로부터 물품을 받고 확인한다. 이 때 Transaction 이벤트가 발생하고 Transaction 이벤트의 Extension Point 에 Pedigree Envelope 을 포함한다. Pedigree Envelope 은 Step 1 에서 생성된 Initial Pedigree, Step 8 에서 생성된 Shipped Pedigree, Step 10 에서 생성된 Received Pedigree, Step 12 에서 생성된 Shipped Pedigree 에 현재 단계의 Received Pedigree 와 전자서명 정보를 포함하여 소매상의 EPCIS 에 저장된다.
- Step 15: 소매상의 Step 14 에서 생성된 Transaction 이벤트는 Pre-step 단계에서 등록해 놓은 연속질의에 따라 해당 도매상으로 전송되어 물건을 안전하게 받았음을 알려준다.

### 4.3 Pedigree 시스템 아키텍처



(그림 9) Pedigree 시스템 아키텍처

Pedigree 시스템의 아키텍처를 나타내면 (그림 9)와

같다. 먼저 파트너 사의 Pedigree 정보를 조회하기 위해서 파트너 사의 EPCIS 에 연속질의를 등록하여 파트너 사의 Pedigree 정보 조회가 가능하도록 설정한다. Capturing Application 에서 파트너 사의 Pedigree 정보를 받아서 현재 단계의 Pedigree 정보와 서명정보를 추가하여 Pedigree 가 포함된 Transaction 이벤트를 생성한다. 생성된 비즈니스 이벤트는 EPCIS 에 저장되고, Accessing Application 을 통해서 Pedigree 정보를 조회한다.

파트너 사의 Pedigree 정보를 조회하고, 현재 단계의 Pedigree 정보 및 서명정보를 추가하는 역할은 Pedigree Library 에서 담당하며, Pedigree Library 는 Pedigree 처리를 위한 API 를 제공한다. Pedigree Library 를 이용하면 기존 EPCIS 시스템에 Pedigree 확장을 하기 쉽다. Capturing Application 의 경우 환경에 의존적 이어서 응용 별로 다른 형태를 가진다. 하지만 이러한 Capturing Application 에 Library 형태로 Pedigree 처리를 위한 API 를 제공한다면 Pedigree 데이터 처리를 위한 확장 부분을 쉽게 추가할 수 있다. <표 1>은 Pedigree Library 의 API 를 보여준다.

<표 1> Pedigree Library API

| API Name               | Input Param Name  | Return Value     | Role  |
|------------------------|---|------------------|---|
| CreateInitialPedigree  | productName, manufacturerName, productCode, expirationDate, quantity, itemserialNumber... | InitialPedigree  | Create InitialPedigree Element                          |
| CreateShippedPedigree  | sender, receiver, senderAddress, receiverAddress, productInfo, itemInfo...                | ShippedPedigree  | Create ShippedPedigree Element                          |
| CreateReceivedPedigree | sender, receiver, senderAddress, receiverAddress, productInfo, itemInfo,...               | ReceivedPedigree | Create ReceivedPedigree Element                         |
| CreateSignature        | certificateFile, Password   | Signature        | Create Signature Element                                |
| CreatePedigree         | InitialPedigree, ShippedPedigree, ReceivedPedigree, Signature                             | Pedigree         | Create Pedigree Element                                 |
| CreatePedigreeEnvelope | Container, Pedigree   | PedigreeEnvelope | Create PedigreeEnvelope Element                         |
| ReceivePartnerPedigree | QueryResults  | PartnerPedigree  | Receive Partner's Pedigree from Continuous QueryResults |

### 5. 결론 및 향후 연구

콜드체인 환경에서 소비자는 저온 유통을 보장하는 신뢰할 수 있는 정보를 제공받기 원한다. 이를 위해서 본 논문에서는 Pedigree 시스템을 제안하였고 기존의 EPCIS 를 확장하여 사용하는 아키텍처를 제시하였다. 이 아키텍처는 Pedigree Library 를 사용하여 Pedigree 데이터 처리를 위한 API 를 제공함으로써 기존 EPCIS 환경에 대하여 확장이 용이하다.

향후 연구로써 각각의 Capturing App.에서 Pedigree 를 생성하지 않고, 모든 EPCIS 에 대해서 Pedigree 를 생성하고 유통하는 서비스를 담당할 시스템에 대한 연구가 필요하다.

### 참고문헌

- [1] EPCglobal, "Drug Pedigree Ratified standard v1.0", January, 2007
- [2] EPCglobal, "EPCIS Standard v.1.0.1", September, 2007
- [3] SupplyScape, Combining EPCIS with the Drug Pedigree Messaging Standard, January 11, 2008
- [4] AmerisourceBergen and IBM, Pharma Solution Guide for EPCIS Pedigree Compliance, Working Draft Version of 07 November 2007