

EPC Sensor Network 에서의 데이터 처리와 정보 공유를 위한 미들웨어

이경태^o, 임장관, 성종우, 김성훈, 하민근, 김대영

한국과학기술원(KAIST)

{handreic, limg00n, jwsung, shkim08, minkeun.ha, kimd}@kaist.ac.kr

A Middleware for Data Processing and Information Sharing in EPC Sensor Network

Gyungtae Lee^o, Janggwan Im, Jongwoo Sung, Seong Hoon Kim, Minkeun Ha,

Daeyoung Kim

Korea Advanced Institute of Science and Technology

요 약

RFID는 각 사물들의 글로벌한 식별을 가능하게 하고 RFID에서 사실 상의 표준인 EPCglobal Architecture Framework는 RFID 데이터를 처리하기 위한 프레임워크로서 만들어졌다. 하지만 EPCglobal Architecture Framework는 RFID 데이터만을 위해서 디자인된 아키텍처이므로 사물들의 주변 환경 정보와 같은 데이터를 제공하는 센서 네트워크에 대해서는 고려하지 않았다. 반대로 센서 네트워크의 경우 다양한 센서 네트워크가 존재하면서 하나의 통일된 규격이 존재하지 않는다. 이를 보완하기 위해 RFID와 센서 네트워크를 통합한 미들웨어 프레임워크인 EPC Sensor Network가 제안되었다. 본 논문에서는 EPC Sensor Network 연구의 일환으로 Extended ALE와 Extended EPCIS를 제안한다. Extended ALE는 복합 이벤트 처리(Complex Event Processing, CEP)를 이용한 이벤트 처리 컴포넌트로 EPCglobal Architecture Framework에서 센서 데이터의 처리에 사용될 수 있다. Extended EPCIS는 RFID 및 센서 데이터의 공유 및 제공을 위한 분산 데이터베이스 시스템이다. 본 논문에서는 식품 안전 시나리오를 기반으로 하는 프로토 타입을 구현함으로써 제안된 미들웨어의 디자인을 검증한다.

1. 서 론

Radio Frequency Identification (RFID)는 사람의 도움 없이 사물의 식별을 가능하게 하는 기술로 EPCglobal Architecture Framework[2] 또는 EPC network는 RFID에 있어서 사실 상의 표준이며 하드웨어/소프트웨어 컴포넌트와 그들 간의 인터페이스에 대해서 정의하고 있다. EPC network에서는 Electronic Product Code (EPC)라는 글로벌한 식별체계를 사용함으로써 각 사물들의 구분을 할 수 있게 한다. EPC network에서는 RFID를 사용함으로써 각 사물들을 인식할 수는 있으나 사물들의 주변 환경에 대한 정보를 처리할 수 없기에 실제 산업에서 좀 더 유용하게 사용하기 위해서는 이에 대한 보완이 필요하다.¹

반대로 센서 네트워크의 경우 다양한 센서들을 통해서 온도, 습도 등의 주변 환경에 대한 정보를 읽고 무선 네트워크 등을 통해서 전송, 원격에서 저장하고

처리할 수 있는 기술이기에 EPC network를 보완할 수 있는 기술로서 각광을 받고 있다. 하지만 센서 네트워크의 경우 응용에 따라서 그에 맞춰진 다양한 센서 네트워크들이 사용되면서 하나의 통일된 규격이 존재하지 않아 다른 센서 네트워크 간의 통신이 어렵다.

이 두 가지 기술을 통합함으로써 서로를 보완할 수 있기에 RFID와 센서 네트워크를 통합한 미들웨어 프레임워크인 EPC Sensor Network(EPCSN)[1]가 제안되었고 본 논문은 그 중에서도 RFID 및 센서 데이터의 처리와 정보 공유를 위한 Extended ALE 및 Extended EPCIS를 제안한다.

EPC network는 RFID 데이터 처리만을 지원해주고 있고 이를 위해서 Filtering and Collection (F&C) 미들웨어와 Application Level Events(ALE) [3] 인터페이스를 정의하고 있다. F&C의 역할은 RFID 데이터의 필터링하고 그룹으로 묶어 ALE 인터페이스에서 정의된 패턴에 따라서 응용 레벨 이벤트를 생성하는 것이다. 하지만 ALE 인터페이스는 RFID 데이터 처리만을 위해 디자인 되었기에 주어진 패턴들은 매우 제한되어 있다. F&C에서는 센서

본 연구는 지식경제부의 지원을 받는 정보통신표준기술력향상사업의 연구결과로 수행되었음

데이터의 처리를 위해서 필요한 각종 산술/논리 연산 역시도 지원하지 않는다.

EPC network는 RFID와 관련된 정보의 저장 및 공유만을 지원한다. EPC Information Service(EPCIS)[4]는 표준 인터페이스를 통해서 사용자간의 EPC와 관련된 데이터의 교환을 가능하게 한다. EPCIS는 이벤트 데이터와 마스터 데이터라는 두 가지 타입의 데이터를 다룰 수 있다. 이벤트 데이터는 비즈니스 프로세스 도중에 벌어지는 이벤트를 나타내며 EPCISEvent, ObjectEvent, AggregationEvent, QuantityEvent, TransactionEvent의 다섯 가지 이벤트가 존재한다. 마스터 데이터는 이벤트 데이터의 해석을 위한 데이터이다. 또한 EPCIS 명세서에서는 세가지 컴포넌트에 대해서 정의한다. EPCIS 캡처링 응용(Capturing Application)은 ALE 미들웨어로부터 받은 리포트를 기반으로 이벤트 데이터를 생성하고, EPCIS 저장소(Repository)는 분산 데이터베이스로서 EPCIS 캡처링 응용에서 생성된 이벤트 데이터를 저장하고 공유한다. EPCIS 액세스링 응용(Accessing Application)은 EPCIS 저장소에 접근하여 정보를 가져온다. EPC network와 마찬가지로 EPCIS 역시도 RFID만을 고려하기 때문에 주변 환경에 대한 정보를 줄 수 있는 센서 데이터의 저장 및 공유가 불가능하다.

본 논문에서 제안되는 Extended ALE 및 Extended EPCIS는 기존 EPC network의 약점을 극복하면서 RFID와 센서 네트워크의 데이터의 처리 및 정보 공유를 가능하게 한다. Extended ALE는 복합 이벤트 처리(Complex Event Processing, CEP)[5]를 이용한 이벤트 처리 컴포넌트로 EPC Network에서 센서 데이터의 처리에 사용된다. 또한 여러 사물의 상이한 능력들로 인해 야기되는 문제를 다루기 위한 방법 역시도 제안되었다. 거기에 더해 ALE 인터페이스를 확장하여 표현성을 늘렸다.

Extended EPCIS는 RFID 뿐만이 아니라 센서 데이터까지도 공유하고 사용자에게 제공할 수 있도록 하는 분산 데이터베이스 시스템이다. Extended EPCIS는 온도와 습도 같은 대부분의 종류의 센서를 표현할 수 있는 센서 표현 포맷을 가지고 있고 이를 이용하여 센서 데이터를 표현한다. 또한 EPCIS 캡처링 응용에서는 상위 레벨 규칙(high-level rule)을 이용하여 ALE 미들웨어로부터 전달받은 RFID 및 센서 데이터를 기반으로 이벤트를 생성한다. 본 논문에서는 식품 안전 시나리오를 프로토타이핑 함으로써 제안된 미들웨어의 디자인을 검증한다.

2. 관련 연구

EPCSN에서의 CEP 연구 [6]는 기존 EPC network의 ALE 인터페이스 외에 자체적인 미들웨어 구조를 디자인했고, 그것은 EPC network의 다른 컴포넌트와

호환이 되지 않는다. RFID의 CEP 연구 [7]는 오직 RFID 데이터만을 다루고 센서 데이터를 저장하거나 공유하는 매커니즘이 없다.

센서 데이터베이스에 관해서 진행된 몇 가지 연구가 있는데 Cougar 프로젝트[8]는 센서 네트워크를 일종의 데이터베이스적인 측면에서 접근한 초기의 연구들 중 하나이다. TinyDB[9]도 마찬가지로 센서 네트워크를 하나의 데이터베이스로 보고 센서 네트워크 위에서의 쿼리 처리에 대해서 연구하였다. 하지만 이러한 연구들은 센서 네트워크에 대해서만 진행이 되었고, RFID에 대해서는 고려되지 않았다.

EPCglobal Architecture Framework는 물류 관리를 위한 RFID 데이터 처리에 있어 사실상의 (de-facto) 표준이다. EPCglobal Architecture Framework에서는 각각의 사물들이 RFID tag를 가지고 있고, 그것은 전세계적으로 유일한 식별자인 EPC를 포함하고 있다. EPCglobal이라는 단체는 EPCglobal Architecture Framework에서 하드웨어 및 소프트웨어 역할과 그 역할간의 인터페이스를 정의하였다. 그림 1은 EPCglobal Network의 구조를 보여준다.

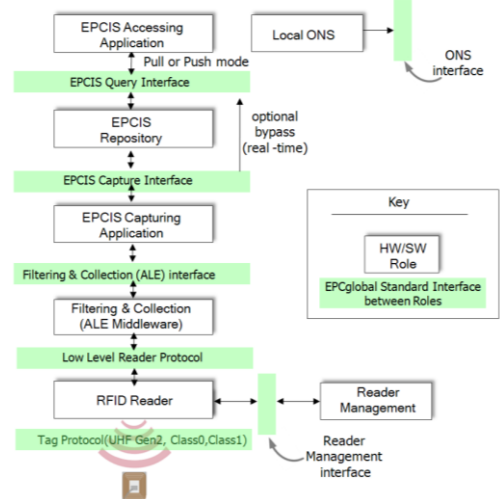


그림 1. EPCglobal Network

EPCglobal Network의 컴포넌트 중의 하나인 Low Level Reader Protocol (LLRP)는 air protocol 명령어 파라미터를 정의하고 수행할 operation을 정의한다. Application Level Events (ALE) 인터페이스는 필터링 정보와 그룹핑 정보를 데이터의 크기를 줄일 수 있도록 클라이언트에게 보낼 요청으로서 정의한다. 이렇게 처리된 데이터는 EPCIS의 Capturing Application에 전달되고 EPCIS Repository에 저장되어 EPCIS Accessing Application을 통해서 공유가 이루어지게 된다. Object Name Service (ONS)는 EPC를 받아 authorized EPCIS의 주소를 리턴 하는 검색 서비스를 제공한다.

3. Extended ALE

3.1 사물들마다 다른 특성

각 사물마다 다른 특성 문제를 해결하기 위해서 ALE 인터페이스에 있는 Tag memory specification (TMSpec) 라는 컴포넌트가 매우 중요한 역할을 수행한다. 이 컴포넌트는 원래 ALE 필드의 속성을 기술하기 위해 만들어졌는데, 그것은 Tag Memory에서의 Byte Array에 대한 정보로 나타내어진다. 그림 2는 TMSpec의 역할을 설명하고 있다.

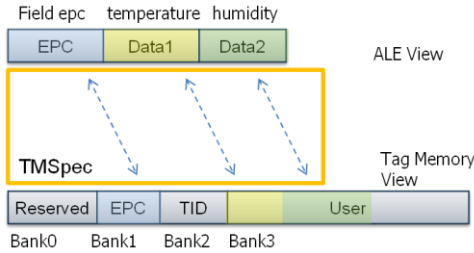


그림 2. TMSpec의 역할

이 확장에서는 TMSpec은 센서 노드의 메타데이터로서 기능하기 때문에 매우 중요하다. Extended ALE는 LLRP 메시지를 TMSpec에 따라 해석한다. 그리고 TMSpec은 Profile Service를 통해 검색될 수 있다. Profile Service는 여러 DNS서버들이 계층적 구조를 가지고 도메인 이름을 가지고 IP주소를 조회하는 것과 비슷하게 동작하는데, 여러 계층적인 Profile Server들이 EPC를 바탕으로 TMSpec을 조회한다. 이를 위해 각 TMSpec에 EPC패턴을 할당하는 것이 필요하다.

3.2. 복합 이벤트 처리

센서 데이터 처리 방법에 있어서, 우리는 복합 이벤트 처리(CEP) 개념을 이용하였다. CEP는 이벤트 구름 안에서 의미 있는 이벤트를 발견해 내는 일에 관련된 개념으로, Extended ALE 컴포넌트에 강력한 센서 데이터 처리 모델을 제공한다. 관심 있는 이벤트는 Extended ALE 컴포넌트에 등록될 수 있으며, 만약 관심 있는 이벤트에 명시된 조건이 만족되었을 경우 이벤트가 Subscriber에게 통보된다. 관심 있는 이벤트는 그림 3에서 볼 수 있듯이 여러 계층으로 나타낼 수 있다.

TMSpec은 여기서도 중요하다 왜냐하면 그것은 CEP에서 최하위 이벤트로서 기능하기 때문이다. 여러 계층의 이벤트를 처리하기 위해서는 최하위 이벤트가 필요하다. 최하위 이벤트는 RFID 태그 데이터 이거나 센서 데이터 이벤트이다. 센서 데이터의 다른 속성들은 TMSpec으로 다루어진다. 이러한 최하위 이벤트 위에 여러 조건을 더한 최하위 이벤트의 조합들은 복합 이벤트라고 불리고 그것들이 CEP에서 기술되고 처리된다.

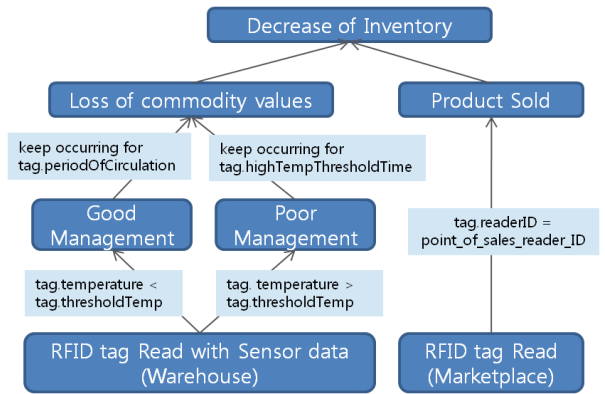


그림 3. 다중 레벨 복합 이벤트 처리의 예제

3.3. Expressible ALE interface

ALE 인터페이스는 여러 계층의 이벤트를 표현할 수 있도록 충분히 표현력이 좋아야 한다. 이를 위해 Event Cycle Specification (ECSpec)이라는 필터링 및 그룹핑에 관련된 ALE의 컴포넌트가 확장되었다. EPC network에서 F&C 미들웨어는 필터링과 그룹핑에 관련된 Event Cycle을 ECSpec에 따라서 정의한다. ECSpec 내부의 Boundary Condition이 만족되면 ALE는 리포트를 생성하고 Subscriber한테 통보한다.

ECSpec의 Boundary Condition으로부터 오는 타이밍 문제가 있다. CEP가 등록된 관심 있는 이벤트의 조건이 만족할 때마다 이벤트를 만들어 리포트를 생성하지만, 그 리포트는 ECSpec의 Boundary Condition이 만족할 때까지 통보되지 않는다. 기존의 Boundary Specification이 사용자 정의 트리거를 허용하고 있기는 하지만, 사용을 추천하지는 않는다. 왜냐하면 ALE 클라이언트가 Extended ALE 컴포넌트마다 트리거 정책을 알아야 하기 때문이다.

타이밍 문제에 대한 우리의 디자인 선택은 기존에 존재하는 Boundary Specification을 사용하지 않고, 새로운 타이밍 스펙인 InputViewSpec이라는 것을 정의한다. 타이밍 조건을 기술하는 방법은 거의 무한하나 4개의 기본적인 타이밍 조건이 기본으로 정의된다.

첫째, ECPeriodicInputViewSpec은 기존 Boundary Condition과 같다. 명시된 duration과 repeatPeriod에 따라 주기적으로 Event Cycle을 활성화시키는 것이다. 둘째, ECSlidingTimeInputViewSpec은 각각 이벤트가 일어나기 전의 duration을 명시한다. 예를 들면, high Temperature이벤트가 일어나기 전 2시간 동안 발생한 이벤트를 모을 수 있다는 것이다. 셋째, ECRBatchInputViewSpec은 이벤트가 발생한 횟수를 셀 수 있다. 그래서 재고관리 등의 응용에 유용하다. 넷째, ExternalSlidingTimeInputViewSpec은 ECSlidingTimeInputViewSpec과 duration이 이벤트 안에 있는 속성에 따라 다를 수 있다는 것을 제외하면 동일하다.

우리는 또한 CEP의 결과가 보고되는 양식과 CEP의

조건들을 명시하기 위해서 CEPOutput과 CEPFilter라는 새로운 속성을 정의했다.

3.1.4. Extended ALE middleware Design

우리는 또한 Extended ALE 미들웨어를 그림 4와 같이 디자인했다. Logical Reader Abstraction 컴포넌트는 몇 개의 Physical Reader를 하나의 Logical Reader로 취급할 수 있게 하고, 이 컴포넌트를 통해 Extended ALE는 LLRP Reader의 실제 배치에 대해서 일일이 신경 쓸 필요가 없다. TMParser는 LLRP 메시지를 TMSpec에 맞추어 해석한다. 만약 해석에 필요한 TMSpec이 정의되어 있지 않을 경우에는 TMSpec Manager는 Profile Service와 ONS를 통해서 해당 TMSpec을 찾는다. Event Condition Action Broker (ECA Broker)는 관심 있는 이벤트를 CEP 엔진에 등록한다. CEP엔진은 가장 중요한 컴포넌트로 관심 있는 이벤트를 찾아서 리포트를 만드는 역할을 한다. 만약에 Actuation 메시지가 이벤트에 정의되어 있다면 Action Manager는 actuation을 위한 LLRP메시지를 보낸다. Report Generator는 CEP엔진의 리포트를 ALE 인터페이스 포맷에 맞게 ECRreport로 바꾸는 역할을 한다. ECRreport는 XML 형식으로 SOAP/HTTP Binding을 통해 등록된 subscriber EPCIS에 전송되게 된다.

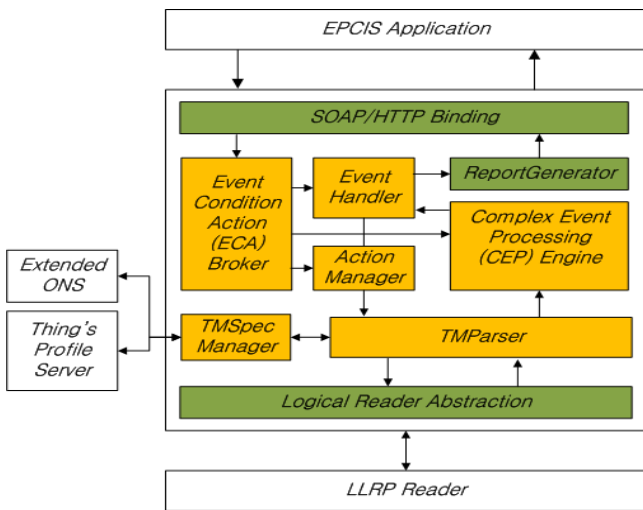


그림 4. Extended ALE 미들웨어 디자인

4. Extended EPCIS

Extended EPCIS는 Extended ALE로부터 전달되는 RFID 및 센서 데이터를 저장하고 이를 필요한 사용자 및 응용에서 표준 인터페이스를 통해서 공유할 수 있도록 해주는 분산 데이터베이스 시스템이다. 이 이후의 절에서 어떠한 식으로 센서 이벤트를 정의하고 어떻게 Extended EPCIS를 디자인하였는지에 대해서 표현할 것이다.

4.1. 센서 이벤트의 표현

Extended EPCIS에서 센서 데이터를 다루기 위해서는 우선적으로 센서 이벤트를 정의할 필요가 있다. EPCglobal Network의 EPCIS 명세서에는 이벤트 확장을 위한 두 가지 방법을 제시하고 있다. Extension points와 subclassing이 그 두 가지 방법이며, extension points의 경우 기존의 이벤트 타입에서 새로운 속성을 추가하여 확장하는 방법이다. 예를 들어서 ObjectEvent에 “Receiver”를 추가함으로써 해당 객체를 누가 받을 것인지에 대해서 표현할 수 있다. Subclassing의 경우 기존의 이벤트 타입을 기반으로 하여서 새로운 이벤트 타입을 정의하는 것이다.

본 논문에서는 subclassing방식을 이용하여서 SensingEvent 및 AbnormalConditionEvent라는 두 가지 이벤트 타입을 정의하였다. SensingEvent는 센서들이 온도나 습도와 같은 센서 데이터를 생성할 때 만들어지는 이벤트이고, AbnormalConditionEvent의 경우, 화재 혹은 과열과 같은 사용자가 예상치 못한 일이 발생한 경우 이벤트가 생성된다. 표 2와 3에서 SensingEvent와 AbnormalConditionEvent에 대한 자세한 설명이 되어있다.

표 2. SensingEvent의 속성

속성	타입	설명
eventTime	Time	이벤트가 벌어진 시간
recordTime	Time	EPCIS가 이벤트를 기록한 시간
eventTimeZoneOffset	String	이벤트가 발생한 장소의 표준 시간대
Epc	EPC	센서 값을 전달한 센서의 EPC
sensorType	SensorTypeID	온도와 습도 같은 센서의 타입
sensingValue	Int	센서 데이터의 값
sensingValueUnit	UnitID	화씨와 섭씨 같은 센서 값의 단위
location	LocationID	이 이벤트가 발생한 장소

표3. AbnormalConditionEvent의 속성

속성	타입	설명
eventTime	Time	이벤트가 벌어진 시간
recordTime	Time	EPCIS가 이벤트를 기록한 시간
eventTime ZoneOffset	String	이벤트가 발생한 장소의 표준 시간대
epc	EPC	센서 값을 전달한 센서의 EPC

state	String	과열 같은 현재 비정상적인 컨디션에 대한 표현
location	LocationID	이 이벤트가 발생한 장소

표2의 SensorTypeID는 센서 종류의 식별자를 나타낸다. 이를 단순한 문자열로 나타내는 경우, 한 가지의 센서를 나타낸다고 하더라도 다양한 표현이 존재할 수 있고 이로 인한 혼란을 초래할 수 있으므로 하나의 ID 체계를 갖추어 표현하도록 한다. 위와 비슷한 이유로 UnitID 및 LocationID를 정의하고 사용하였다. 하지만 AbnormalConditionEvent의 경우, 각 응용에 따라 다양한 상태가 있을 수 있으므로 식별자 체계를 만들기 보다는 문자열로 표현해 좀 더 다양한 상태를 가질 수 있도록 하였다.

4.2 Extended EPCIS의 구조

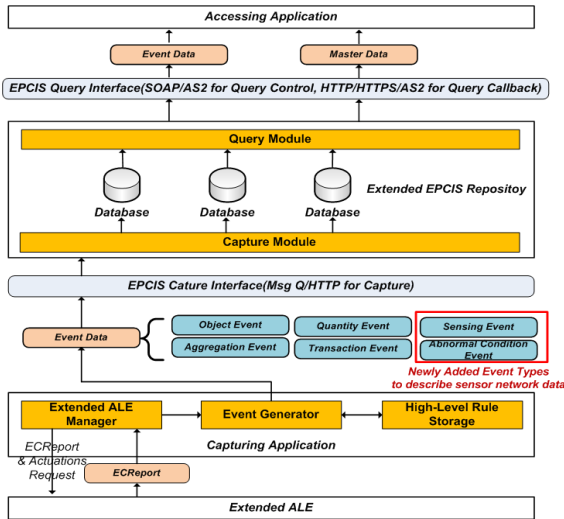


그림 5. Extended EPCIS의 구조

위에서 새로이 정의된 센서 이벤트 타입들을 이용하여 본 논문에서는 Extended EPCIS의 디자인을 제안한다. 그림 5에서 디자인된 Extended EPCIS의 구조를 보여준다.

캡처링 응용은 세가지 컴포넌트로 구성되어있다. Extended ALE Manager는 ECSpec과 따른 행동 규칙을 Extended ALE로 등록한다. 그리고 Extended ALE로부터의 ECReports를 Event Generator로 전달한다. Event Generator는 받은 ECReports와 상위레벨 규칙을 이용하여 이벤트를 생성하게 된다. 일련의 과정을 거치는 동안 상위 레벨 규칙에 따라서 필요하다면 새로운 ECSpec이나 행동 규칙을 Extended ALE에 등록하도록 하는 경우도 있다. 그리고 생성된 이벤트들은 Extended EPCIS Repository에 EPCIS Capture Interface를 통해서 저장된다.

Extended EPCIS Repository는 분산 데이터베이스로서 RFID와 센서 이벤트들을 저장하고 필요한 사용자 및 응용들 사이에서 공유한다. Capture

Module은 캡처링 응용으로부터 이벤트를 전달받고 이를 데이터베이스에 저장한다. 그리고 사용자 혹은 응용이 특정 이벤트들에 대한 데이터를 요청한다면 Query Module은 그에 해당하는 이벤트들을 데이터베이스로부터 가져와 사용자 혹은 응용에게 전달한다. 요청에 따라서는 주기적으로 이벤트 데이터를 전달하는 경우도 있다.

액세싱 응용은 Extended EPCIS의 클라이언트로 EPCIS Query Interface를 통해서 이벤트 데이터 및 마스터 데이터에 대한 요청을 하고 이에 대한 데이터를 받게 된다. 또한 요청 시에 한번만 받을 것인지 주기적으로 받을 것인지에 대해서도 정할 수 있다.

5. 구현

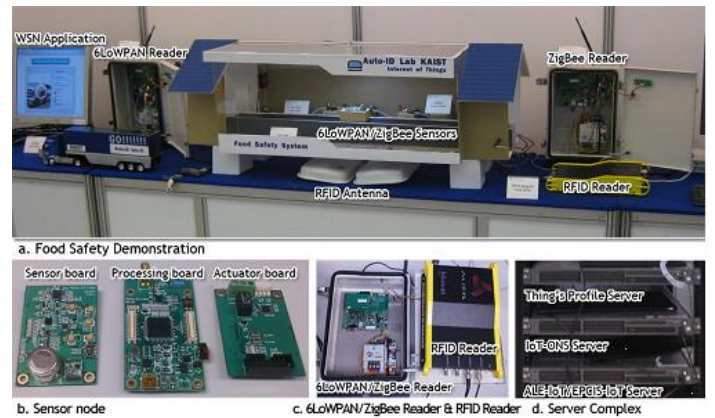


그림 6. EPC Sensor Network의 프로토타입

EPCSN의 구현은 음식 안전 보관 시스템을 프로토타이핑 하였다. 온도, 습도, 조도, 자이로 센서가 붙은 노드가 각각의 음식 아이টে 붙는다. 고온, 다습 등의 아이টে 좋지 않는 환경일 경우에는 선풍기를 작동하는 등의 적절한 actuation이 가해진다. 이 프로토타입에서는 IEEE 802.15.4 PHY/MAC위에서 동작하는 Zigbee와 6LoWPAN이 센서 노드 통신 프로토콜로 사용되었다. Zigbee Gateway와 6LoWPAN Gateway가 있어서 센서 데이터를 확장된 LLRP message로 바꾸어 Extended ALE에 전달하게 된다.

Extended ALE 미들웨어는 EPCglobal Network의 오픈소스 프로젝트인 Fosstrak [10]을 바탕으로 구현되었다. ALE 미들웨어 오픈소스 0.4.0은 ALE 1.1을 완벽하게 지원을 하지 않고 있기 때문에 우리는 ALE 1.1 인터페이스에 관련된 TMSpec Manager 등의 컴포넌트를 구현해야 했다. 그 결과 디자인 다이어그램에서의 노란 박스로 표시된 컴포넌트를 추가했다. 녹색 박스는 Fosstrak에 있는 기존 legacy이다.

CEP 엔진은 Java 기반의 오픈소스 CEP엔진인 Esper [11]를 사용해 구현하였다. Esper는 query 스타일의

그들의 자체 EPL을 가지고 있다. ECSpec에 기술된 이벤트는 Esper 형식으로 변환되어 CEP엔진에 등록된다.

Extended ALE 미들웨어와 profile service의 integration은 이 버전에서는 완성되지 않았다. 대신 TMSpec은 Extended ALE의 설정이 미리 등록되어 있어야 한다.

Extended EPCIS 역시 Fosstrak을 통해 구현되었고, MySQL이 데이터베이스로 사용되었다. 그리고 액세싱 응용을 우리의 프로토 타입에 맞게 수정하였다. 캡처링 응용은 새롭게 구현되었다. 캡처링 응용에서의 상위 레벨 룰의 적용은 완벽하지 않으나 프로토 타입을 검증하기 위해 시나리오에 관련된 몇 개의 룰을 hard-coding 하는 방식으로 구현하였다.

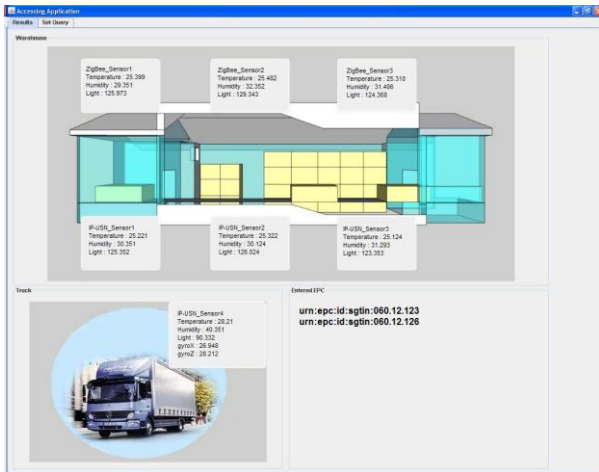


그림 7. 액세싱 응용의 GUI화면

그림 7은 액세싱 응용의 GUI를 보여준다. 이 프로토 타입에서는 첫 사용자가 주기적으로 결과를 요청하기 위해 subscription query를 설정하여야 한다. 그 후 액세싱 응용은 결과를 받아서 그림 7에서 보여준다. Abnormal Condition Events가 감지되면 액세싱 응용은 상태와 location 정보와 함께 경고 메시지를 보여준다

6. 결론

우리는 EPCSN의 컴포넌트로서의 Extended ALE 및 Extended EPCIS를 디자인하고 구현하였다. 이것은 EPCglobal Architecture Framework에서 기존의 약점을 극복하면서 RFID와 센서 데이터 처리를 모두 지원하도록 확장된 것이다. 우리의 디자인을 검증하기 위해 음식 안전 보관 시나리오를 바탕으로 EPCSN의 프로토타입을 구현하였다. 이 프로토타입은 RFID 리더뿐만 아니라 ZigBee나 6LoWPAN등의 상이한 센서 네트워크 또한 지원한다.

몇 가지 향후 연구 과제가 있다. 첫째, 우리가 복합 이벤트의 새로운 속성을 정의했지만 그것은 CEP 결과를 명시하는 가장 좋은 방법이라고 말할 수는 없다.

이벤트를 정의하는 언어는 Event Processing Language (EPL)이라고 부르는데, 다양한 인프라에서 사용될 수 있을 정도로 표준화할 만 한 강력한 EPL을 연구하는 작업이 필요하다. 둘째, 성능과 확장성에 대한 연구도 필요하다. 셋째, Extended EPCIS의 캡처링 응용에서 상위 레벨 규칙을 기술하는 언어에 대한 연구 또한 필요하다.

참고 문헌

- [1] Jongwoo Sung, Tom´as S´anchez L´opez, Daeyoung Kim, The EPC Sensor Network for RFID and USN Integration Infrastructure, to appear in theWiP session of Percom 2007, New York, March 19–21, 2007
- [2] EPCglobal, “EPCglobal Architecture Framework version 1.3”, http://www.epcglobalinc.org/standards/architecture/architecture_1_3-framework-20090319.pdf
- [3] EPCglobal, “The Application Level Events (ALE) Specification Version 1.1.1”, http://www.epcglobalinc.org/standards/ale/ale_1_1_1-standard-core-20090313.pdf
- [4] EPCglobal, “EPC Information Services (EPCIS) Version 1.0.1 Specification”, http://www.epcglobalinc.org/standards/epcis/epcis_1_0_1-standard-20070921.pdf
- [5] Luckham, David C. and Frasca, Brian. “Complex Event Processing in Distributed Systems”, Stanford University Technical Report CSL-TR-98-754, 1998
- [6] Weixin Wang, Jongwoo Sung, Daeyoung Kim, "Complex Event Processing in EPC Sensor Network Middleware for Both RFID and WSN", 11th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2008), May 5–7, Orlando, Florida, USA
- [7] Wang, F., Liu, S., Liu, P., Bai, Y.: Bridging Physical and Virtual World: Complex Event Processing for RFID data Streams. In: Proc. of the 10th International Conference on Extending Database Technology, Munich Germany (2006)
- [8] Philippe Bonnet, Johannes Gehrke, Praveen Seshadri, “Towards sensor database systems”, Mobile Data Management, pp. 3–14, 2001
- [9] Samuel R. Madden , Michael J. Franklin , Joseph M. Hellerstein , Wei Hong, “TinyDB: an acquisitional query processing system for sensor networks”, ACM Transaction on Database Systems(TODS), vol. 30, no. 1, pp. 122–173, 2005
- [10] EPCglobal Network Open Source Project, “Fosstrak”, <http://www.fosstrak.org>
- [11] EsperTech, “Esper – Event Stream and Complex Event Processing for Java”, <http://esper.codehaus.org/esper-2.3.0/doc/reference/en/html/index.html>