

---

# ALE 기반 RFID 미들웨어 설계 및 구현

홍연미\* · 변영철\*\*

A Design and Implementation of ALE-compliant RFID Middleware System

Yeon-mi Hong\* · Yung-cheol Byun\*\*

---

본 연구는 산업자원부와 한국기술재단의 지역혁신인력양성사업으로 수행된 연구결과임

---

## 요 약

유비쿼터스 컴퓨팅을 실현하기 위하여 자동식별, 센서 네트워크, 홈 네트워크, 텔레매틱스 등 다양한 분야에서 연구가 진행되고 있다. 특히 RFID 기술을 이용한 래거시 시스템과의 통합 및 데이터 수집, 장치 제어, 관리 등을 위하여 RFID 미들웨어의 중요성에 대한 인식이 확산되고 있다. 이에 발맞추어 EPCglobal에서는 기존의 Savant를 대체하는 개념으로서 RFID 하드웨어와 응용 사이에 존재하는 ALE(Application Level Events) 표준 스펙을 정의하였다. 이는 레이어의 내부 구조 및 구현 기술 등은 명시하지 않고 오직 외부 인터페이스만을 정의함으로써 향후 상호 운용성 및 표준 적합성 검증을 용이하게 한다. 본 논문에서는 이러한 ALE 스펙에 기반하여 RFID 태그 데이터를 효율적으로 처리할 수 있는 RFID 미들웨어 시스템을 설계하고 구현하였다.

## ABSTRACT

Nowadays, to realize ubiquitous computing environment, many research activities have been going on within various kinds of research domains including automatic identification, sensor network, home network, telematics and so on. Especially, RFID middleware that supports the aggregation of RFID tag data, control and management, and the integration with legacy systems has recently gained a lot of attention. Meanwhile, EPCglobal defined an ALE(Application Level Events) standard specification, which exists between RFID readers and applications, and substitutes the previous systems called Savant. In the specification internal structures and implementation technologies of ALE are not mentioned and only external interfaces are defined. This approach eases the verification of standard compliance and inter-operability of the layer. In this paper, we present the design of ALE-compliant RFID middleware systems that process RFID tag data efficiently.

## 키워드

Middleware, RFID, Ubiquitous, ALE, RFID 미들웨어

---

\* 제주대학교 대학원 컴퓨터공학과 석사과정  
\*\* 제주대학교 통신컴퓨터공학부 조교수 (교신저자)

접수일자 : 2007. 3. 14

## I. 서 론

유비쿼터스 컴퓨팅을 실현하기 위하여 자동식별, 홈네트워크, 센서 네트워크, 텔레메티스 등 다양한 분야에서 연구가 활발히 이뤄지고 있다. 특히 자동식별 분야에 있어서 사람 및 사물의 신원 정보를 제공할 수 있는 RFID(Radio Frequency Identification) 기술이 주목받고 있다.

최근에는 RFID 기술을 이용하여 물류, 소매업, 의료, 공장·가정·사무실 자동화, 보안, 재난 방지, 재산 관리 등 다양한 서비스를 제공하는 응용들이 개발되고 있다 [1, 2, 3]. 응용 서비스 개발자가 RFID 기술을 이용한 다양한 유비쿼터스 응용 서비스를 쉽게 구축할 수 있으려면 RFID 태그가 부착된 객체와 응용 서비스 사이에서 교량 역할을 하는 미들웨어가 필요하다[4].

RFID 미들웨어는 여러 가지 센서를 관리하며 센서로부터 다양한 프로토콜을 이용하여 데이터를 수집하고, 또한 수집되어 가공되지 않은 데이터로부터 의미 있는 정보, 혹은 응용이 사용하기 쉬운 형태의 정보를 추출하여 응용 서비스에 전달하는 기능을 기본적으로 수행하여야 한다.

미들웨어 시스템 제품 또는 솔루션으로는 오라클의 Sensor Edge Server, CapTech의 TagsWare, SUN의 Java System RFID 소프트웨어 등이 있다. 하지만, 복잡한 비즈니스 로직의 처리나 다양한 형식을 가지는 센서 데이터의 처리에 대한 고려가 미진한 상태이다[5]. 그리고 응용 서비스와 미들웨어 간 독립성을 제공하기 위하여 제안된 사실상의 표준에 기반하여 개발하지 않음으로 다양한 외부 비즈니스 로직과의 상호 운영성에 대한 서로 다른 제약으로 인한 문제점이 발생하여 비용 및 유통성 면에서 이점을 극대화하지 못한다.

최근 EPCglobal에서 진행되고 있는 EPC 네트워크 구성 요소 중 RFID 리더 장치와 응용 사이에 존재하는 ALE(Application Level Events)가 기존의 Savant 개념을 대체하였는데, ALE 스펙에서는 미들웨어 내부에 대한 상세한 구현에 대해서는 다루지 않고 오직 응용과의 인터페이스에 대해서만 기술함으로써 향후 미들웨어와 응용 간 상호운용성 및 표준 적합성 검증을 용이하게 한다.

본 논문에서는 원시 EPC 데이터를 획득하는 하부 구조 모듈과 데이터를 필터링하고 카운팅하는 구조적 모듈, 그리고 데이터를 사용하는 클라이언트 응용 간의 독립성을 제공하는 ALE 표준 스펙에 기반하여 RFID 데이터를 효율적으로 처리하고 이를 응용에게 제공하는

RFID 미들웨어를 설계하고 구현한다.

본 논문의 구성은 다음과 같다. II장에서는 본 논문과 관련된 연구 및 기술에 대하여 살펴보고, III장에서는 본 논문에서 제안하는 ALE 표준 스펙 기반 RFID 미들웨어 시스템을 설계한다. IV장에서는 설계에 따른 구현과 실행을 통하여 미들웨어가 제대로 동작하는지 실험한다. 마지막으로 V장에서 본 연구에 대한 결론을 맺는다.

## II. 관련연구

### 2.1. RFID 기반 미들웨어 제품 및 솔루션

Oracle의 Sensor Edge Server는 데이터 수집, 이벤트 처리, 데이터 분배(dispatching) 등의 기능을 지원하는 센서 기반 서비스 통합 플랫폼으로 미리 정의된 필터들을 이용하여 데이터를 추출하여, 보다 복잡한 데이터를 처리하고자 할 경우 직접 필터를 작성할 수 있다[6].

CapTech 사의 TagsWare는 RFID 태그 데이터를 응용에 전달하기 위한 링크, RFID 장비로의 표준 인터페이스를 지원하는 드라이버, 그리고 응용에서 링크와 드라이버의 사용을 지원하는 응용 기반 요소들로 구성된다. 리더 장치로부터 수집된 원시 데이터에서 태그 데이터를 추출하여 평활화(smoothing)한 후 응용에 전달하기 위한 링크 컴포넌트가 체인으로 연결된 구조를 제공한다[7].

SUN 사의 Java System RFID 소프트웨어는 다양한 센서로부터 오는 센서 데이터 스트림을 이벤트 관리기에서 처리하고, 리더 어댑터, 필터, 로거(logger), 엔터프라이즈 게이트웨이 등으로 구성된다. 델타(delta)와 평활화 질의를 할 수 있고, 필터들을 서로 연결하여 특정 마스크(mask) 조건을 만족하는 EPC 데이터 값을 얻을 수 있을 뿐만 아니라 사용자 정의 필터를 개발할 수 있는 도구를 지원한다[8].

ETRI의 UbiCore 시스템은 XML 기반 미들웨어 시스템으로 다양한 센서를 관리하며, XQueryStream이라는 XQuery에 기반한 연속 질의 언어를 제공한다. 필터링과 중간 결과 재사용을 통하여 스트림 데이터에 대한 질의 처리 속도를 개선하였고, 연속적으로 생성되어 들어오는 실시간 데이터뿐만 아니라 저장된 이력 데이터에 대한 질의를 지원하며 컨텍스트와 서비스의 연계 정보를 표현하기 위한 마크업 언어인 CSML(Context-driven Service Markup Language)를 제공한다.

이러한 RFID 미들웨어들은 다양한 기능 및 데이터 처리 방법을 제공함으로써 데이터 처리의 효율성 면에서 뛰어나다. 하지만 응용 서비스와 미들웨어 간 독립성을 제공하기 위한 사실상의 표준을 따르지 않음으로써 향후 다양한 외부 비즈니스 로직과의 상호 운영이 필요할 경우 서로 다른 제약으로 인한 문제점 발생으로 인하여 비용 및 융통성 면에서 개선의 여지가 있다.

## 2.2. EPCglobal과 RFID 미들웨어 표준

EPCglobal은 2003년 10월에 EAN과 UCC가 Auto-ID 센터를 흡수하여 설립한 비영리 기구로서 Auto-ID 센터에서 개발한 EPC 네트워크에 대한 기술의 표준화와 상용화, 그리고 EPC 코드 보급과 관리 등을 목적으로 활동하고 있다[9]. EPCglobal은 RFID 미들웨어 표준화를 주도하고 있으며, 소프트웨어 관련 산업체에서는 이를 참조하여 솔루션을 개발하고 있다.

EPCglobal은 RFID 태그로부터 외부 응용에 이르기까지의 구성 요소를 계층 구조로 표현하고 각 계층 간의 인터페이스를 표준화 대상으로 삼고 있다. EPCglobal은 RFID 미들웨어와 관련하여 2002년에 구현 스펙 중심의 Savant 버전 0.1과 2003년에 Savant 버전 1.0을 제안하였고, 2005년 11월에는 다양한 센서로부터 EPC 코드 데이터를 입력받아 필터링하고 통합하여 클라이언트 응용에게 제공하는 미들웨어 인터페이스에 관한 ALE 스펙 버전 1.0을 발표하였다[10].

ALE 계층은 데이터를 수집하고 필터링하여 응용으로 제공하는 의미 있는 이벤트를 생성하는데 관심이 있으면 반면 비즈니스 계층은 비즈니스 처리 및 이벤트 저장에 관심이 있다. ALE 계층은 데이터를 사용하는 응용에게 표준 인터페이스를 제공하여 독립성을 보장함으로써 미들웨어 기술 제공자와 이벤트 데이터를 사용하는 사용자에게 이점을 제공한다[11].

## 2.3. ALE 스펙 주요 요소

EPCglobal ALE 표준 스펙에는 Main API인 ALE 인터페이스 메소드, 메소드 처리에 필요한 예외와 ALE에서 사용하는 자료 형, 다양한 센서에서 들어오는 EPC 데이터들을 응용이 필요로 하는 데이터로 가공 처리하는 필터링과 그룹핑 등 다양한 처리를 위한 연산자가 기술되어 있다.

응용이 요구하는 데이터를 전송하려면 데이터를 모

으는 과정이 필요하며 이를 이벤트 사이클(event cycle)이라고 한다. 이벤트 사이클은 그림 1과 같이 1개 혹은 여러 개의 리더에서 수행되는 1번 이상의 읽기 사이클(read cycle)로 구성되는데, 이러한 이벤트 사이클은 응용의 관점에서 볼 때 단일 동작으로 간주된다. 참고로 클라이언트 응용은 이벤트 사이클 정보를 스펙에 기술하여 ALE 미들웨어로 전송한다[11].

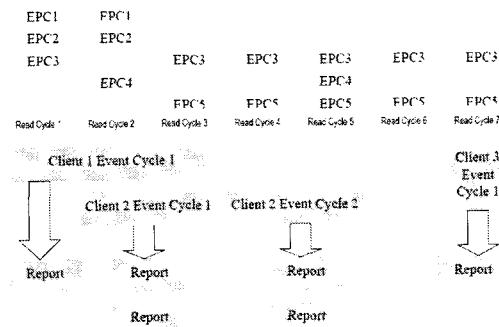


그림 1. 이벤트 사이클  
Fig. 1. Event cycles of an ALE layer

어떻게 이벤트 사이클을 수행할지, 이벤트 사이클 동안 어떤 데이터를 수집할지, 그리고 수집한 데이터를 어떻게 가공하여 클라이언트 응용으로 전송할지에 대한 정보를 작성해야 한다. 이를 위하여 그림 2와 같은 ECSSpec이라는 클래스가 있다. 여기에는 어느 리더 장치로부터 EPC 데이터를 읽을 것인지에 대한 정보가 들어 있다. 즉, 이벤트 사이클의 범위 정보(boundaries)와 이벤트 사이클에 포함될 읽기 사이클들을 수행하는 리더 장치 리스트 정보가 기술된다(readers). 또한 EPC 데이터에 대해 수행할 연산에 대한 정보가 기술된다.

```

readers : List // List of logical reader names
boundaries : EBoundarySpec
reportSpecs : List // List of one or more ECReportSpec
includeSpecInReports : boolean
<>extension point<>
---
```

그림 2. ECSSpec 자료 형  
Fig. 2. ECSSpec data type

스펙을 이용하여 EPC 데이터를 정제한 후 클라이언트 응용으로 전송하여야 한다. 이를 위한 자료 형으로서 ECReports 클래스가 있다.

ECReports 객체는 ECSpec에 기술되어 있는 한 번의 이벤트 사이클을 수행하여 생성된다. 한 개의 리포트에는 여러 개의 ECReport 객체가 들어있으며 각각의 ECReport 객체는 ECSpec의 ECReportsSpec 객체를 참고하여 생성된다. ECReports 자료 형은 그림 3과 같다.

```
specName : string
date : dateTime
ALOID : string
totalMilliseconds : long
terminationCondition : ECTerminationCondition
spec : ECSpec
reports : List // List of ECReport
<extension point>
---
```

그림 3. ECReports 자료 형  
Fig. 3. ECReports data type

#### 2.4. 인터페이스의 호출 시 처리 과정

하나 혹은 그 이상의 클라이언트들이 미들웨어의 ALE 인터페이스에 있는 메소드를 호출하면 미들웨어는 내부적으로 ECSpec에 기술되어 있는 내용을 분석하여 그에 따른 적절한 동작을 수행한 후 결과를 만들어 응용으로 전달한다.

응용이 미들웨어에게 데이터를 요청하려면 스펙을 작성하여 전송해야 하며, 혹은 그 이전에 미들웨어에 등록한 스펙을 이용할 수 있다. 그림 4는 스펙의 일부분으로서 ECSpec 자료 형에 정의되어 있는 필드들을 XML로 표현한 것이다. logicalReader 요소는 어디에서 EPC 데이터를 읽을지를, boundarySpec 요소는 언제, 어떻게 데이터를 읽는지에 대한 정보를 표현한다. 필터링 요소는 어떤 데이터를 선택할지를 기술하며, 그룹 요소는 어떻게 데이터를 가공할지를 기술한다.

```
<logicalReaders>
  <logicalReader><dock_1a></logicalReader>
  <logicalReader><dock_1b></logicalReader>
</logicalReaders>
<boundarySpec>
  <startTrigger>http://sample.com/trigger1</startTrigger>
  <repeatPeriod unit="MS">3000</repeatPeriod>
  <stopTrigger>http://sample.com/trigger2</stopTrigger>
  <duration unit="MS">3000</duration>
</boundarySpec>
<reportSpecs>
  <reportSpec reportName="report1">
    <reporter set="CURRENT"/>
    <output includesTag="true"/>
  </reportSpec>
</reportSpecs>
```

그림 4. ECSpec의 XML 인스턴스 일부분  
Fig. 4. An example of ECSpec XML instance

스펙은 RFID 미들웨어의 동작에 따라 초기 상태인 Unrequest, 그리고 Request와 Active 상태로 존재할 수 있

다. 스펙을 작성한 후 미들웨어에 등록하려면 인터페이스의 define 메소드를 호출한다. 이 경우 스펙은 Unrequest 상태가 되고, 현재 미들웨어가 스펙을 사용하는 단계가 아니기 때문에 사용자가 원할 경우 스펙을 삭제하거나 수정할 수 있다.

subscribe 메소드를 호출하면 등록된 스펙을 이용하여 이벤트 사이클을 시작할 수 있다. 스펙에 있는 StartCondition 조건에 따라 스펙 상태가 변경될 수 있는데, 조건이 Trigger일 경우 미들웨어는 조건을 만족하는 EPC 데이터가 입력되면 스펙 상태는 Request에서 Active 상태로 전이되어 이벤트 사이클이 시작된다.

subscribe 메소드로 시작한 이벤트 사이클을 종료하려면 unsubscribe 메소드를 호출하면 된다. unsubscribe를 호출하면 스펙의 상태는 다시 Unrequest 상태로 전이되고, 미들웨어는 그때까지 받은 데이터만을 이용하여 응용에게 전송할 리포트를 생성한다. 이 경우 미들웨어는 리포트 생성 시 이벤트 사이클의 종료 조건을 기술하는 부분에 Unrequest를 기술한다.

### III. ALE 스펙 기반 RFID 미들웨어 설계

#### 3.1. 제안하는 시스템 개요 및 주요 구성요소

본 연구에서 제안하는 미들웨어는 다양한 센서 혹은 리더 장치로부터 들어오는 방대한 양의 EPC 데이터를 응용이 사용하기 편리한 이벤트로 변환하여 제공한다. 이러한 미들웨어는 응용과 물리 계층을 분리함으로써 기술 제공자와 사용자에게 비용 및 유통성 면에서 이점을 제공한다. 그림 5는 제안하는 전체 시스템 구조도이다.

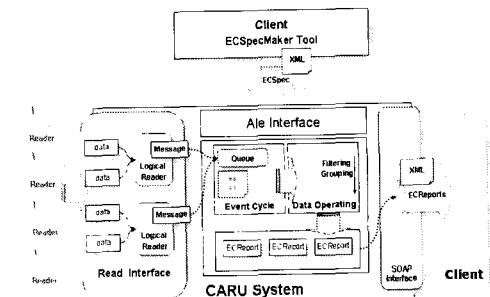


그림 5. 시스템 전체 구조도  
Fig. 5. An overall structure of the system

제안하는 시스템은 다양한 센서 연동 및 관리 기능, ALE 인터페이스 제공 기능, 그리고 EPC 데이터 처리 기능을 갖도록 설계하였다. 다양한 센서 연동 및 관리 기능을 위한 장치 인터페이스 모듈을 통하여 여러 유형의 EPC 호환 RFID 리더와 연동할 수 있으며, 다양한 리더로부터 입력된 데이터를 입력받기 위하여 장치 관리와 연결 관리를 수행한다. 또한 물리적으로 다른 여러 리더 장치로부터 입력받은 데이터를 하나의 장치에서 입력받은 것으로 인식하기 위하여 하나 혹은 여러 물리 리더 장치를 하나의 논리(logical) 리더로 정의하여 관리하는 일을 수행한다.

ALE 표준 인터페이스는 응용 개발자로 하여금 원하는 EPC 데이터 유형을 정의하고 데이터 전송을 요청하는 등의 기능을 제공한다. 기존의 Savant에서는 응용이 미들웨어 인터페이스에게 데이터를 요청하기 보다는 미들웨어 관리자를 통하여 간접적으로 받고자 하는 데이터를 정의하였다. 따라서 ALE 표준 인터페이스는 응용 개발자에게 큰 유통성을 제공한다.

EPC 데이터 처리 모듈은 스펙을 분석하여 하나의 이벤트 사이클 동안에 데이터를 얻은 후 필터링과 그룹핑 등을 수행한다. 그런 다음 응용 개발자가 원하는 다양한 형태로 데이터 포맷을 변환한 후 리포트로 만들어 응용으로 전송한다.

### 3.2. EPC 데이터 처리 과정

클라이언트 응용이 스펙을 작성한 후 define 인터페이스 메소드의 파라미터로 미들웨어 시스템에 등록한다. 이때 스펙은 스펙을 관리하는 스펙 리스트 자료구조에 등록된 후 예상치 못한 시스템의 다운 및 이상이 발생할 경우에 대비하기 위하여 일정한 시간 간격으로 백업 모듈에 의하여 데이터베이스에 저장된다.

응용은 subscribe와 poll 메소드를 이용하여 시스템에 등록되어 있는 스펙을 지정함으로써 원하는 데이터를 요청할 수 있다. subscribe 메소드를 호출하여 데이터를 요청하면 해당 스펙의 상태가 SpecStatesList라는 자료구조에 등록된다. 그런 다음 논리리더로부터 들어오는 데이터를 저장하기 위한 이벤트 큐를 생성한다.

큐를 생성한 후 이벤트 사이클의 시작 조건과 종료 조건에 의해 얻어진 중복 제거된 EPC 데이터를 수집하여 필터링, 그룹핑, 데이터 셋, 전송할 데이터 유무 체크 등의 처리 과정을 수행한 후 지정된 데이터 포맷으로 변환하여 최종 결과물인 리포트 객체를 만들어 클라이언트 응용으로 전송한다.

### 3.3. 데이터 처리를 위한 주요 모듈 설계

EPC 데이터를 처리하기 위한 주요 패키지에는 EventCycleProcess, DataOperator, Reporting 패키지 등이 있다. EventCycleProcess 패키지는 스펙에 기술되어 있는 Boundary 정보를 바탕으로 이벤트 사이클을 수행하여 EPC 데이터를 얻는 클래스들로 구성된다. 그림 6은 이 패키지를 구성하는 클래스들의 다이어그램이다.

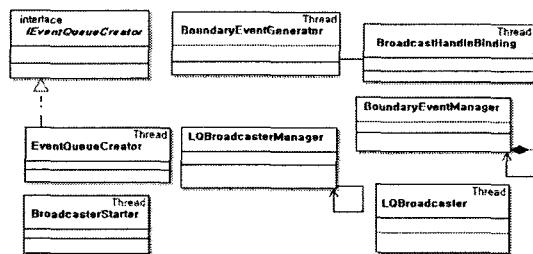


그림 6. Event Cycle Process 패키지

Fig. 6. A class diagram of EventCycleProcess package

DataOperator 패키지에는 그림 7과 같이 응용에서 필요한 데이터를 생성하기 위하여 스펙에 기술되어 있는 필터링, 그룹핑 등의 연산 처리를 담당하는 클래스 및 관련 패키지가 들어있다.

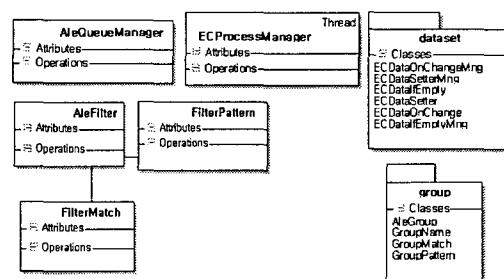


그림 7. Data Operator 패키지

Fig. 7. A class diagram of Data Operator package

ECProcessManager 모듈은 이벤트 사이클 수행으로부터 얻은 데이터를 받아 AleFilter 모듈을 이용하여 필터링하고, ECDataIsEmptyMng 모듈을 이용하여 처리된 결과 데이터가 존재하는지 여부를 검사한다. 또한 ECDataSetterMng 클래스 모듈과 ECDataOnChangeMng 모듈, AleGroup 모듈 등을 이용하여 클라이언트가 지정한 연산을 수행한다.

Reporting 패키지는 일련의 정제 과정을 통해 얻은 테

이터를 ECReports 형태로 가공하여 응용으로 전송하는 역할을 담당한다.

EPC 데이터 처리를 위하여 필요한 자료구조들이 있는데, 그림 8은 스펙을 관리하기 위한 SpecPool 테이블로서 define 메소드를 호출할 경우 스펙 관련 정보를 저장하기 위한 SpecPool이 있다. 이 테이블은 스펙 이름에 따라 스펙이 유일하게 존재하도록 하기 위하여 해쉬 맵 자료구조를 이용하여 설계하였다.

method	specName	spec	Dirty bit
define	mySpec <String>	ECSpec <Doc>	1

그림 8. 스펙 관리 테이블

Fig. 8. The table for spec management

*define, undefine, subscribe* 등의 메소드를 호출함에 따라 스펙의 상태가 달라진다. 스펙 이름을 키로 하여 스펙 객체가 저장되며, 메소드 이름 필드에는 스펙의 상태를 변화시킨 메소드 이름을 저장한다. 스펙 상태 정보가 변경되면 백업 비트가 1이 되며, 이에 따라 백업 모듈이 스펙 상태 정보를 데이터베이스에 백업한다.

## IV. 구현 및 실험

### 4.1. 구현 환경

본 연구에서는 제안하는 시스템을 구현하기 위하여 이클립스 환경에서 자바 언어를 이용하였다. 데이터베이스로는 MySQL 4.0을, 웹 관리 도구를 실행하기 위한 웹 서버로는 Tomcat 5.0을 이용하였다. RFID 리더로는 A사의 900Mhz 2가지 종류, I사의 모바일 리더 장치, 그리고 애뮬레이터를 이용하였다.

### 4.2. 주요 클래스 구현

제안하는 시스템의 메인 클래스로서 CaruApp라는 클래스를 구현하였으며, AleCore라는 EPC 데이터 처리를 위한 최상위 클래스를 구현하였다. AleCore 클래스는 리더 장치로부터 EPC 데이터를 입력받기 위한 ListenerStarter 클래스, 큐를 생성하는 QueueCreator 클래스, 리더 장치 모듈을 관리하는 ReaderModuleManager 클래스, 리더 그룹을 관리하는 ReaderGroupManager 클래스, LogicalReaderManager 클래스 등을 멤버로 갖는다. 그림 9는 이를 기술한다.

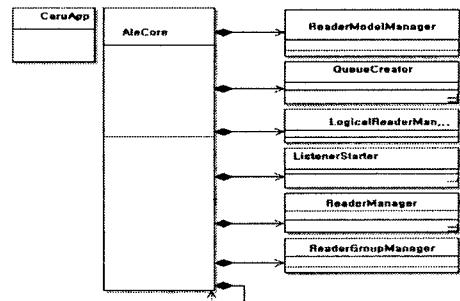


그림 9. 제안 시스템의 클래스 다이어그램

Fig. 9. Class diagram of the system

### 4.3. 실험 내용 및 환경

본 실험에서는 미들웨어 시스템을 구동하고, 물리 리더 설정 및 관리, 논리 리더 관리, 스펙 생성 도구를 이용한 스펙 작성, 미들웨어 내부에서의 데이터 처리 및 변환, 클라이언트 응용으로의 결과 전달 등을 위해 Alien 리더 1대, 리더 애뮬레이터 3대, 모바일 리더 1대 등 총 5 대를 이용하여 실험하였고, 논리 리더 설정은 물리 리더와 1대 1일 대응도 가능하나 동일한 포맷 데이터일 경우 하나의 논리 리더로 설정하였다.

### 4.4. 서비스 요청과 데이터 처리

클라이언트 응용이 *subscribe* 메소드로 서비스를 요청할 경우 미들웨어에 의해 수행되는 과정을 협력도로 표현하면 그림 10과 같다. 그림 11은 스펙을 작성하여 데이터를 요청할 경우 데이터베이스에 스펙 정보가 저장된 모습이다.

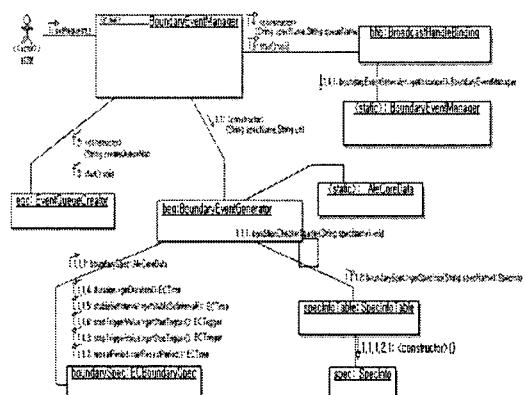


그림 10 subscribe 요청 시 처리과정

Fig. 10. Collaboration diagram for subscribe operation



제6호, 2006.

- [ 6 ] "Enterprise Information Architecture for RFID and Sensor-Based Services," Oracle White Paper, 2006.
- [ 7 ] "TagsWare: Agile RFID Solutions," <http://www.tagsware.com>.
- [ 8 ] Sun Microsystems, "The Sun EPC Network Architecture," Technical White Paper, 2004.
- [ 9 ] EPCglobal, <http://www.epcglobalinc.org>.
- [10] Auto-ID Lab., <http://www.autoidlabs.org>
- [11] EPCglobal, "The Application Level Events (ALE) Specification Version 1.0," 2005.

### 저자소개



홍 연 미(Yeon-Mi Hong)

2004년 제주대학교 통신.컴퓨터공학부 컴퓨터공학전공 학사  
2006년 3월~ 현 제주대학교 컴퓨터공학과 석사과정

※ 관심분야 : 미들웨어, 클러스터링, 유비쿼터스 컴퓨팅, 상황인지



변 영 철(Yung-Cheol Byun)

1993년 제주대학교 정보공학과 학사  
1995년 연세대학교 컴퓨터과학과 석사  
2001년 연세대학교 컴퓨터공학과 박사

2001년 한국전자통신연구원 선임연구원  
2002년~현 제주대학교 공과대학 컴퓨터공학전공 교수  
※ 관심분야 : 패턴인식, 시맨틱 웹, 상황인식, 지능형 컴퓨팅, 유비쿼터스 미들웨어