

---

# 대용량 데이터처리를 위한 XML기반의 RFID 미들웨어시스템

## RFID Middleware System based on XML for Processing Large-Scale Data

---

박병섭  
인하공업전문대학 컴퓨터시스템과

ByoungSeob Park(bspark@inhatc.ac.kr)

---

### 요약

본 논문에서는 대용량 데이터 처리를 위한 XML 기반 RFID 미들웨어 시스템을 구현하였다. 개발된 RFID 미들웨어 시스템은 태그 데이터를 수집하는 리더 인터페이스, 태그 데이터를 의미있는 데이터로 가공하여 응용인터페이스에게로 전송하는 이벤트 매니저, 그리고 기존 응용과의 인터페이스를 위해 다양한 응용접근프로토콜을 지원하는 응용 인터페이스를 설계 구현하였다. 리더 인터페이스는 고정형 장비뿐만 아니라 핸드헬드 유형의 PDA형 리더도 지원하도록 어댑터 형태로 구현되었다. 구현된 미들웨어 시스템은 빠른 필터링 처리기능을 가지며, 응용프로그램 접근 프로토콜인 HTTP, XML, JMS, SOAP을 지원한다. 성능은 CPU 이용률 측면에서 성능을 입증하였다.

■ **중심어** : | RFID미들웨어 | 리더인터페이스 | 이벤트매니저 | 응용인터페이스 |

### Abstract

We implement the RFID middleware system based on XML for large-scale data processing. The implemented middleware system are consist of the reader interface for tag data collection, the event manager for a data filtering, and application interface for the RFID application. The implemented RFID middleware system is to support both a fixed type's reader and portable type's reader. we analyze the middleware function with four application accessing protocol, HTTP, XML, JMS, and SOAP, and demonstrate a filtering speed in terms of CPU utilization.

■ **keyword** : | RFID Middleware | Reader Interface | Event Manager | Application Interface |

---

## 1. 서론

RFID(Radio Frequency IDentification)가 다양한 응용 서비스에서 사용되기 위해서는 RFID 장치에 저장되어 있는 데이터를 적절한 장소와 적절한 시간에 응용 서비스로 전달하는 RFID 기반 미들웨어가 요구된다 [1-3]. RFID 기반 미들웨어가 정확한 데이터를 전달한

다는 것은 RFID 장치로부터 수집된 정보 중 응용 서비스가 관심 있는 데이터만을 필터링하여 전달하는 것을 의미하며, 데이터 필터링 기능은 데이터의 형식 및 응용 프로그램 환경에 따라 요구되는 기능이 달라진다. 단순한 EPC (Electronic Product Code)[2]를 활용하는 응용프로그램에서 필요한 정보를 얻는 방법과 훨씬 더 복잡한 구조를 갖는 데이터를 이용하는 응용프로그램

---

\* 본 연구는 2005년도 산업자원부 신기술실용화기술개발사업 연구비 지원에 의해 수행되었습니다.

에서 정보를 필터링하는 방법은 다르다. 또한 처리되어야 할 데이터의 양, 동시에 처리되는 필터링 조건의 수 등을 고려하여 데이터 처리 방법을 채택하여야만 데이터의 손실 없이 실시간 처리가 가능하다.

최근에 RFID 기반 미들웨어 제품 및 솔루션들이 많이 개발되고 있으나, 주로 EPC 코드 등과 같은 간단한 형식의 데이터를 처리하며, 대량의 데이터 처리에 대한 고려가 미진한 상태이다. 기 개발된 해외 또는 국내업체의 RFID 미들웨어가 단순히 비즈니스 애플리케이션과 핵심 기반구조 사이의 통신만을 지원하는 형태이다. 또한 최근에 EPCglobal에서는 미들웨어 인터페이스 표준으로 ALE(Application Level Event)을 제시하였다[3]. 이러한 ALE 기반 미들웨어들도 속속 소개되고 있다[4]. LGCNS의 RFID 미들웨어는 ALE를 기반으로 컴포넌트 기반 구조의 자바를 이용하여 높은 유연성과 확장성을 갖도록 하였으며[5], ETRI는 UbiCore라는 XML 기반의 미들웨어를 발표하였고[6], 한울은 센서 데이터 네트워크 미들웨어인 CrossOVER를 발표하였는데[7], 이는 다수의 센서를 통해 입력된 이벤트에 따라 연계된 정보 및 서비스를 자동으로 제공, 연동할 수 있도록 개발되었다. 레피아컴의 UltraALE는 ALE를 기반으로 개발된 최초의 RFID 미들웨어이다[8]. 또한 센서 및 RFID 정보에 기반한 상황인지형 미들웨어 기술도 최근 발표되었다[9].

본 논문에서는 기반구조의 확장성과 통합, 그리고 이 기종 리더로부터의 실시간/대용량으로 유입되는 데이터의 처리능력과 통합 프로세스관리를 지원하며, Java 기반의 XML을 지원하는 형태로 개발하여 보다 공통 데이터관리 모델을 지향하고자 한다. 또한 효율적이고 빠른 성능을 가능하게 하는 경량화기술에 기반하여 개발하였으며, 현재 개발된 시스템은 빠른 성능과 경량화 기능에 초점을 맞추어 개발되었으므로 ALE 기능은 지원하지 않는다. 현재 ALE 버전은 개발 중에 있어 조만간 본 시스템과 통합 예정이다.

본 논문은 서론에 이어 제2장에서는 RFID 미들웨어 구조에 대해 간략히 언급하고, 제3장에서는 본 논문에서 구현한 미들웨어 시스템의 기능별로 설명한다. 제4장에서는 필터링 데이터의 보안기술적응에 대해 기술

한다. 5장에서는 RFID 미들웨어의 기능에 대해 통합 테스트를 수행한다. 마지막으로 6장에서 결론을 맺는다.

## II. 구현된 RFID 미들웨어 시스템구조

### 1. 기본구조

본 RFID 미들웨어 시스템의 기본 구조는 그림에서처럼 실시간 스트림 태그 데이터를 읽어내는 리더인터페이스(RI : Reader Interface), 수집된 데이터를 의미 있는 데이터로 가공/필터링하는 이벤트 관리자(EM : Event Manager), EM은 데이터를 필터링 조건에 따라 의미있는 데이터로 가공하는데, 이는 Xquery 엔진을 사용하여 수행된다. 또한 결과 데이터를 RFID 응용프로그램으로 전달하기 위한 HTTP, JMS(Java Message Service), SOAP(Simple Object Access Protocol) 기반의 응용인터페이스(AI : Application Interface)로 구성된다.

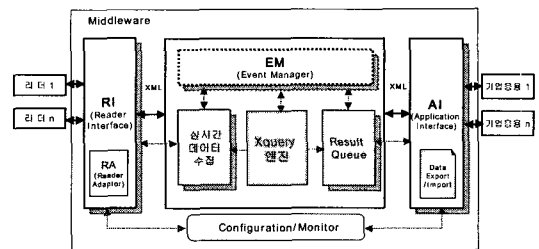


그림 1. RFID 미들웨어 구조도

### 2. 데이터처리 흐름

RFID 미들웨어의 데이터 흐름도는 [그림 2]와 같다. RI에서 리더를 통해 태그 데이터를 읽어 들이면, EM 모듈에서 필터링 및 이웃 모듈로의 전송을 수행하고, 이렇게 필터링된 데이터는 AI의 적절한 응용프로그램 접근 프로토콜을 거쳐 응용프로그램으로 전달된다.

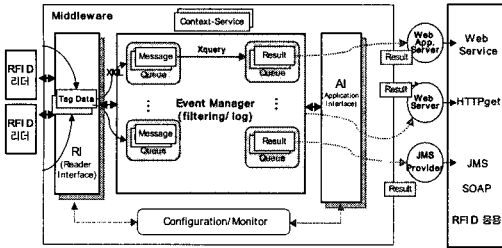


그림 2. RFID 미들웨어의 데이터처리 흐름도

두 접속 시도, 또는 선택 접속버튼을 통해 필요한 리더기만 접속 시도  
 ③ 리더기 상태 화면에서 리더기의 접속 상태 확인  
 다음은 리더 접속에 필요한 주요 프로그램 코드이다.

```
import com.alien.enterpriseRFID.reader.*;
import Matrics.*;
AlienClass1Reader reader = new AlienClass1Reader[v1];
reader[readerNo].open();
RFIDLibrary conn = new RFIDLibrary[v2];
conn[1].Open(ip,(short) port);
```

다음은 현재 설정되어 있는 리더의 정보를 변경하거나 추가에 관련된 화면이다. 리더의 IP주소나 포트 정보, 또는 추가되는 리더들을 등록할 수 있다.

### III. RFID 미들웨어 시스템의 기능

#### 1. 리더인터페이스(RI)

RI는 다양한 이기종 RFID 리더로부터 태그 데이터를 읽어서 이벤트 매니저로 전송하는 역할을 수행한다. 본 미들웨어의 RI에서는 어댑터 방식으로 다양한 장비를 지원하도록 구현하였는데, 현재 지원되는 리더 장비를 위한 어댑터로는 Alien 어댑터, Matrics 어댑터, Intermec PDA 어댑터, 국내 B사의 PDA 어댑터를 지원하도록 하였다.

아래 화면은 각 어댑터에서 동시에 태그 데이터를 읽을 수 있도록 RFID 미들웨어와 리더를 연동하는 RI 화면이다. 리더들은 모두 TCP/IP로 미들웨어에 연동되며, 화면에 미들웨어에 연결된 리더들의 상태와 IP(Internet Protocol)주소가 표시된다.

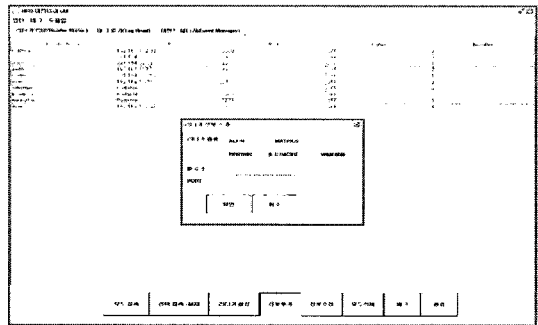


그림 4. 리더 정보추가 화면

#### (2) 태그 데이터 읽기

다음 화면은 리더인터페이스에 구현된 각각의 어댑터를 통하여 각 사의 리더기로부터 동시에 태그를 읽는 화면이다. 이 화면에는 읽은 태그 ID와 개수, 리더, 안테나 등 여러 관련 정보들이 실시간으로 표시된다.

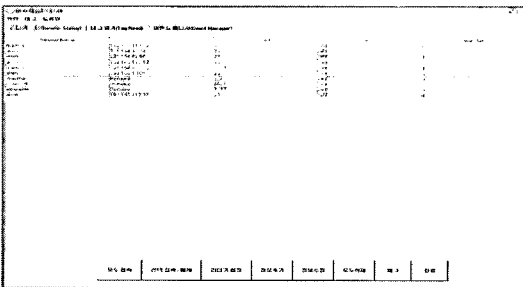


그림 3. RFID 미들웨어의 초기화면

#### (1) 리더 접속

- ① 정보추가, 수정, 삭제 버튼을 통해 리더기의 정보를 파일로 관리
- ② 모두 접속버튼으로 저장된 리더기 정보를 통해 모

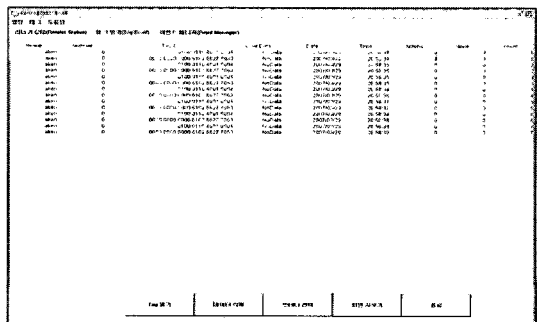


그림 5. 태그 읽기 화면

다음 그림은 핸드헬드 타입의 Intermec IP4 700 시리즈를 서버와 연동하여 서버의 RFID 미들웨어에서 태그 정보를 읽는 과정을 보여준다. PDA 타입의 리더에서 읽어들이는 태그 데이터는 실시간으로 미들웨어 서버로 전송된다.

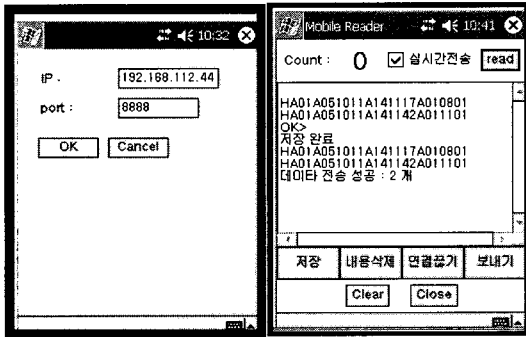


그림 6. Intermec IP4의 RFID 매니저 구동화면

다음은 리더를 통해 Tag 데이터를 읽어내는데 필요한 주요 프로그램 코드이다.

```
import com.alien.enterpriseRFID.tags.*;
import org.w3c.dom.*;
Tag[] tagList = reader[Num].getTagList();
result = conn[Num].X22ReadFullFieldCommand( (short)0x04,
                                             (short)0xA0);
GetIdOf = result.GetIdOf(hh);
TagID = ByteToString( RFIDLibrary.ReverseHex( GetIdOf ));
TransformerFactory transformerFactory =
    TransformerFactory.newInstance();
Transformer trans = transformerFactory.newTransformer();
trans.Transform(new DOMSource(doc), new StreamResult(new
    FileOutputstream(f)));
```

2. 이벤트 매니저(EM)

EM의 기능은 리더 인터페이스로부터 읽어들이는 태그 데이터를 가공/필터링하여 의미있는 정보로 응용 인터페이스에 전달하는 기능으로 클라이언트에서 들어오는 필터링 요구 컨텍스트에 따라 필터링 엔진에서 가공/처리하여 AI의 응용 프로그램 접근 프로토콜을 통해 응용 서비스 (또는 클라이언트)에게 전달해주는 기능이다.

(1) 이벤트 필터링 프로세스

EM 엔진의 필터링 과정은 다음과 같다.

- ① 클라이언트가 JSP(Java Server Page) 웹을 통해 서버에 접속 후 필터링 조건을 xml 파일로 저장
- ② 필터링 조건 버튼을 통해 저장된 조건 검색 가능
- ③ 저장된 필터링 조건을 실시간으로 선택하고 필터링하고, 필터링 결과를 관리하는 클라이언트 필터 매니저에서 여러 클라이언트와 동기화를 보장하면서 XML파일 처리
- ④ 선택된 조건에 대한 필터링 결과를 화면에 디스플레이 함.

다음 그림은 EM의 데이터 필터링 엔진 프로세스를 보여준다.

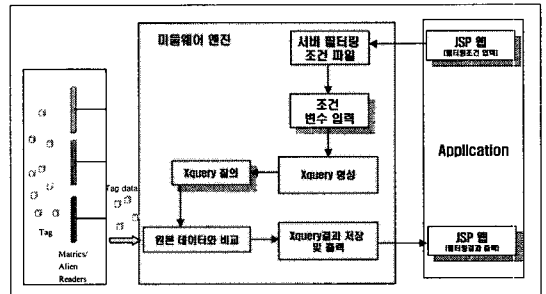


그림 7. EM에서 태그 데이터 필터링 과정

(2) EM의 이벤트 필터링을 위한 컨텍스트

본 EM 엔진에서 적용하는 필터링 컨텍스트 (필터링 변수)는 다음과 같다;

- ① 시간정보(time)
- ② 태그정보(Tag ID)
- ③ 리더기정보(리더기 #)
- ④ 안테나 정보(안테나 #)
- ⑤ 96비트 ID의 각 필드

다음은 이벤트 필터링 조건에 따라 수행된 필터링 결과를 보여주는 화면이다. 화면의 윗부분은 필터링조건이며, 아래는 조건에 따라 필터링된 결과이다. 필터링 조건은 응용 프로그램에서 해당 응용프로그램 접근 프

로토콜을 이용해 입력하도록 구성하였다. 화면의 필터링 조건의 예는 Alien 리더기로부터 2006.4.5~4.10 기간, 그리고 특정 태그 ID를 갖는 정보를 실시간으로 추출하도록 구현하여 실행한 화면이다.

ID	Reader	Number	Discovery Time	Read Count	
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 04	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 11	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 16	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 20	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 23	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 26	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 28	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 30	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 31	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 32	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 36	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 45	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 47	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 48	0

그림 8. 태그 ID에 따른 필터링

다음 그림은 96비트의 태그 ID중에서 특정 태그 필드에 따라 필터링한 그림이다.

ID	Reader	Number	Discovery Time	Read Count	
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 04	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 11	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 16	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 20	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 23	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 26	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 28	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 30	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 31	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 32	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 36	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 45	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 47	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 48	0

그림 9. 각 태그 ID의 필드에 따른 필터링

다음은 이벤트 매니저의 주요 프로그램 코드이다. Xquery 엔진 설정과 필터링 관련 부분들이 포함되어 있다.

```

import net.sf.saxon.om.*;
import net.sf.saxon.query.*;
import net.sf.saxon.value.*;

DocumentBuilderFactory dbf =
    DocumentBuilderFactory.newInstance();
DocumentBuilder db = dbf.newDocumentBuilder();
File docFile = new File("/xml/XMLfilter.xml");
Element root = doc.getDocumentElement();
Node nTag = root.getFirstChild();
nTag = nTag.getNextSibling();
DynamicQueryContext dynamicContext = new
    DynamicQueryContext(config);
Configuration config = new Configuration();
StaticQueryContext sqc = new StaticQueryContext(config);
XQueryExpression exp = sqc.compileQuery(xquery);

Properties props = new Properties();
props.setProperty(OutputKeys.METHOD, "xml");
props.setProperty(OutputKeys.INDENT, "no");
File fl = new File("rf54.xml");
exp.run(dynamicContext, new StreamResult(fl), props);
    
```

(3) EM의 이벤트 로그 기능

이벤트 로그는 현재 응용 프로그램에서 서버에 접속한 상황 및 이벤트 처리 과정 등의 로그 정보를 실시간으로 모니터링 해준다. 또한 각 RFID 응용 프로그램에서 마일웨어의 응용 인터페이스에 어떤 RPC(Remote Procedure Call)로 접속했는지에 대한 정보도 확인 가능하다.

ID	Reader	Number	Discovery Time	Read Count	
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 04	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 11	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 16	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 20	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 23	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 26	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 28	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 30	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 31	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 32	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 36	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 45	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 47	0
0000 0000 0000 0107 6627 7050	alien	0	2006040508	20 54 48	0

그림 10. 이벤트 로그 관리

다음은 이벤트 매니저의 로그 관리 기능에서 요구되는 주요한 프로그램 코드들이다.

```
String str = IP + ConnectDay + ConectTime + FilteringCount +
    QueryresultCount + RPC;
File fi = new File("log.txt");
if(fi.exists()) fi.createNewFile();
FileWriter fw = new FileWriter(fi, true);
fw.write(str);
```

### 3. 응용인터페이스(AI)용 클라이언트

클라이언트 기능은 미들웨어가 실행되고 있는 서버에 응용 프로그램이 접속하는 상황을 테스트하기 위해 구현되었다. 전체적인 로직은 JSP를 이용하여 구현하였으며, HTTP, JSP, SOAP, XML 등의 RPC를 지원하도록 구현 되었다. 클라이언트에서의 동작과정은 다음과 같다;

- ① 각 입력폼에 필터링 조건을 입력 후 필터링실행 (Filtering Run) 버튼 클릭으로 서버에 위치한 필터링 파일에 필터링 조건(필터링 변수)을 전달.
- ② 필터링조건보기 버튼을 통해 필터링 조건 확인가능.
- ③ 결과보기 버튼을 통해 필터링된 결과를 구현된 응용 접근 프로토콜을 통해 클라이언트로 가져옴.

다음 화면은 클라이언트에서 필터링 조건에 따라 미들웨어에서 응용 프로그램으로 필터링/가공된 데이터 결과를 전달받은 상황을 보여준다.

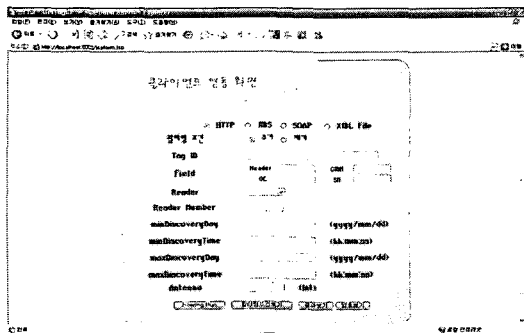


그림 11. 클라이언트 응용프로그램 접근 프로토콜 설정 및 필터링 조건 입력 화면

다음 화면은 필터링 결과를 서버에서 클라이언트로 읽어온 화면이다.

그림 12. 클라이언트로 전송된 필터링 결과 화면

### IV. 미들웨어 데이터의 보안

#### (1) RFID 미들웨어 시스템 보안

RFID 미들웨어 시스템 보안에서는 우선 미들웨어 관리자 모드에서 RFID 미들웨어를 관리해야 하므로, 인증 절차를 거쳐 인증된 관리자만이 미들웨어에 접속하도록 하였다.



그림 13. 로그인 화면

#### (2) RFID 미들웨어 데이터 보안

이벤트 매니저의 필터링 엔진을 거쳐 클라이언트로 전송되는 데이터는 중요한 가치를 지니므로 암호화 알고리즘을 사용하여 암호화 과정을 거쳐 전달된다.

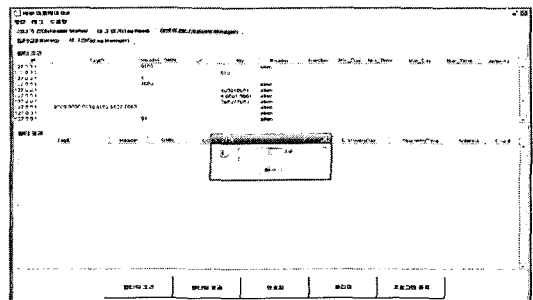


그림 14. 필터링 데이터의 암호화

다음 화면은 클라이언트에서 필터링 결과를 볼때 복호화 과정을 거쳐 필터링 데이터를 보는 화면이다.

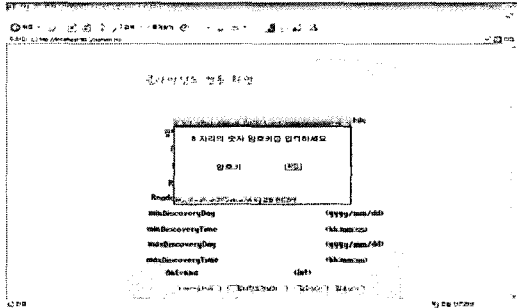


그림 15. 클라이언트에서 복호화 화면

### V. RFID 미들웨어 통합 테스트

다음 그림은 RFID 태그 데이터를 캡처하는 PDA 단말을 무선 LAN으로 서버와 연동한 후, RFID 미들웨어에서 수집된 데이터를 실시간으로 처리하는 통합 테스트 시나리오이며, 이 시나리오에 따라 기능 테스트를 수행하였다.

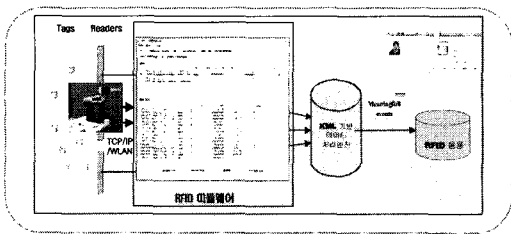


그림 16. 통합 테스트 시나리오

개발된 RFID 미들웨어서 기능 통합 테스트를 수행한 결과는 다음 표와 같이 정리할 수 있다. 각 4개사의 리더기를 지원하는 어댑터가 구현되었고, 필터링 속도는 초당 수천건이 가능하며, 지원되는 응용프로그램 접근 프로토콜은 아래의 4가지 형태의 RPC 들이 구현되었다.

표 1. RFID 미들웨어 기능 통합 테스트

회사(기종)	리더 인터페이스	필터링속도	응용 API
Alien	○	수천개/sec	HTTP, JSP, SOAP, XML
Matrics	○		
Intermec	○		
국내 B사	○		

다음 그래프 [그림 17]은 태그 개수를 100~1200개로 변경하면서 시간에 따른 CPU 이용율을 측정된 것이다. 이 그래프로 볼때 개발된 RFID 미들웨어는 초당 수천개의 태그 데이터 처리에도 안정된 성능을 보여준다.

또한 이기종 리더기로부터 실시간으로 대용량의 태그 데이터를 처리할 수 있는 기능을 갖는다. [그림 18]은 현재 국내외적으로 개발된 엔터프라이즈형 RFID 미들웨어의 기능을 보여준다.

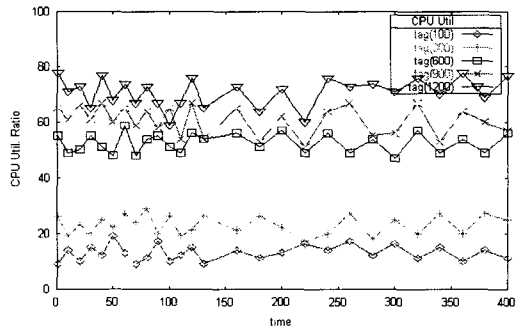


그림 17. 태그 개수에 따른 CPU 이용율 측정

분류	OAT System	SUN Java System RFID Software	Oracle Edge Server/Sensor Data Hub	Ours
지원 리더기 종류	Matrics, Alien, ThingMagic, SAMSys, AWID	Alien, Matrics, Intermec, ThingMagic	Alien, Intermec, lightstick	Alien, Intermec(PDA용), Matrics, 국내B사
데이터 형식	EPC	EPC	EPC	EPC
컨택스트 처리 (이벤트)	Predefined filter (SQL like)	Predefined filter	Predefined filter	Predefined filter
엔터프라이즈 연동	XML via File, JMS, http	File System, JMS, XML/HTTP/SOAP	Stream, JMS, Web Service, Http Post	HTTP, JMS, SOAP, XML
초당 데이터 처리	?	?	?	수천/sec

그림 18. 각 미들웨어 기능 비교표

## VI. 결론

본 논문에서 RFDI 미들웨어 시스템을 고유의 RFID 미들웨어 기능 및 대용량 데이터처리용 플랫폼으로 개발하였으며, 실시간 RFID 미들웨어 시스템을 위한 플랫폼 개발을 목표로 하여 연구를 수행하였다. 또한 인프라스트럭처의 확장성과 통합, 그리고 대용량의 데이터 처리와 통합프로세스관리를 지원하며, Java 기반의 XML을 지원하는 형태로 개발하여 보다 공통 데이터 관리 모델을 지향하고있으며, 효율적이고 빠른 성능을 가능하게 하는 경량화기술에 기반하여 개발하였으며, 현재 개발된 시스템은 빠른 성능과 경량화 기능에 초점을 맞추어 개발되었으므로 ALE 기능은 지원하지 않는다.

개발된 RFID 미들웨어 시스템은 리더인터페이스 이벤트 매니저, 응용 인터페이스의 3-tier 구조로 개발되었고, 이기종의 다양한 리더 어댑터를 제공하고 있다. 또한 클라이언트를 연동하기 위해 HTTP, JMS, SOAP, XML 등 여러 응용프로그램 접근 프로토콜을 사용할 수 있으며, 필터링 결과에 대한 암호기능도 제공하여 데이터 보안 기능을 강조하였다.

### 참고 문헌

- [1] EPCglobal, *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz*, Version 1.0.9, 2005.
- [2] EPCglobal, *Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz*, Version 1.0.9, 2005.
- [3] K. Traub, S. Bent, T. Osinski, S. N. Perertz, S. Rehlinh, S. Rosenthat, and B. Tracey, *The Allication Level Event(ALE) Specification*, ver1.0, 2005.

[4] 홍연미, 조윤상, 변지용, 노영식, 박상열, 오상현, 변영철, "ALE기반 RFID 미들웨어 시스템 설계", 한국콘텐츠학회 2006년 추계학술대회 논문집, 제4권, 제2호, pp.469-475, 2006.

[5] <http://www.lgcns.com>

[6] 이훈순, 최현화, 김병섭, 미명철, 박재홍, 이미영, 김명준, 진성일, "UbiCore : XML 기반 RFID 미들웨어 시스템", 한국정보과학회논문지:데이터베이스, 제33권, 제6호, pp.578-589, 2006.

[7] <http://www.hanwol.co.kr>

[8] <http://www.repia.com>

[9] T. S. Lopez and D. Y. Kim, "A Context Middleware Based on Sensor and RFID Information," Proc. of IEEE Percom'07, pp.331-336, 2007.

### 저자 소개

박 병 섭(ByoungSeob Park)

중신회원



- 1989년 2월 : 충북대 컴퓨터공학과(공학사)
- 1992년 2월 : 서강대학교 컴퓨터학과(공학석사)
- 1997년 2월 : 서강대학교 컴퓨터학과(공학박사)

• 1997년 4월 ~ 2000년 2월 : 국방과학연구소 선임연구원

• 2000년 3월 ~ 2002년 8월 : 우석대학교 컴퓨터교육과 교수

• 2002년 9월 ~ 현재 : 인하공업전문대학 컴퓨터시스템과 교수

<관심분야> : RFID/USN, Mobile-IPv6