Journal of the Society of Korea Industrial and Systems Engineering
Vol, 31, No. 3, pp.80－89, September 2008.

비즈니스 프로세스 로직 표현을 지원하는 RFID 미들웨어 개발

박철순* · 배성문**† · 고 로*

*창원대학교 산업시스템공학과
**경상대학교 산업시스템공학부/공학연구원

Development of RFID Middleware with Business Process Logic Representation Capability

Chul-Soon Park* · Sung-Moon Bae**† · Gao Lu*

*Dept. of Industrial and Systems Engineering in Changwon National University
**Dept. of Industrial and Systems Engineering in Gyeongsang National University/Engineering Research Institute

Because of different hardware specifications, there are no unified protocol commands to use with various kinds of RFID readers. The current commercial RFID middlewares do not satisfy the various requirements from users to support business process logic representation. The EPCglobal, which is leading organization for the RFID research, suggested a RFID middleware architecture which is called ALE(Application Level Events) standard. However, their architecture also does not provide the application level's interfaces. Therefore, a new RFID middleware architecture is required to provide basic RFID functions, conform to ALE's specification and, additionally, support application level's business logic representation.

This paper proposes a ALE-based RFID middleware architecture which provides business process logic representation. At first, the basic RFID control functionalities are identified. Secondly, the business process logic requirements in RFID applications are identified and classified into six categories. Third, the Middelware architecture is implemented with Java and XML technology so that it can easily extended to support the various RFID hardware's protocols. Finally, an example RFID prototype system is developed to show the proposed architecture's feasibility and validate it. The proposed middleware is expected to be used in various application areas since it is using XML technology for easy adaptation and it also conforms to ALE interface which is standard specification.

Keywords : RFID Middleware, Business Process Logic, RFID Protocol, ALE Specification, XML

1. 서 론

최근 공정 자동화, 공급망 관리 등과 같은 다양한 응용분야에서 RFID의 활용이 증가하고 있다[1]. 다양한 분야에서 RFID 기술은 가상의 IT공간과 실세계 사이에 존재하는 갭을 연결시켜 많은 문제점들을 해결해 주고 있다[4, 5]. 하지만, 원활한 RFID 시스템 연동을 위해서는 연관되어 있는 조직 내에서 RFID 하드웨어의 통일을 요구하며 연관 조직 간 관련 데이터의 통합을 요구하고 있다. 또한 사용 중인 RFID 하드웨어가 추가되거나 교체될 때 기존의 시스템에 연동시키기 위해서는 프로그램의 재작성 또는 수정을 요구하곤 하는데 이러한 작업은 많은 비용과 자원을 소요한다.

통상 RFID 하드웨어 벤더들은 자사의 하드웨어를 이

논문접수일 : 2008년 08월 21일 논문수정일 : 2008년 09월 03일 게재확정일 2008년 09월 03일
† 교신저자 bsm@gnu.ac.kr

용한 RFID 응용프로그램 개발을 지원하기 위해 하드웨어 프로토콜에 대한 기술문서와 함께 별도의 API(Application Programming Interface)를 제공해 주고 있다. 이런 API는 응용프로그램을 개발하는 시간을 줄여주고 복잡한 하드웨어 프로토콜에 대한 지식 없이도 쉽게 프로그램을 작성할 수 있게 한다.

하지만, 특정 하드웨어에 특화된 API를 사용하게 되면 응용프로그램이 특정 하드웨어에 종속되어 RFID 하드웨어가 변경되거나 새로운 하드웨어를 도입하는 경우 응용프로그램을 재작성 해야 하는 문제점이 발생할 수 있다. 더욱이 하드웨어에 따른 프로토콜 명령어 차이로 인해 모든 하드웨어에 대해 동일하게 사용가능한 통일된 프로토콜 명령체계는 존재하지 않고 있다. 따라서 현재 다양한 하드웨어들을 통합할 RFID 미들웨어의 필요성은 점점 더 증대되고 있다[2, 6].

한편 프로토콜 표준화의 필요성에 따라 현재 RFID에 대한 표준화 작업을 주도하고 있는 EPCglobal에서는 ALE (Application Level Events)라 불리는 RFID 미들웨어 표준 스펙을 제안하였다[3]. 하지만 이 스펙은 응용프로그램 작성 시 필요한 비즈니스 로직을 표현하기 위한 기능을 제공하지 않고 기본적인 하드웨어 관리, 데이터 필터링 및 중복데이터 처리 등의 핵심 기능만을 제안하고 있다. 따라서 ALE 표준스펙을 준수하면서 응용프로그램 수준에서 요구되는 비즈니스 로직을 표현할 수 있는 RFID 미들웨어 개발의 필요성이 증대되고 있다. 따라서 본 논문에서는 RFID 미들웨어가 제공해야 하는 기본적인 핵심기능을 제공하면서 ALE 표준스펙을 준수하고 비즈니스 프로세스 로직을 표현할 수 있는 새로운 미들웨어 아키텍처를 제시하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 먼저 제 2장에서는 RFID 미들웨어와 관련된 문헌조사를 통해 RFID 미들웨어가 제공해 주어야 하는 기능들을 규명하였다. 즉, RFID 미들웨어가 제공해 주어야 하는 기본적인 핵심 기능 및 비즈니스 프로세스 로직에 대한 요구사항을 분석하고 정리하였다. 제 3장에서는 본 연구에서 제시하는 RFID 미들웨어의 아키텍처에 대해 알아보았다. 제 4장에서는 프로토타입 시스템 구현과 예제를 통해 본 연구에서 제시하고 있는 미들웨어 아키텍처를 검증하고 실현가능성을 보여주었다. 마지막으로 제 5장에서는 추후 연구과제에 대한 언급과 함께 결론을 맺고 있다.

2. RFID 미들웨어 요구사항 분석

통상 RFID 리더기와 호스트 컴퓨터사이의 통신방식은 RS-232C 기반의 직렬통신과 소켓을 기반으로 한 LAN 통신 방식 등이 있다. 원활한 통신을 위해서는 RFID 하드웨어 벤더에서 제공해 주는 통신 프로토콜을 따라야 한다. <그림 1>은 두 가지 RFID 하드웨어에 대한 통신 프로토콜의 일부분을 보여주고 있다[7, 12].

명 령	Host to Reader (request)	Reader to Host (response)
다중읽기	04 00 40 FF	0C FG DSFID UID FF
시스템 정보 얻기	04 02 2B FF	11 FG IF UID DSFID AFI BSN NBB IMC FF
태그쓰기	09 02(42) 21 BN WD FF	03 FG FF

(a) FirmSys(r) Reader Protocol (13.56 MHz)

명령	Host to Reader (request)	Reader to Host (response)
다중읽기	01 04 06 22 A0 CRC CRC	01 04 Length 22 01 A0 NumTags Data CRC CRC
시스템 정보 얻기	01 04 05 14 CRC CRC	01 04 26 14 00 05 CRC CRC
태그쓰기	01 04 Len 33 A0 10 00 ID CRC CRC	01 04 06 33 00 CRC CRC

(b) Symbol(r) AR-400 Protocol (900MHz)

<그림 1> RFID 하드웨어 프로토콜 예

여기서 두 하드웨어의 통신 프로토콜은 구조가 유사함을 알 수 있다. 즉, 각 명령들은 16진수 형태의 바이트 스트림으로 이루어져 있으며 각 스트림은 시작, 종료, 명령구분, 하드웨어정보, 그리고 체크 비트 등과 같이 유사한 패턴으로 구성되어 있다. 하지만, 하드웨어에 따라 각 항목에 할당되는 값이 다름을 알 수 있다. 따라서 RFID 시스템에서 사용 중인 하드웨어를 교체하거나 새로 추가할 경우에는 프로그램 코드를 수정해야 한다. 즉, RFID 미들웨어는 다양한 벤더에서 제공하는 하드웨어에 대처할 수 있도록 프로토콜 설정에 대한 기능을 포함하고 있어야 한다. 이러한 요구사항을 위해 통신 프로토콜은 플랫폼에 중립적인 XML과 같은 언어를 이용해서 설계되어야 한다. RFID 미들웨어는 이러한 RFID 하드웨어의 프로토콜을 기반으로 다양한 종류의 리더기를 관리할 수 있어야 한다. 미들웨어에서 기본적으로 제공해야 하는 기능은 읽기, 쓰기, 다중태그 읽기(anti-collision), 시스템정보 읽기 등이 있다[2, 6, 8].

한편, 2005년 RFID에 대한 표준화 연구를 주도하고 있는 EPCglobal에서는 RFID 하드웨어와 응용프로그램 층을 연결하는 RFID 미들웨어에 대한 인터페이스의 통일을 위해 ALE 표준스펙을 제안하였다[3]. 이 스펙에 따르면 미들웨어는 지속적 읽기 시간설정, 끊어 읽기 간격설정, 읽지 않는 시간 간격설정 등과 같이 하드웨어를

통제하는 기능을 제공해야 한다. 한편 태그가 읽혀지면 RFID 미들웨어는 필터링 또는 그룹화 등과 같은 통계처리를 통해 데이터를 가공할 수 있는 기능을 제공해야 한다. 이러한 ALE 스펙을 준수하는 RFID 미들웨어는 이 스펙을 준수하는 어떤 RFID 응용프로그램에서도 사용될 수 있게 된다.

현재 RFID 시장에는 몇 가지의 RFID 미들웨어 제품이 출시되어 있다[8, 9, 10, 11, 13]. 이들 제품은 주로 웹 환경에서 RFID 시스템과 자사의 전사적 시스템과의 통합에 초점을 맞추고 있다. 즉, 이들 제품은 기능측면에서는 앞에서 언급한 기본적 핵심 기능 외에 RFID 데이터 필터링 및 간단한 통계기법 등이 기능만을 제공하고 있다.

따라서 RFID 미들웨어는 위에서 언급한 기본 기능과 EPCglobal에서 표준으로 제안하고 있는 ALE 계층의 기능을 제공해야 한다. <표 1>은 미들웨어에서 하드웨어 관리를 위해 제공해야 할 기능을 보여주고 있다. 이 계층에는 읽기, 쓰기, 다중태그읽기(anti-collision), 시스템 정보읽기 등과 같은 기본관리 기능과 통신 관리를 위한 기능을 제공할 필요가 있다. 한편 <표 2>에는 ALE 계층에서 지원해야할 기능을 보여주고 있다. ALE 계층에서는 RFID 하드웨어를 통제하기 위한 고급기능 및 통계처리에 대한 기능들을 제공해야 한다.

<표 1> 미들웨어 하드웨어 계층 기능

하드웨어 계층	기 능
기본관리	태그 읽기 다중태구 읽기 태그 쓰기 시스템정보 획득 안테나신호세기 조절 리더 정지
통신관리	통신 파라미터 설정 직렬포트 연결 소켓 연결 데이터 송신 데이터 수신

위에서 언급된 요구사항들은 RFID 미들웨어에서 제공해야 할 기본적인 핵심기능들이다. 하지만 통상 RFID 응용 프로그램에서는 단순하게 물체의 ID 인식 이외에 물체에 관련된 비즈니스 로직 정보들을 사용할 필요가 있다. 예를 들면, 자산관리시스템에서는 물체의 ID 인식 자체 보다는 위치차원의 정보가 중요하며 매장관리시스템에서는 진열된 상품에 대한 유통기한 같은 시간정보가 중요하다. 하지만 위에서 언급한 RFID의 기본적인 기능만으로는 이러한 응용프로그램에서 필요한 정보를 얻는 것이 쉽지 않다. 따라서 RFID 미들웨어에서 이러한 비즈니스 프로세스 차원에서의 정보를 제공해 줄 수 있어야 한다.

<표 2> ALE 계층 기능

ALE 계층	기 능
읽기 관리	트리거 관리 지속적 읽기기간 설정 끊어 읽기기간 설정 태그를 읽지 않는 기간 설정
통계 관리	데이터 누계 패턴필더 관리 그룹화 계수 부가정보 읽은회수

본 연구에서는 문헌조사와 연구경험을 바탕으로 RFID 시스템에서 필요한 비즈니스 프로세스 로직 표현을 위한 기능들을 파악하여 이들을 위치, 시간, 수량, 방향, 속도, 안전과 같이 여섯 개의 카테고리로 분류하였다. 각각의 카테고리에 속하는 기능들은 <표 3>과 같다.

<표 3> 비즈니스 로직 표현 요구사항

Category	Requirements
Location	특정 시점의 예상위치 계산 현재시점의 위치 계산 예상위치에서 벗어나는 예외상황 물체의 포함관계 계산 특정물체가 특정위치에 있었는지 계산 특정물체의 과거위치 추적
Quantity	물체수량 합계 계산 특정 위치에서 특정 리더기와 관련된 물체수량 계산 특정위치의 물체재고 역치값 설정 물체재고량 실시간 추적을 통한 예외사항 파악 특정 시간대의 특정위치의 수량 계산
Time	특정물체의 제조일자 및 유효기한 설정 유효기간 예외사항 체크 특정물체의 특정위치 체류기간 계산 특정위치에서 다른 위치까지의 이송시간 계산
Direction	움직이는 물체의 방향 계산 물체가 통과한 Route 정보 계산
Speed	이송물체의 최대허용속도 설정 이송물체의 최소허용속도 설정 이송물체의 현재속도 계산 최대속도 위반 체크 최소속고 위반 체크
Security	특정물체가 해당위치에 있는지 체크 특정물체간의 혼적/혼합 여부 체크

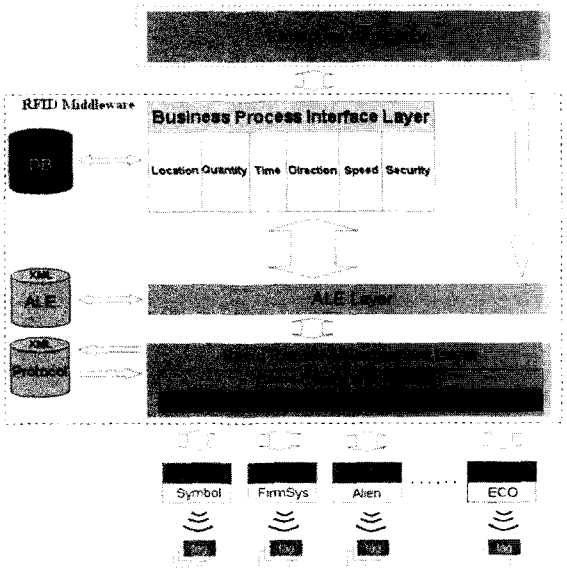
먼저, 위치(location) 카테고리는 물체의 위치를 파악하기 위한 기능을 포함하고 있다. 둘째, 수량(Quantity) 카테고리는 물체의 수량에 대한 로직을 포함하고 있다. 셋

째, 시간(Time) 카테고리는 시간에 관련된 정보를 표현할 수 있도록 한다. 넷째, 방향(direction) 카테고리는 물체의 이동 방향을 표현할 수 있도록 한다. 다섯째, 속도(speed) 카테고리는 물체의 이동속도에 대한 정보를 표현할 수 있다. 마지막으로, 안전(security) 카테고리는 안전여부를 표현할 수 있다.

따라서 본 연구에서는 RFID 하드웨어에 대한 기본적인 관리기능을 지원하면서 EPCglobal에서 표준으로 제안한 ALE 규약을 준수하고, 다양한 비즈니스 프로세스 상에서 요구되는 기능들을 지원할 수 있는 RFID 미들웨어 아키텍처를 지원하는 제품이 필요함을 알 수 있다.

3. RFID 미들웨어 아키텍처

지금까지 앞장에서는 RFID 미들웨어에서 지원해 주어야 할 기능들에 대해 알아보았다. 먼저 하드웨어 통제를 위한 기본적인 기능, EPCglobal에서 제시한 ALE 표준규약, 그리고 비즈니스 프로세스 로직 표현을 위한 기능들에 대해 알아보았다. 이번 장에서는 앞에서 규명한 기능들을 제공할 수 있는 RFID 미들웨어 아키텍처를 제시 한다. <그림 2>는 본 연구에서 제시하는 RFID 미들웨어 아키텍처를 보여주고 있다. 본 아키텍처는 기본적 리더기 관리 계층, ALE 계층, 비즈니스 프로세스 인터페이스 계층과 같이 세 개의 계층으로 구성되어 있다.

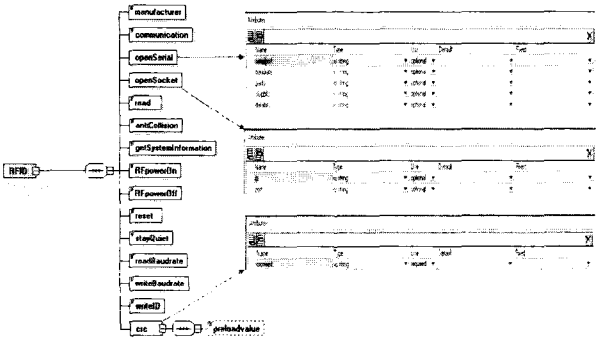


<그림 2> RFID 미들웨어 아키텍처 제안

3.1 Basic Reader 관리계층

본 연구에서 제시하고 있는 미들웨어 아키텍처의 최하단 층은 기본리더관리 계층(Basic Reader Management Layer)으로 기본리더 기능(Basic Reader Functionality), 예외처리와 핵심기능을 포함하는 핵심 기능(Core Functionality) 및 XML 프로토콜 인터페이스 저장소와 같이 세 개의 모듈로 구성되어 있다.

첫째, XML 프로토콜 인터페이스에는 RFID 하드웨어의 특성값 및 기능을 관리하기 위한 프로토콜 파라미터를 저장할 수 있다. RFID 하드웨어에서 제공하는 대표적인 프로토콜 명령은 읽기, 쓰기, 다중태그읽기(anti-collision) 등이 있다. 본 연구에서는 이러한 프로토콜 정보를 표현하기 위해 <그림 3>과 같이 XML Schema를 설계하였다.



<그림 3> 프로토콜에 대한 XML Schema

XML 기반의 프로토콜 문서는 XML의 특성에 따라 프로토콜 정보의 내용체크가 수월하고 쉽게 확장할 수 있으며 Java등의 프로그래밍 언어에서 XML 문서를 다룰 수 있는 풍부한 인터페이스를 제공하고 있다. 위에서 제시된 XML Schema를 사용하여 <그림 4>와 같은 XML 프로토콜 파일을 작성할 수 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<RFID xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Gaoiuiu\Desktop\RFIDXML3\RFID.xsd">
  <manufacturer>FirmSys</manufacturer>
  <communication>SERIAL</communication>
  <openSerial databits="DATABITS_8" serialport="COM1" stopbits="STOPBITS_1" parity="PARITY_NONE" baudrate="9600"/>
  <openSocket port="" ip=""/>
  <read>05260100FF</read>
  <antiCollision>040040FF</antiCollision>
  <getSystemInformation>04022SFF</getSystemInformation>
  <RFpowerOn>04006AFF</RFpowerOn>
  <RFpowerOff>04008BFF</RFpowerOff>
  <reset>040226FF</reset>
  <stayQuiet>0C2202FF</stayQuiet>
  <readBaudrate>040080FF</readBaudrate>
  <writeBaudrate>06008101FF</writeBaudrate>
  <writeID/>
  <crc crcneed="NO"/>
</RFID>
```

<그림 4> XML 프로토콜 문서 파일

위의 프로토콜 파일은 FirmSys(r) RFID 장비에 대한 프로토콜 설정 정보를 보여주고 있다. 위의 파일에서는 통신방식 및 관련 통신 파라미터를 설정하고 있다. 또한 읽기, Anti-Collision 등에 사용하는 하드웨어 프로토콜 파

라미터를 설정해 주고 있다. 이와 같이 제공되는 프로토콜 정보는 DOM 파서와 같은 프로그래밍 툴을 사용해서 미들웨어의 최하위 모듈에서 하드웨어 관리에 사용한다.

둘째, 핵심기능(Core Functionality) 모듈은 미들웨어 내부의 다른 모듈에서 사용하는 핵심기능을 포함하고 있다. 이 기능들은 직렬 또는 소켓을 이용한 연결, 리더기에 명령을 보내고 응답 얻기 등의 기능을 수행한다. 또한, 핵심기능 모듈에는 전송 시 에러를 감지할 수 있는 에러체크 기능을 포함하고 있다.

셋째, 기본리더기능(Basic Reader Functionality) 모듈은 RFID 리더기 관리를 위한 기능들을 포함하고 있다. 이 모듈에는 태그 읽기, 태그 쓰기, Anti-Collision 등과 같은 RFID 장비를 실제로 통제할 때 사용할 수 있는 기능들을 포함하고 있다.

3.2 ALE 관리 계층

ALE 규약은 응용프로그램에 대한 RFID 인터페이스를 정의하기 위해 2005년 EPCglobal에 의해 제안되었다. RFID 리더기가 작동하는 동안에 태그는 계속해서 읽혀지므로 설정된 시간간격 동안에 한번이상 읽혀질 수 있다. 따라서 읽혀진 태그정보가 직접 응용프로그램으로 전달되면 시스템 자원을 낭비할 수 있다. 따라서 이와 같은 태그 정보를 수집해서 필터링한 후 유용한 정보를 가공할 수 있는 기능이 필요해 지는데 이와 같은 기능을 수행하는 것이 ALE 계층의 역할이다. <그림 5>에서 EPCglobal에서 제시하고 있는 ALE 인터페이스를 보여주고 있다.

```
define (specName : string, spec : ECSpec) : void
undefine (specName : string) : void
getECSpec (specName : string) : ECSpec
getECSpecNames () : List  // returns a list of specNames as strings
subscribe (specName : string, notificationURI : string) : void
unsubscribe (specName : string, notificationURI : string) : void
poll (specName : string) : ECReports
immediate (spec : ECSpec) : EcCReports
getSubscribers (specName : String) : List  // of notificatioin URIs
getStandardVersion () : string
getVendorVersion () : string
<< extension point >>
```

<그림 5> EPCglobal에서 제안한 ALE 인터페이스 예

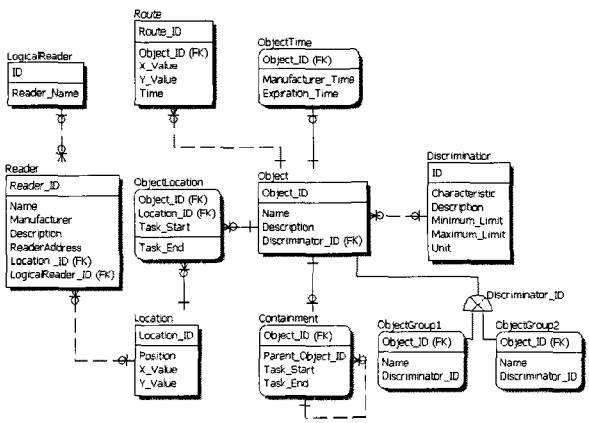
ALE에서는 가공된 정보를 응용프로그램 또는 상위층에서 사용할 수 있도록 전달하기 위해서는 XML 파일의 형태로 저장하는 방법(subscribe 인터페이스 사용)과 ECReport 타입의 변수에 가공된 정보를 채워서 응용프로그램으로 즉시 전달(immediate 인터페이스 사용)하는 방법을 사용할 수 있다.

ALE를 사용함으로써 RFID 시스템은 태그를 읽어 들일 리더기, 읽어 들일 시간간격, 데이터를 필터링할 조건, 누계와 같은 수집할 통계치 종류 등을 설정할 수 있다. 하지만, EPCglobal에서는 ALE에 대한 상세 구현에 대한 언급을 제시하지는 않고 인터페이스 명칭만을 제안하고 있다. 따라서 RFID 미들웨어와 같은 RFID 응용프로그램을 작성하는 당사자가 직접 ALE 인터페이스를 구현한 후 사용해야 한다. 더욱이, EPCglobal에서는 ALE 처리를 어느 단계에서 수행해야 할지에 대해 제시하지 않고 있다.

3.3 비즈니스 프로세스 인터페이스 계층

초기의 RFID 미들웨어 개발의 목적은 주로 상이한 RFID 하드웨어들을 통합하는 인터페이스를 구축하여 RFID 하드웨어간의 데이터를 연동시키는데 초점을 맞추었다[2]. 하지만 RFID의 적용분야가 점점 확대되고 RFID를 하나의 시스템 솔루션으로 활용하려는 추세에 따라 많은 비즈니스 규칙들을 RFID 미들웨어에 포함시킬 필요성이 생겼다. 본 연구에서는 문헌조사 등을 통해서 비즈니스 프로세스 인터페이스들을 파악하여 이를 위치(location), 수량(quantity), 시간(time), 방향(direction), 속도(speed) 및 안전(security)과 같이 여섯 개의 카테고리로 분류하였다.

먼저 앞에서 언급된 비즈니스 프로세스 인터페이스들을 구현하기 위해 <그림 6>과 같은 데이터베이스 스키마를 디자인 하였다. 이 데이터베이스 스키마에는 *LogicalReader, Reader, Object, Location, Discriminator, ObjectLocation, ObjectTime, Containment, Route*와 같은 엔터티들이 있는데 이를 이용해서 비즈니스 프로세스 인터페이스 구현에 필요한 정보를 표현할 수 있다.



<그림 6> 비즈니스 프로세스 인터페이스를 위한 Database Schema

3.3.1 위치(Location) 카테고리 인터페이스

위치(Location) 카테고리는 물체의 위치정보 요구사항에 대한 인터페이스를 제공한다. 예를 들면, 특정 팔레트의 위치를 파악할 수 있다. 이 카테고리에서는 물건의 위치파악에 대한 정보를 제공하기 위해 아래와 같이 여섯 개의 인터페이스를 제공한다. 첫째, *get_expected_location* 인터페이스는 관심 있는 물체에 대해 정위치를 찾기 위해 사용한다. 즉, 만일 물체 '0000001'이 현재 있어야 할 위치를 알고 싶다고 가정하자. 현재 <그림 7(a)> 의 ObjectLocation 테이블을 살펴보면 해당 물체는 L1 즉, Area 1의 위치에 입고 후 출고가 이루지지지 않았다. 따라서 해당 물체는 Area 1에 있어야 한다.

Object_ID	Location_ID	Task_Start	Task_End
0000001	L1	2008-06-27 10:00:00	
0000002	L1	2008-06-27 09:00:00	
0000003	L1	2008-06-27 10:30:00	
0000004	L1	2008-06-27 08:00:00	
0000005	L2	2008-06-27 09:20:00	

(a)ObjectLocation 테이블

Location_ID	Position	X_Value	Y_Value
L1	Area 1		
L2	Area 2		

(b) Location 테이블

<그림 7> Get_expected_location 예제

둘째, *get_current_location* 인터페이스는 현재 해당 물체의 위치를 찾기 위해 사용한다. 이 정보도 마찬가지로 위의 ObjectLocation 테이블을 이용해서 쉽게 찾을 수 있다. 셋째, *check_location_exception* 인터페이스는 해당 물체가 정확한 위치에 있는지 아닌지를 판단할 때 사용한다. 이 인터페이스는 앞의 get_current_location과 get_expected_location 인터페이스의 결과를 비교하여 구할 수 있다. 넷째, *query_object_containment* 인터페이스는 물체들 사이의 포함관계를 표시해 준다. 즉, 특정 박스나 팔레트에 어떤 물체가 포함되어 있는지를 보여준다. 다섯째, *query_object_identification* 인터페이스는 어떤 물건이 특정한 위치에 있었는지 여부를 판단할 때 사용한다. 이 인터페이스는 ObjectLocation 테이블을 사용하는데 이 테이블에서 관심 있는 물체의 입고시간(Task_Start)과 출고시간(Task_End)이 모두 값을 가지고 있는 경우 해당 위치에 있었다고 볼 수 있다. 마지막으로 *get_location_history* 인터페이스는 특정 물체의 위치 히스토리를 얻을 때 사용한다.

3.3.2 수량(Quantity) 카테고리 인터페이스

수량 카테고리 인터페이스는 물체의 수량 집계 및 관리 요구사항에 대한 정보를 제공해 준다. 이 카테고리에는 아래와 같이 물건의 수량과 관련된 다섯 개의 인터페이스를 제공해 주고 있다.

첫째, *get_total_quantity* 인터페이스는 RFID 시스템 영역 내에 있는 물체의 합계를 구해준다. 둘째, *get_quantity_in_specific_location* 인터페이스는 앞의 get_total_quantity 인터페이스와 유사하지만 특정 위치에 있는 리더기와 관련된 물체의 수량을 구해준다. 셋째, *set_threshold_quantity* 인터페이스는 특정 위치의 재고 역치를 설정할 때 사용한다. 넷째, *check_threshold_exception* 인터페이스는 RFID 시스템 영역내의 재고 수량이 설정된 역치 아래로 내려갈 때 알려준다. 다섯째, *get_temporal_aggregation* 인터페이스는 주어진 시간대에서 특정 위치의 수량을 나타내준다.

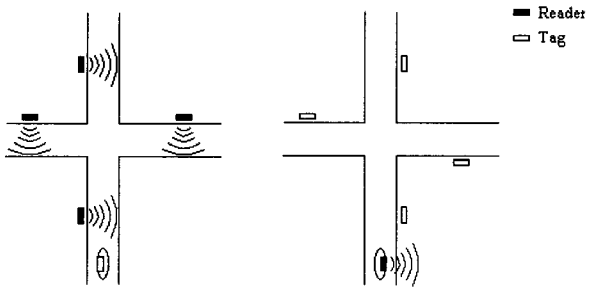
3.3.3 시간(Time) 카테고리 인터페이스

시간 카테고리 인터페이스는 시간에 관련된 요구사항에 대한 정보를 표현하기 위해 아래와 같이 네 개의 인터페이스를 제공하고 있다. 첫째, *set_product_time* 인터페이스는 RFID 시스템 영역에 있는 물체에 대한 제조일자와 유효기간을 ObjectTime 테이블에 설정할 때 사용된다. 둘째, *check_expiration* 인터페이스는 물체에 대한 유효기간을 체크하기 위해 사용한다. 셋째, *get_dwell_time* 인터페이스는 해당 물건이 특정 위치에 얼마나 오래 체류했는가에 대한 정보를 제공한다. 넷째, *get_object_moving_time* 인터페이스는 한 위치에서 다른 위치로 옮겨지는데 걸린 시간을 알려준다. 이 인터페이스를 이용하면 물건이 출고되어 판매되는 순간까지의 시간을 알 수 있다.

3.3.4 방향(Direction) 카테고리 인터페이스

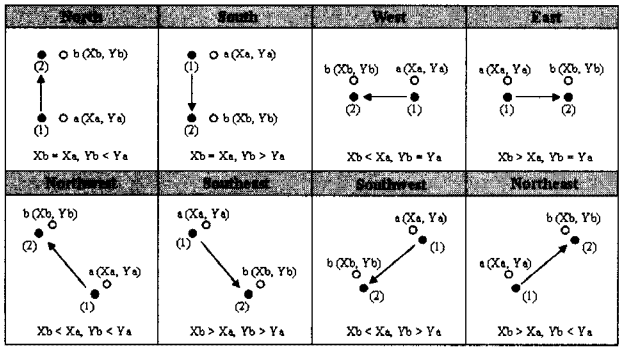
때로 움직이는 물체의 방향 정보와 루트 정보가 필요할 때가 있다. 이 카테고리는 물체의 움직임을 추적하는 상황에서 사용될 수 있다. 이 카테고리에 대한 아래와 같이 두 개의 인터페이스를 제공해 주고 있다. 먼저, *get_direction* 인터페이스는 RFID 시스템 영역에서 움직이는 물체의 방향을 판단할 때 사용된다. <그림 8>은 두 가지 방식의 방향 탐색 방법을 보여주고 있다. 즉, <그림 8(a)>는 물건에 태그를 붙여서 이동을 감지하는 방안이고 <그림 8(b)>는 물건에 리더기를 붙여서 이동을 감지하는 경우이다. 전자의 방법은 리더기의 비용이 고가이거나 무게가 무거울 때 사용할 수 있고, 후자는 반대의 경우에 적용할 수 있다. 이때 RFID 시스템 영역의 좌측상단을 원점으로 하면 <그림 9>와 같은 방법에

의해 이동방향을 구할 수 있다. 예를 들면, <그림 9>의 좌측상단의 경우 a 지점에서 b 지점으로 이동한 예를 보여주고 있다. 이때 Xa = Xb이고 Ya < Yb이므로 북쪽으로 물체가 이동했음을 알 수 있다. 둘째, *get_route* 인터페이스는 get_direction 인터페이스와 유사한데 차이점은 물체가 통과한 전체 루트 정보를 구해준다.



(a) Object attached with tag (b) Object attached with reader

<그림 8> 두 가지 방식의 방향 탐색 방법



<그림 9> 방향을 판단하는 알고리즘

3.3.5 속도(Speed) 카테고리 인터페이스

속도 카테고리 인터페이스는 움직이는 물체의 속도에 대한 정보를 표시하기 위해 사용한다. RFID 응용프로그램에서 속도가 중요한 이유 중 하나는 물체의 이동에 대한 효율을 분석할 수 있다. 속도 카테고리에서는 아래와 같이 다섯 개의 인터페이스를 제공해 주고 있다. 첫째, *set_max_speed_limit* 인터페이스는 허용 가능한 최고 이송속도를 설정하기 위해 사용하며 *set_min_speed_limit* 인터페이스는 허용 가능한 최소 이송속도를 설정하기 위해 사용한다. 둘째, *get_speed* 인터페이스는 두 지점 사이의 이송에 소요된 시간을 이용해서 속도를 계산할 수 있다. 셋째, *check_overspeed*와 *check_underspeed*는 현재 속도를 계산한 후 과속 및 저속 여부를 판단할 수 있게 한다.

3.3.6 안전(Security) 카테고리 인터페이스

안전 카테고리 인터페이스는 물체들의 특성 값을 이용하여 안전과 관련된 사항을 제시한다. 안전 카테고리 에는 아래와 같이 두 개의 인터페이스를 제공해 주고 있다. 먼저, *detect_missing_object* 인터페이스는 물체가RFID 영역 내에 현재 위치하고 있는지의 여부를 체크할 수 있도록 한다. 둘째, *check_mutual_exclusion* 인터페이스는 물체의 특성 값을 이용해서 혼적이 불가한 경우 미리 파악하여 대처할 수 있도록 한다.

3.4 Commercial 미들웨어 제품과의 비교

본 연구에서 제안하고 있는 미들웨어 제품과 현재 시장에 출시되어 있는 미들웨어 제품과의 기능적인 측면에서 비교를 해 보았다. <표 4>에는 비교결과를 보여주고 있다.

<표 4> 제안 미들웨어의 기능 비교

기 능	제안 미들웨어	타사 미들웨어 제품
Reader management	◦ XML기반의 하드웨어 프로토콜 기능지원 => 다양한 하드웨어 지원 가능 ◦ ALE 차원에서의 RFID 데이터에 대한 XML Report 기능 지원	◦ Oracle Sensor Edge Server와 Sun Java System RFID Software의 경우 RFID 데이터의 통합을 위해 XML 지원
ALE layer support	◦ ALE layer 완벽지원	◦ IBM WebSphere RFID Device Infrastructure는 ALE지원 ; UCLA WinRFID 미지원.
Application interface adaptability	◦ 다양한 응용분야에 사용할 수 있는 Business Process Interface 지원 => 특징임	◦ OAT Foundation Suite의 경우 SCM 분야 지원 ; WebSphere RFID Premises Server의 경우 자산관리에 초점
Business rule integration	◦ 자체 DB에 근거한 비즈니스 프로세스 차원에서의 기능 지원	◦ WinRFID와 Sun Java System RFID Software의 경우 data dispatching, filtering, 및 reader 관리기능 지원
Extendibility	◦ Java와 XML 기반이므로 확장성 뛰어남	N/A

위의 표를 보면 본 연구에서 제안한 미들웨어가 여러 측면에서 기존의 미들웨어에 비해 우수함을 알 수 있다. 즉, 기존 미들웨어의 경우 기본적인 리더기 관리, 데이터 필터링, 데이터 교환, 자사의 전사적 시스템과의 통합 등에 초점이 맞추어져 있음을 알 수 있다. 하지만 본 연구에서 제안하고 있는 미들웨어의 경우 Java와 XML 기반으로 설계되어 있기 때문에 플랫폼에 중립적이며 확장성이 우수함을 알 수 있다. 또한, 본 연구에서 제안하는 미들웨어의 특징은 EPCglobal에서 제안하고 있는 ALE 인터페이스 규약을 준수하기 위해 ALE 계층을 포함하는 세 개의 층으로 구성된 미들웨어 아키텍처를 제안하고 있다. 먼저 최 하단 층에서는 리더기 관리

에 대한 기본적인 기능만을 제공하는 리더기 관리 계층을 두고 있고, 두 번째 계층에서는 아랫단의 리더기 관리계층을 기반으로 ALE 계층을 구현하고 있으며 마지막으로 최상위 층에서는 비즈니스 프로세스 인터페이스를 위한 계층을 두어 미들웨어 아키텍처를 구성하고 있다. 이와 같은 구성을 통해서 RFID 응용프로그램을 작성할 때 유연하게 대처할 수 있다. 즉, 만일 RFID 리더기로부터 전송된 데이터를 직접 세밀하게 관리하고 싶을 때는 최 하단의 기본 모듈을 사용하면 되며, 만일 프로그래머가 ALE 규약을 준수하도록 프로그램을 작성하고 싶을 때는 ALE 층의 인터페이스를 사용하면 되며, 마지막으로 비즈니스 프로세스 레벨에서의 RFID 로직이 필요할 때는 최상단의 비즈니스 프로세스 계층에서 제공하는 인터페이스를 사용하면 된다.

4. 프로토타입 시스템 구축

이번 장에서는 앞장에서 제시한 미들웨어 아키텍처의 실현가능성을 보여주고 검증하기 위해서 프로토타입 시스템을 구현하였다. 제 4.1절에서는 ALE 구현에 대해서 설명한 후 제 4.2절에서는 비즈니스 프로세스 인터페이스 계층을 활용한 기자재 관리 시스템 예제를 보여준다.

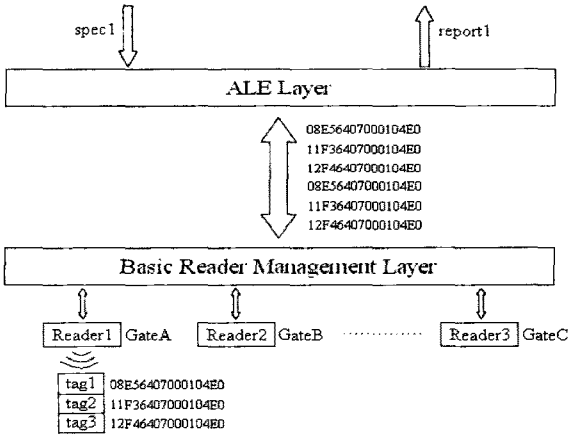
4.1 ALE 구현

앞장에서 설명한 것처럼 EPCglobal에서는 RFID 미들웨어 제품에 대한 ALE 인터페이스 계층을 규정하고 있다. 이 규정에서는 RFID 응용프로그램에 대한 인터페이스와 입력(ECSpec) 및 출력 데이터 타입(ECReports)을 명시하고 있다. 따라서 ALE 규약을 준수하는 모든 RFID 미들웨어는 ALE 규약을 따르는 RFID 응용프로그램에서 활용될 수 있다. 하지만 EPCglobal에서는 ALE 인터페이스 명칭만을 제시해 주고 있을 뿐 실제로 구현한 프로그램을 제공해 주지 않고 있다. 따라서 본 연구에서는 Java 언어를 이용하여 ALE 인터페이스 계층을 구현하였다. <그림 10>은 Java 언어를 이용해서 ECSpec 인스턴스를 생성하는 부분을 보여주고 있다.

```
ECSpec spec1 = new ECSpec();
spec1.readers[0] = "GateA";
spec1.boundaries.startTrigger.setTriggerURI("True");
spec1.boundaries.duration.duration = 5000;
spec1.boundaries.duration.unit.equals(ECTimeUnit.MS);
spec1.reportSpecs[0].reportName = "report1";
spec1.reportSpecs[0].reportSet.equals(ECReportSetSpec.CURRENT);
spec1.reportSpecs[0].output.setIncludeEPC(true);
spec1.includeSpecInReports = false;
```

<그림 10> ECSpec 인스턴스 생성 코드

위의 인스턴스에서는 RFID 리더기의 논리명칭을 *GateA*, 태그 읽기 지속시간을 5초로 설정하고 있다. 또한 태그를 읽어 들인 후 그 정보를 report1로 지정하고 있다. 이와 같은 방법으로 ECSpec 인스턴스를 생성한 후 ALE 인터페이스의 subscribe를 호출하면 <그림 11>과 같은 방식으로 ALE 인터페이스 계층은 미들웨어 최하위 계층인 기본리더관리 계층을 통해 실제로 RFID 리더기를 통제해서 태그를 읽는다.



<그림 11> ALE Layer와 Basic Reader Management Layer의 연동

<그림 12>에는 ALE 인터페이스를 통해서 태그를 읽어 들인 후 생성된 XML 형태의 report 파일을 보여주고 있다. 실제로 <그림 11>에서는 다섯 개의 태그를 읽어 들였지만 <그림 12>에서 보면 5초 동안에 실제로 중복되는 태그를 제외한 후 현재 세 개의 태그만을 report하고 있음을 확인할 수 있다.

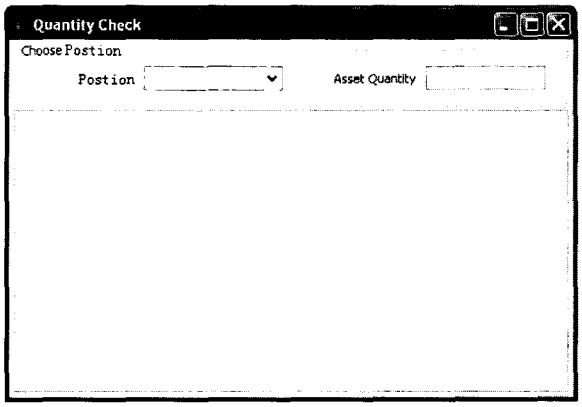
```
<?xml version="1.0" encoding="UTF-8"?>
<ale:ECReports xmlns:ale="urn:epcglobal:ale:xsd:1"
               xmlns:epcglobal="urn:epcglobal:xsd:1"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="urn:epcglobal:ale:xsd:1
               C:\Documents and Settings\Gaolulu\jbproject\RFID-Middleware\
               ale_1_0-schema-20050915\EPCglobal-ale-1_0.xsd"
               date="2008-03-18 15:58:28"
               totalMilliseconds="5000"
               creationDate="2008-03-18 15:58:28"
               terminationCondition="DURATION"
               specName="spec1"
               ALEID="BIS1"
               schemaVersion="1.0">
  <reports>
    <report reportName="report1">
      <group>
        <groupList>
          <member><tag>08E56407000104E0</tag></member>
          <member><tag>11F36407000104E0</tag></member>
          <member><tag>12F46407000104E0</tag></member>
        </groupList>
      </group>
    </report>
  </reports>
</ale:ECReports>
```

<그림 12> XML 형태의 report 파일

4.2 기자재 관리시스템 예제

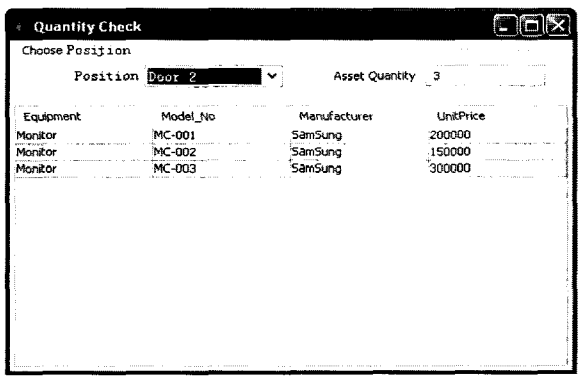
이번 절에서는 앞에서 제시한 미들웨어 아키텍처의 검증을 위해 구현한 기자재관리 시스템을 살펴본다. 본 시스템은 900MHz 모바일 RFID 환경에서 JBuilder 2006, SUN사의 Javacomm 통신패키지, Crimson XML 처리패키지 및 MS사의 Access DB를 사용하여 구현하였다.

<그림 13>은 수량 카테고리에 속하는 위치 인터페이스 메뉴를 선택한 화면이다. 왼쪽의 Position 필드에 위치를 선택하면 해당 위치에 존재하는 모든 기자재 종류와 함께 수량이 집계되어 <그림 14>와 같이 오른쪽에 표시 된다.



<그림 13> 수량 카테고리의 위치 인터페이스 실행 예

즉, 응용프로그램에서 비즈니스 프로세스 인터페이스 계층에 속한 특정위치의 수량체크 로직 명령을 호출하면 이 명령은 ALE 계층을 거쳐 최하단의 기본관리계층으로 전달된다. 기본관리계층에서는 원시태그 ID를 읽어서 상위의 ALE 계층으로 전달하면 여기서 중복데이터 제거 및 필터링 작업을 수행한 후 상위의 비즈니스 프로세스 인터페이스 계층으로 정보를 전달하여 이 계층에서 처리결과를 보여준다.



<그림 14> 지정된 위치의 기자재 정보 표시

5. 결 론

본 연구에서는 EPCglobal ALE 인터페이스 규약을 준수하는 RFID 미들웨어에 대한 아키텍처를 제시하였다. 먼저, 문헌조사 및 연구경험을 바탕으로 RFID 하드웨어 통제를 위한 핵심 기본기능을 파악하였다. 또한 다양한 RFID 응용분야에서 요구되는 비즈니스 로직 기능들을 규명한 후에 이들을 위치(Location), 수량(Quantity), 시간(Time), Direction(방향), 속도(Speed), 안전(Security)과 같이 여섯 개의 카테고리로 분류하여 제시하였다. 둘째, 스펙문서만을 제공하는 EPCglobal ALE 계층을 본 연구에서 제안하는 미들웨어의 RFID 미들웨어 표준스펙 준수를 위해 Java 언어를 사용하여 구현하였다. 셋째, ALE 계층을 RFID 하드웨어와 연결시켜 기본적인 RFID 통제기능을 담당하는 기본 리더기 관리 인터페이스를 구현하였다. 본 미들웨어의 경우 하드웨어 중립적인 XML을 사용하여 프로토콜 명령을 표현하고 Java 플랫폼을 기반으로 구현되었으므로 새로운 하드웨어가 도입되거나 변경되는 경우에도 쉽게 확장 및 기존의 시스템에 통합이 가능하다. 마지막으로 기자재 관리 시스템 개발을 통해 본 연구에서 제시하고 있는 RFID 미들웨어를 검증하고 실현가능성을 보여 주었다.

본 미들웨어 아키텍처는 세 개의 층으로 구성되어 있는데 세 개의 층이 각각 독립적으로 동작하는 것이 가능하여 필요에 따라 유연하게 미들웨어를 실제 응용프로그램 제작 시 활용할 수 있다. 즉, 비즈니스 프로세스 레벨에서는 최상위 계층의 인터페이스를 사용할 수 있고, ALE 인터페이스 규약을 준수해야 하는 응용프로그램의 경우 ALE 인터페이스를 활용하면 된다. 특히, 세밀하게 RFID 하드웨어를 관리해야 하는 경우에는 최하단에 위치한 기본리더관리 인터페이스를 활용하면 응용 프로그램을 작성할 수 있다.

참고문헌

[1] 이충훈, 장경렬, 김재곤, 유우식; "항만 컨테이너 터미널에서의 RFID 적용을 위한 시뮬레이션 연구", 산업경영시스템학회지, 30(4) : 30-38, 2007.

[2] Burnell, John; "What is RFID Middleware and Where Is It Needed?," *RFIDUpdate*, 2006.

[3] EPCglobal; "The Application Level Events (ALE) Specification version 1.0," EPCglobal Inc, Sept 15, 2005.

[4] Finkenzeller, Klaus; *RFID Handbook : Fundamentals and Applications in Contactless Smart Cards and identification*, 2nd Edition, JohnWiley and Sons, 2003.

[5] Fleisch, E. and Dierkes, M.; Ubiquitous computing : Way auto-id is the logical next step in enterprise automation, Technical Report STG-AUTOID-WH-004, Auto-ID center, 2003.

[6] Floerkemeier, C. and Lampe, M.; "RFID middleware design-addressing application requirements and RFID constraints," Proceedings of sOc-EUSAI 2005(Smart Object Conference), Grenoble, 2005.

[7] FrimSYS; RFID Reader Protocol 13.56MHz(Host Communication Description), 2005.

[8] Hoag, J. E. and Thompson, C. W.; "Architecting RFID Middleware," *IEEE Internet Computing*, 10(5) : 2006.

[9] OATSystems; "OAT C4 Architecture," OAT White Paper, 2006.

[10] Oracle; "Enterprise Information Architecture for RFID and Sensor-Based Services," Oracle White Paper, 2006.

[11] Sun; "The Sun Java System RFID Software Architecture," Sun Microsystems Inc, 2005.

[12] Symbol; AR400 API Programmer's Manual, 2004.

[13] Yang Liu; "Integrate Your Enterprise Application with IBM WebSphere RFID Middleware," IBM White Paper, 2006.