

ALE 미들웨어를 위한 다양한 RFID 리더 처리 방법

Processing of Various RFID Reader Devices for ALE Middleware

노영식*, 변영철*, 이동철**
제주대학교 컴퓨터공학과*, 제주대학교 경영정보학과**

Young-Sik Noh(solyo@cheju.ac.kr)*, Yung-Cheol Byun(ycb@cheju.ac.kr)*,
Dong-Cheol Lee(dchlee@cheju.ac.kr)**

요약

RFID 기술은 동시에 여러 개의 사물을 자동으로 인식할 수 있어 유비쿼터스 응용 서비스를 제공하는데 활용될 수 있으며, 이 경우 응용 서비스와 RFID 리더 장치 사이의 교량 역할을 하는 RFID 미들웨어가 필요하다. 본 논문은 기존 EPCglobal의 ALE 미들웨어에서 다양한 RFID 리더를 처리하기 위하여 미들웨어 내부에 소스 코드 형태로 RFID 리더의 스트림 정보 등을 하드 코딩하여 처리하고 있어, 새로운 RFID 리더를 처리하기 위하여 다시 RFID 미들웨어의 소스를 수정해야 하는 단점을 해결하고자 다양한 유형의 RFID 리더 장치들을 효과적으로 처리하기 위한 방법을 제안한다. 이를 위하여 다양한 RFID 리더들의 미들웨어 접속 정보, 데이터 프로토콜 정보, 스트림 정보 등을 온톨로지 메타 데이터로 구축하여 데이터베이스에 저장한 후 리더 장치에서 읽은 데이터를 해석하기 위하여 활용한다. 본 방법을 이용하면 새로운 유형의 RFID 리더 장치가 등장하여도 이에 대한 온톨로지만 추가함으로써 미들웨어 변경 없이 리더 장치를 처리할 수 있음은 물론, RFID 리더 온톨로지 재사용을 통하여 미들웨어를 효과적으로 확장할 수 있다.

■ 중심어 : | RFID 미들웨어 | ALE | 온톨로지 | 유비쿼터스 컴퓨팅 |

Abstract

For realizing ubiquitous computing, many research activities are geared towards various areas including embedded computing, RFID, USN, home networking, context-awareness, and etc. By using the ability of RFID technology to recognize a number of objects simultaneously, more convenient ubiquitous application services are effectively provided. In this case, RFID middleware playing a role as a bridge between RFID reader devices and application services is required as well. In this paper, we propose a method of handling a number of types of RFID reader devices in ALE middleware of EPCglobal. For this, the information of connection and data protocol for a reader device is stored in a database as ontology meta-data, and used to interpret the data read by a reader device. By adding ontology data into a database, even though an RFID device newly emerges, ALE middleware can not only handle the device, but also be effectively extended through reusing ontology data, without any changes in the middleware.

■ keyword : | RFID Middleware | ALE | Ontology | Ubiquitous Computing |

* 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.

접수번호 : #090323-003

심사완료일 : 2009년 04월 20일

접수일자 : 2009년 03월 23일

교신저자 : 변영철, e-mail : ycb@cheju.ac.kr

I. 서론

현재 유비쿼터스 컴퓨팅을 실현하기 위한 핵심 기술로서 자동식별, 센서 네트워크, 홈 네트워크 등의 다양한 분야들에 대한 연구가 활발히 진행되고 있다. 특히 네트워크의 발달로 인해 온라인으로 정보교환을 하는 일이 많아짐에 따라 컴퓨터로 정보를 입력하는 자동화된 방법이 필요하게 되었다.

이러한 기술을 자동인식 및 데이터 획득(AIDC; Automatic Identification and Data Capture)기술이라 하며 AIDC의 한 부분인 RFID(Radio Frequency Identification)는 사람의 작업이나 판단을 궁극적으로 배제하고 상품이 갖는 정보를 자동적으로 취득해서 온라인으로 관련 정보를 처리하는 자동처리 시스템 구현의 핵심요소 기술이며 리더의 안테나를 통해 접촉하지 않고 태그의 정보를 판독하거나 인식하는 객체인식 기술 중의 하나이다[1].

이러한 자동식별 분야의 RFID 기술은 무선 주파수를 이용하여 기존의 바코드로 구성된 시스템보다 높은 인식률을 보여주고 있으며, 동시에 여러 개의 물체를 인식할 수 있어 보다 편리한 서비스를 제공할 수 있으며, 데이터의 수집, 가공, 전달, 장치 제어 및 관리 등을 위하여 응용 서비스 개발자가 RFID 기술을 이용한 유비쿼터스 응용 서비스를 쉽게 구축할 수 있도록 교량 역할을 하는 RFID 미들웨어의 중요성에 대한 인식이 확산되고 있다. 또한 RFID 리더의 데이터 융통성을 높이기 위하여 장치 독립적인 미들웨어에 대한 관심이 점차 높아지고 있으며, 이를 지원하기 위한 확장 미들웨어에 대한 연구가 진행되고 있다.

본 논문은 EPCglobal에서 정의한 RFID 표준인 ALE(Application Level Events)[2] 스펙을 기반으로 한 RFID 미들웨어에서 다양한 RFID 리더들의 미들웨어 접속 정보 및 데이터 프로토콜 정보를 온톨로지 메타데이터로 구축하여, 효율적으로 다양한 RFID 리더를 RFID 미들웨어에서 지원할 수 있는 방법을 제안한다. 그럼으로써 RFID 리더 온톨로지의 재사용은 물론 향후 추가 개발되는 RFID 리더의 접속정보 및 프로토콜 데이터에 대한 온톨로지만 추가함으로써 미들웨어로 처

리할 수 있도록 쉽게 확장할 수 있다.

본 논문의 구성은 다음과 같다. 우선 II장에서 RFID 리더와 프로토콜, 메타 데이터 기술을 분석하고, III장에서는 ALE 기반의 RFID 미들웨어에서 온톨로지를 이용하여 다양한 RFID 리더를 처리하는 방법에 대하여 설명한다. IV장에서는 제안하는 방법을 구현하여 실험한 후 결과를 분석하고, 마지막 V장에서는 본 연구의 결론에 대해 설명한다.

II. 관련 연구 및 기술

1. RFID 리더

[표 1]은 현재 EPCglobal의 RFID 표준 스펙을 준수하여 리더기를 개발하고 있는 국내·외의 회사들을 정리한 것이다.

표 1. RFID 리더 장치 개발회사

국외	국내
Alien Technology	인트정보시스템
SAMSys	세연테크놀로지
ThingMagic	LS산전
Intermec	미네트바코리아
Symbol, Matrics	유컴테크놀로지
Phillips	키스컴
EM Micro	삼성테크윈
TI	제일트로닉스
Hitachi	한세텔레콤
Impinj	크레디패스
DAG System	하이트랙스
Psionteklogix	이씨오
JETT	햄팩스
AWID	쓰리에이로직스
SIRIT	ED
ACG	ATID
FEIG	스마트로
PROMAG	제일정보통신

[표 1]의 회사 외에도 많은 회사에서 RFID 태그는 물론 RFID 리더 제품을 개발하고 있으며, 보다 상세적으로 13.56MHz, 860MHz~960MHz, 2.45GHz 등 주파수

대역의 데이터 처리, EPCglobal에서 제안하고 있는 Gen1 및 Gen2 태그 처리, 수동형과 능동형 태그 처리에 대한 연구 개발을 하고 있다.

이러한 다양한 RFID 리더를 RFID 미들웨어에서 처리하기 위해서 다양한 통신 인터페이스 기술을 이용하여 태그정보를 수집하고 있으며, 대표적인 예로 COM Port를 이용한 RS-232C, RS-485 등의 Serial 통신과 LAN Port를 이용한 TCP/IP Socket 통신 등이 있다. 또한 각각의 RFID 리더마다 인식한 RFID 태그 데이터를 RFID 미들웨어로 보내는 방법이 다르며, 일반적으로 Serial 통신과 Socket 통신을 이용하여 XML 혹은 TEXT 스트림 데이터를 RFID 미들웨어로 자동 및 수동으로 전송한다.

자동형식의 RFID 리더는 전용 터미널을 이용하여 리더의 펌웨어로 접근하여 Serial 및 Socket 통신에 필요한 설정 값을 입력하여 RFID 미들웨어로 태그 데이터를 자동으로 전송할 수 있도록 하며, 수동형식의 RFID 리더는 RFID 미들웨어에서 해당 프로토콜 명령을 사용하거나 API 명령을 이용하여 태그 데이터를 전송 받는다. 따라서 RFID 미들웨어 측면에서 다수의 통신 방식 및 프로토콜 정보, 태그 데이터 스트림 정보를 효율적으로 관리하여 다양한 RFID 리더를 처리할 수 있는 방법이 필요하다.

2. RFID 리더 프로토콜

리더 프로토콜의 표준화는 ISO와 EPCglobal에서 진행하고 있다. [표 2]는 두 기관이 제안한 리더 프로토콜을 비교한 것이다. EPCglobal은 2006년 6월에 EPCglobal Reader Protocol Standard v1.1(이후 RP로 표시) 발표하였다. EPCglobal RP는 EPC Class 규격에 준하는 태그를 읽고, 쓰기위한 Web 기반의 표준 인터페이스를 미들웨어 제공한다[3].

ISO에서는 리더와 응용간의 인터페이스로 15961/15962를 제안하였다[4][5]. 15961/15962는 ISO에서 정의한 RFID 태그 프로토콜 ISO 18000 Family 태그를 수집 및 관리하기 위한 기능을 제공한다. 특히, ISO 18000-7은 항만 물류의 컨테이너를 위한 능동형 RFID 태그 프로토콜 표준으로서 Savi상의 RFID 특허를 근간

으로 하고 있다.

표 2. ISO와 EPCglobal 리더 프로토콜 표준 비교

구분	EPCglobal	ISO
리더 프로토콜	Reader Protocol Standard v1.1	15961/15962
메시지 형식	XML, HTTP, SOAP	ASN.1
태그 프로토콜	EPC Class	18000 Family
동작 방식	수동형	수동형/능동형

그러나 이러한 표준화가 아직 산업 현장에 적용되고 있지는 않으며 현재, 대부분의 RFID 리더는 독자적인 리더 프로토콜을 사용한다. RFID 미들웨어는 이기종 RFID 환경에서 발생하는 태그 정보를 수집하기 위해 이기종 리더의 독자적인 프로토콜을 처리 할 수 있는 기술이 필요하다[6]. 또한 현재 존재하는 RFID 미들웨어의 경우 다양한 형태의 RFID 리더를 지원함에 있어 ISO와 EPCglobal에서 제안한 표준을 바탕으로 한 장비들이 주를 이루고 있다. 그러나 현존하는 다양한 RFID 리더 중 일부만을 지원하는 경우가 대부분이며, 지원하는 RFID 리더를 처리함에 있어 미들웨어에 종속적인 형태로 관리하고 있어서 새롭게 정의되는 다양한 RFID 리더를 처리하도록 확장함에 있어서 효율적이지 않다.

3. 온톨로지 기반 메타 데이터

온톨로지는 시맨틱 웹의 핵심 기술로 지식을 표현하고 추론하기 위하여 사용되며, 표현 언어로는 RDF/RDFS, DAML+OIL, OWL 등이 있다. W3C의 온톨로지 언어인 OWL은 온톨로지의 생성과 공유를 위한 시맨틱 마크업 언어로서 체계적인 온톨로지 구축을 지원할 수 있는 언어이다[7][8].

이러한 언어로 구축된 온톨로지 기반의 정보 검색 기술은 자원을 빠르게 찾아 사용할 수 있다는 점과 자원을 찾는 정확도를 향상시킬 수 있다는 점, 또한 온톨로지에 정의된 개념과 개념간의 관계를 이용하여 사용자의 질의의 의미를 분석하고 동일한 의미를 갖는 키워드에 대한 검색이 가능하다는 장점들을 갖고 있다[9][10].

본 논문에서는 이러한 온톨로지의 장점을 바탕으로 현존하는 다양한 RFID 리더들의 미들웨어 접속 정보

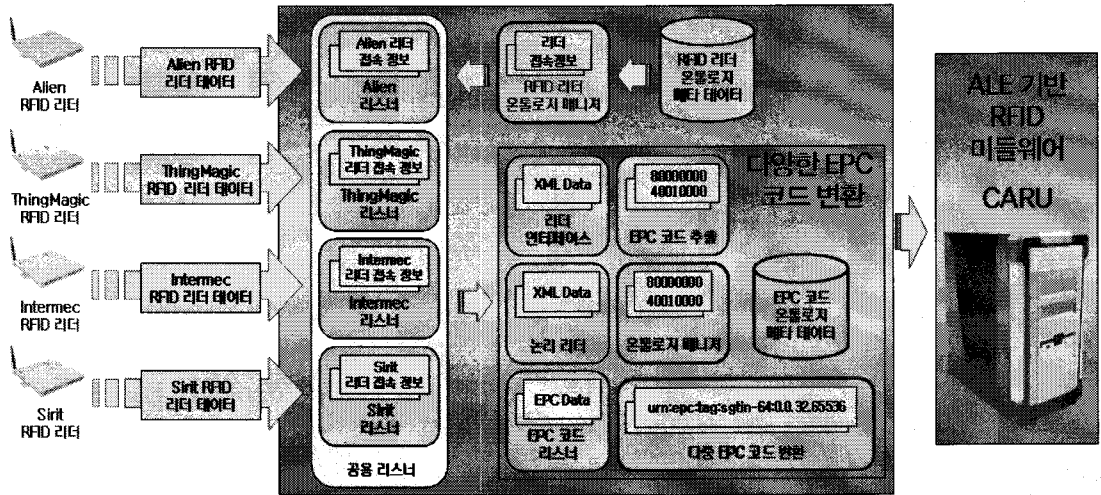


그림 1. 제안 방법의 개요

및 데이터 프로토콜 정보 각각을 온톨로지로 구축하여 체계적인 RFID 리더 관리 및 새로운 RFID 리더에 대해 RFID 미들웨어 측면에서 유연하게 처리 될 수 있도록 관련 인터페이스를 설계 및 구현 하고자 한다.

4. 다양한 RFID 리더 처리 방법

RFID 시스템에는 다수의 RFID 리더에서 생성된 수많은 데이터를 효율적으로 관리하기 위한 미들웨어가 필수이며, 이러한 RFID 미들웨어는 유비쿼터스 환경에 존재하는 다양한 이기종 리더를 처리 할 수 있어야 한다. 기존 다양한 이기종 RFID 리더를 처리하는 연구로 다양한 RFID 리더를 물리적 리더로 설정하고 이를 효율적으로 활용하기 위해 논리적 리더로 매칭하여 추상

화를 기반으로 이기종 리더의 공통 리더 인터페이스를 정의하여 확장성을 지원하는 방법[11]과 EPCglobal의 Reader Management Protocol을 지원하는 다양한 RFID 리더들에 대해서 표준 프로토콜에 따라 리더를 통합 관리하고, 이를 처리하기 위해 RFID 리더 프로토콜을 관리하는 RPFManager를 구현하여 S/W 계층에서 다양한 RFID 리더를 지원할 수 있는 방법[12]들이 있다. 기존 연구들에서는 미들웨어 내부에 소스 코드 형태로 RFID 리더의 스트림 정보 등을 하드 코딩하여 처리하고 있어, 새로운 RFID 리더를 처리하기 위하여 다시 RFID 미들웨어의 소스를 수정해야 하는 문제점이 있다.

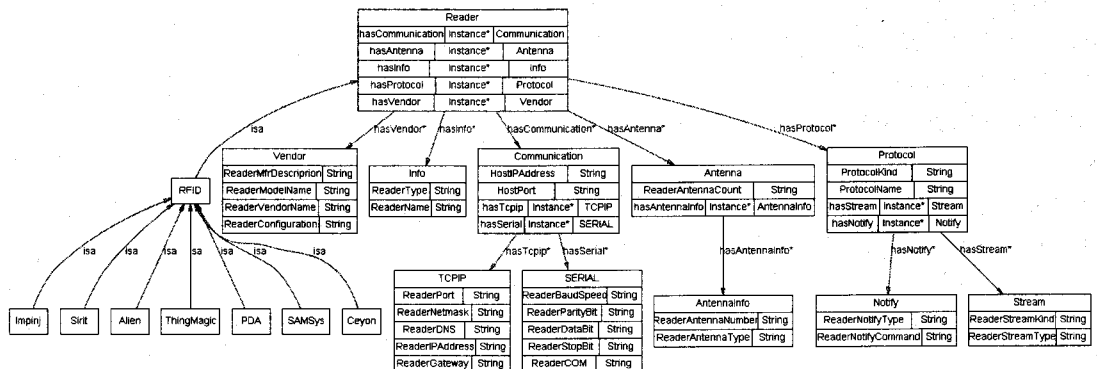


그림 2. 다양한 RFID 리더 처리를 위한 온톨로지 구조도

III. 제안하는 방법

1. 개요

[그림 1]은 본 논문에서 제안하는 방법의 개요이다. ALE 기반의 RFID 미들웨어에서 Alien, ThingMagic, Intermec, Sirit 등의 RFID 리더 장치로부터 다양한 유형의 RFID EPC 코드 데이터를 입력받아야 할 경우 Alien 리더 장치의 미들웨어 접속 정보, ThingMagic 리더 장치의 미들웨어 접속 정보, Intermec 리더 장치의 미들웨어 접속 정보, Sirit 리더 장치의 미들웨어 접속 정보를 바탕으로 각각의 수집 리스너를 생성하며, 이러한 리더 장치 접속 정보는 RFID 리더 온톨로지 메타 데이터의 내용을 RFID 리더 온톨로지 메니저를 이용하여 처리한다. 또한 각각의 RFID 리더 장치로부터 수집되는 태그 데이터 스트림 정보 및 프로토콜 정보 역시 RFID 리더 온톨로지 메타 데이터를 이용하여 다중 EPC 코드 변환 시스템으로 RFID 태그 정보를 XML 형식 혹은 TEXT 스트림 형태로 제공하여, 다중 EPC 데이터 변환을 위한 EPC 코드 온톨로지 메타 데이터를 이용하여 URN 코드 데이터로 변환하고 이를 ALE 표준 스펙을 기반으로 한 RFID 미들웨어에게 제공한다.

2. RFID 리더 온톨로지 구조 설계

[그림 2]는 다중 RFID 리더를 처리하기 위한 온톨로지 구조를 설계한 모습이다. Reader 클래스를 중심으로 RFID 리더를 처리하기 위한 RFID 클래스가 Reader 클래스의 부 클래스로서 구성되며 기타 다양한 RFID 리더를 지원하기 위한 관련 속성들을 정의하였다. [그림 2]에서 설계한 온톨로지 구조의 상세적인 클래스 및 속성 설명은 [표 3]과 같다. 대표적으로 Info 클래스에는 리더 이름 등의 정보, Vendor 클래스에는 리더 제품의 제조 정보, TCPIP 클래스와 SERIAL 클래스는 리더의 Local 통신 셋팅 정보, Communication 클래스는 리더가 미들웨어에 접속하는 네트워크 정보를 구축하게 된다.

표 3. 다양한 RFID 리더 지원을 위한 온톨로지 속성

클래스	속성정보	
Info	ReaderType	리더 유형
	ReaderName	리더 이름
Vendor	ReaderVendorName	리더 제조사
	ReaderMfrDescription	리더 제조이력
	ReaderConfiguration	리더 구성
	ReaderModelName	리더 모델이름
TCPIP	ReaderIPAddress	리더 IP주소
	ReaderPort	리더 TCP포트
	ReaderGateway	리더 게이트웨이
	ReaderNetmask	리더 넷마스크
SERIAL	ReaderDNS	리더 DNS
	ReaderCOM	리더 COM 주소
	ReaderBaudSpeed	리더 시리얼 전달속도
	ReaderStopBit	리더 시리얼 스탑 비트
Communication	ReaderDataBit	리더 시리얼 데이터 비트
	ReaderParityBit	리더 시리얼 패리티 비트
	HostIPAddress	수집 호스트 서버 IP주소
	HostPort	수집 호스트 서버 TCP포트
Protocol	ProtocolKind	프로토콜 종류
	ProtocolName	프로토콜 이름
Stream	ReaderStreamKind	리더 스트림 종류
	ReaderStreamType	리더 스트림 유형
Notify	ReaderNotifyType	리더 통지 유형
	ReaderNotifyCommand	리더 통지 명령어
Antenna	ReaderAntennaCount	안테나 개수
AntennaInfo	ReaderAntennaNumber	안테나 번호
	ReaderAntennaType	안테나 유형

3. 온톨로지 기반 다양한 RFID 리더 처리

[그림 3]은 RFID 리더 온톨로지로 구축한 리더 처리 정보를 바탕으로 다양한 RFID 리더를 RFID 미들웨어에서 처리하기 위한 과정을 표현한 그림이다.

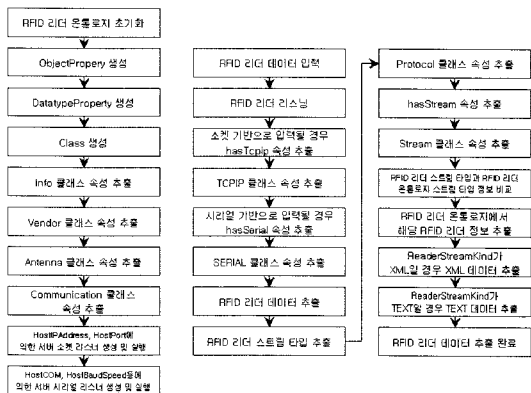


그림 3. 다양한 RFID 리더 처리 흐름도

흐름도의 좌측은 RFID 미들웨어가 초기에 실행했을 시에 처리되는 흐름을 보여주며, 우측은 미들웨어가 실행

행된 후 RFID 리더로부터 데이터가 입력될 시에 처리되는 흐름을 보여준다. RFID 미들웨어에서 다양한 RFID 리더로부터 태그 데이터를 수집하기 위하여 그림 2의 클래스와 속성을 연결하는 관계 정보인 온톨로지 ObjectProperty와 속성 정보에 해당하는 온톨로지 DatatypeProperty를 생성하여 초기화 한다.

다음으로 Info 클래스에서 리더의 이름과 유형 정보를 추출하며, Vendor, Antenna 클래스에서 관련 정보를 추출한다. 그리고 Communication 클래스에서 수집 호스트 서버의 정보를 추출하며, HostIPAddress, HostPort 정보와 HostCOM, HostBaudSpeed 정보 등에 의한 리스너를 생성 및 실행하여 소켓 및 시리얼 통신을 이용하여 다양한 유형의 RFID 리더로부터 태그 데이터를 수집할 수 있도록 한다. RFID 리더로부터 태그 데이터가 미들웨어로 수집되면 TCP/IP 클래스와 SERIAL 클래스의 정보를 바탕으로 소켓 및 시리얼 통신으로 태그 데이터를 수집 받으며, 수집된 태그 데이터는 Protocol, Stream 클래스의 정보를 기반으로 실제 RFID 미들웨어에서 필요한 Hex 타입의 스트림 정보를 추출하여 미들웨어 내부적으로 활용되는 URN 타입으로 변환하여 다양한 응용에 제공한다.

IV. 구현 및 실험

1. 구현 및 실험 환경

RFID 리더 온톨로지 메타 데이터를 구축하여, 효율적으로 다양한 RFID 리더를 RFID 미들웨어에서 지원하기 위한 시스템을 구현한 환경은 [표 4]와 같다.

표 4. 구현 및 실험 환경

구분	내용
온톨로지 구축 및 검색	OWL, Protege 3.3.1 Sun Java SDK 1.5.0_07, Jena API
테스트 데이터	Alien, Ceyon, Impiny, SAMSys, Sirit, ThingMagic사 등 8종, PDA 2종

온톨로지 설계 및 구현을 위하여 OWL, Protege를 이용하였으며, Jena API를 이용하여 온톨로지 정보를 검

색하고 추론하였다. 실험 데이터로는 Alien, Ceyon, Impiny, SAMSys, Sirit, ThingMagic사의 RFID 리더 8종과 HP 등 PDA RFID 리더 2종을 이용하였으며, RFID 리더 온톨로지 정보를 검색, 활용하여 효율적으로 다양한 RFID 리더를 RFID 미들웨어에서 지원하는 지에 대하여 실험하였다.

2. 모듈 구성

[그림 4]는 본 논문에서 제안하는 다양한 RFID 리더를 RFID 미들웨어에서 지원하기 위한 시스템 패키지 다이어그램이다. ReaderApp는 시스템의 메인 클래스이며, listener 패키지는 다양한 유형의 RFID 리더로부터 소켓 리스너와 시리얼 리스너를 바탕으로 태그 데이터를 수집하기 위한 클래스로 구성된다. owlprocess 패키지는 RFID 리더 온톨로지의 속성 자료를 처리하는 PropertyData 클래스와 RFID 리더 온톨로지의 객체 자료를 처리하는 ObjectData 클래스로 구성되며, RFID 리더 온톨로지 정보는 register 패키지의 RegisterReader 클래스에 의해 물리 리더, RegisterLogicalReader 클래스에 의해 논리 리더로 각각 관리된다. 또한 reader 패키지의 각 클래스에 의해 리더 정보, 리더 그룹, 예외 상황 등을 처리되며, datasource 패키지의 클래스에 의해 관리된다. 끝으로 util 패키지는 RFID 리더로부터 태그 데이터를 수집하고, 미들웨어 내부적으로 활용할 수 있도록 부가적으로 필요한 xml, serial 등의 패키지로 구성된다.

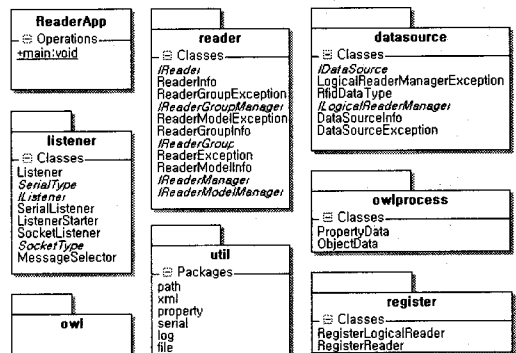


그림 4. 시스템 패키지 다이어그램

3. 코드 온톨로지 구축

[그림 5]는 온톨로지 작성 툴인 Protege를 이용하여 [그림 2]의 RFID 리더 온톨로지 구조도를 바탕으로 클래스와 Object 및 Data 속성을 바탕으로 실제 RFID 마들웨어에서 다양한 유형의 RFID 리더를 처리할 때 필요한 RFID 리더 온톨로지 인스턴스를 작성한 모습이다.

[표 4]에 명시한 EPCglobal의 표준을 따르는 RFID 리더 정보들을 구축하였으며, 현존하는 다른 RFID 리더와 향후 추가 개발되는 RFID 리더들도 인스턴스 생성을 통하여 쉽게 추가할 수 있다.

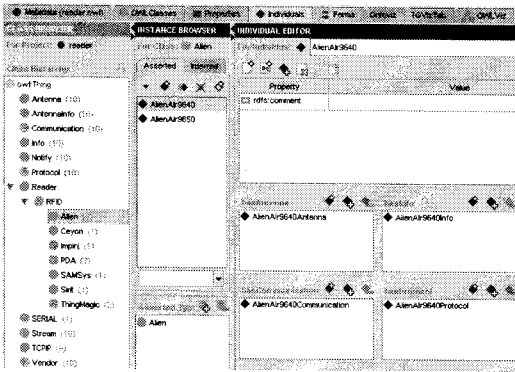


그림 5. RFID 리더 온톨로지 인스턴스 생성 모습

[그림 5]와 같이 Protege는 OWL 기반의 온톨로지를 구축하고자 했을 때 클래스 브라우저와 인스턴스 브라우저를 이용하여 많은 클래스를 트리 구조로 쉽게 관리할 수 있으며, 온톨로지의 실제 구축 데이터인 클래스의 인스턴스 또한 쉽게 구축 및 관리할 수 있다.

[그림 6]은 Protege의 TGVizTab 기능을 이용하여 [그림 5]와 같이 구축된 RFID 리더 온톨로지의 구조를 그래픽으로 보여주는 화면으로 Reader 클래스와 연관되어 있는 Protocol, Communication 등과 같은 타 클래스의 관계와 hasProtocol, hasCommunication 등과 같은 ObjectProperty 정보를 보여주며, [그림 2]의 다양한 RFID 리더 처리를 위한 온톨로지 구조도를 바탕으로 클래스의 관계가 구축되었음을 알 수 있다.

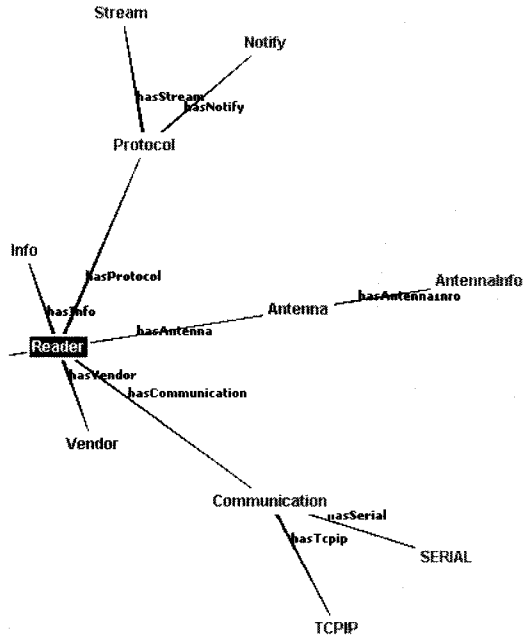


그림 6. Reader 온톨로지 클래스 구조의 일부

[그림 7]은 [그림 6]과 같이 Protege의 TGVizTab 기능을 이용하여 [그림 5]에서 구축한 인스턴스 중 Alien사의 900Mhz 대역을 사용하는 Alr9640 리더 정보에 해당하는 AlienAlr9640 인스턴스 관계도로 클래스와 구축한 인스턴스들의 관계 정보를 보여준다.

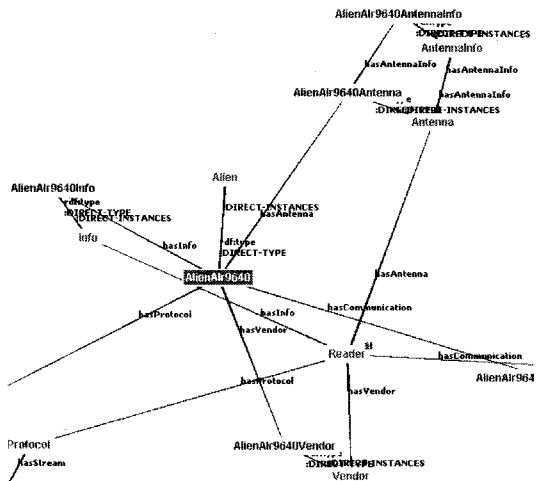


그림 7. AlienAlr9640Class 인스턴스 관계도의 일부

Reader 클래스를 중심으로 RFID 리더의 유형과 이름 정보를 구축한 Info 클래스의 AlienAlr9640Info 인스턴스, RFID 리더 프로토콜의 종류와 이름 정보를 구축한 Protocol 클래스의 AlienAlr9640Protocol 인스턴스, RFID 리더의 제조사 이름과 제품 모델명 등을 구축한 Vendor 클래스의 AlienAlr9640Vendor 인스턴스 등의 관계를 상세적으로 확인 할 수 있다.

이와 같이 [표 1]의 RFID 리더들의 미들웨어 접속 정보 및 데이터 프로토콜 정보를 [표 3]의 RFID 리더 온톨로지 메타 데이터로 구축하여 관리한다.

[그림 8]은 본 논문에서 제안한 다양한 RFID 리더 처리를 위한 온톨로지를 실제로 [그림 2]의 온톨로지 구조를 바탕으로 OWL을 기반으로 온톨로지 작성 툴인 Protege를 이용하여 구축한 온톨로지의 일부를 보여준다.

```

- <owl:disjointWith>
  <owl:Class rdf:ID="Alien" />
- </owl:disjointWith>
- <rdfs:subClassOf>
  <owl:Class rdf:ID="RFID" />
- </rdfs:subClassOf>
</owl:Class>
+ <owl:Class rdf:about="#PDA">
<owl:Class rdf:ID="AntennaInfo" />
+ <owl:Class rdf:about="#ThingMagic">
<owl:Class rdf:ID="Vendor" />
<owl:Class rdf:ID="Protocol" />
<owl:Class rdf:ID="TCPIP" />
<owl:Class rdf:ID="Notify" />
+ <owl:Class rdf:about="#Ceyon">
+ <owl:Class rdf:about="#SAMSys">
+ <owl:Class rdf:about="#Sirit">
+ <owl:Class rdf:about="#RFID">
<owl:Class rdf:ID="SERIAL" />
<owl:Class rdf:ID="Communication" />
<owl:Class rdf:ID="Stream" />
+ <owl:Class rdf:about="#Alien">
<owl:Class rdf:ID="Antenna" />
- <owl:ObjectProperty rdf:ID="hasTcpiip">
  <rdfs:range rdfs:resource="#TCPIP" />
  <rdfs:domain rdfs:resource="#Communication" />
</owl:ObjectProperty>
+ <owl:ObjectProperty rdf:ID="hasStream">
+ <owl:ObjectProperty rdf:ID="hasCommunication">
+ <owl:ObjectProperty rdf:ID="hasAntenna">
+ <owl:ObjectProperty rdf:ID="hasAntennaInfo">
+ <owl:ObjectProperty rdf:ID="hasProtocol">
+ <owl:ObjectProperty rdf:ID="hasVendor">
- <owl:FunctionalProperty rdf:ID="ReaderStreamKind">
  <rdfs:range rdfs:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdfs:type rdfs:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:domain rdfs:resource="#Stream" />
</owl:FunctionalProperty>
+ <owl:FunctionalProperty rdf:ID="ReaderAntennaNumber">
- <owl:FunctionalProperty rdf:ID="ReaderPort">
  <rdfs:range rdfs:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdfs:domain rdfs:resource="#TCPIP" />
  <rdfs:type rdfs:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
</owl:FunctionalProperty>

```

그림 8. 다양한 RFID 리더 처리를 위한 온톨로지 일부

Alien 클래스는 Ceyon, Sirit 클래스 등 다른 RFID 리더 제조회사와 겹치지 않음을 구축하기 위하여 OWL의 disjointWith 정보로 기술하며, 클래스들 간의 관계 정보를 표현하는 ObjectProperty 중 하나인 hasTcpiip 정보를 구축하기 위하여 OWL의 range, domain 정보를

이용하여 기술한다. 또한 실제 온톨로지 데이터를 표현하는 FunctionalProperty 정보를 구축하기 위하여 OWL의 range 정보에 string등의 데이터 타입을 정의하여 기술한다.

4. 데이터 처리 결과

[그림 9]는 RFID 미들웨어 측면에서 Alien 사의 리더 장치 중 ALR-9640 모델을 이용하여 [그림 4]의 모듈들을 바탕으로 테스트한 결과이다.

```

----- Reader Data -----
<Alien-RFID-Reader-Auto-Notification>
<ReaderName>SS_ALR-9640</ReaderName>
<ReaderType>Alien RFID Tag Reader, Model: ALR-9640</ReaderType>
<IPAddress>203.253.213.160</IPAddress>
<CommandPort>4000</CommandPort>
<Time>2008/07/07 17:02:25</Time>
<Reason>TIMED MESSAGE</Reason>
<StartTriggerLines>0</StartTriggerLines>
<StopTriggerLines>0</StopTriggerLines>
<Alien-RFID-Tag-List>
<Alien-RFID-Tag>
  <TagID>880D A0D0 560D 08A1</TagID>
  <DiscoveryTime>2008/07/07 17:02:25</DiscoveryTime>
  <LastSeenTime>2008/07/07 17:02:25</LastSeenTime>
  <Antenna>0</Antenna>
  <ReadCount>12</ReadCount>
</Alien-RFID-Tag>
</Alien-RFID-Tag-List>
</Alien-RFID-Reader-Auto-Notification>
----- Reader Data -----
Reader Ontology - ReaderType : TCPIP
Reader Ontology - ReaderName : SS_ALR-9640
Reader Ontology - ReaderVendorName : Alien
Reader Ontology - ReaderStreamKind : XML
Reader Ontology - ReaderStreamType : <Alien-RFID-Reader-Auto-Notification>
Reader Type : <Alien-RFID-Reader-Auto-Notification>
----- PML Document -----
<?xml version="1.0" encoding="UTF-8"?><pmc:core:Sensor xmlns:pmc:core="urn:autoid:specification:interchange:PMLCore:xml:schema:1" xmlns:pluid="urn:autoid:specification:universal:Identifier:xml:schema:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:autoid:specification:interchange:PMLCore:xml:schema:1 .. /SchemaFiles/Interchange/PMLCore.xsd">
<pluid:ID>203.253.213.160</pluid:ID><pmc:core:Observation>
<pluid:ID>0</pluid:ID><pluid:ID><pmc:core:DateTime>2008/07/07 17:02:25</pmc:core:DateTime>
<pmc:core:Tag><pluid:ID>urn:epit:tag:sgt:in-64:1.5.43.2977</pluid:ID>
</pmc:core:Tag></pmc:core:Observation></pmc:core:Sensor>
----- PML Document -----

```

그림 9. Alien ALR-9640 리더 테스트의 예

RFID 리더 온톨로지에 구축되어 있는 ReaderType, ReaderName, ReaderStreamKind, ReaderStreamType 등의 속성 정보를 바탕으로 통신 프로토콜 및 스트림 정보를 분석하여 태그 데이터를 추출하고, [그림 1]의 다양한 EPC 코드 변환 모듈에 의해 URN 코드로 변환하여 ALE 기반의 RFID 미들웨어에서 활용되는 PML 문서로 변환된 결과를 보여준다. Alien, Ceyon, Impiny, SAMSys, Sirit, ThingMagic사 등의 8종의 RFID 리더 장치를 이용하여 실험한 결과 리더 장치의 유형에 관계없이 ALE 미들웨어를 수정하지 않더라도 리더 장치가 읽은 태그 데이터를 효과적으로 처리할 수 있음을 확인

할 수 있었다.

V. 결론

본 논문은 기존 EPCglobal에서 정의한 사실상의 RFID 미들웨어 표준인 ALE 스펙을 기반으로 한 미들웨어에서 다양한 RFID 리더를 처리하기 위하여 미들웨어 내부에 소스 코드 형태로 RFID 리더의 스트림 정보 등을 하드 코딩하여 처리하고 있어, 새로운 RFID 리더를 처리하기 위하여 다시 RFID 미들웨어의 소스를 수정해야 하는 단점을 해결하고자 다양한 RFID 리더들의 미들웨어 접속 정보, 데이터 프로토콜 정보, 스트림 정보 등을 온톨로지 메타 데이터로 구축하여, 효율적으로 다양한 RFID 리더를 RFID 미들웨어에서 지원할 수 있는 방법을 제안하였다.

RFID 리더 장치가 RFID 미들웨어에 접속하는 정보와 태그 데이터를 전송하는 프로토콜에 대한 정보를 온톨로지 메타 데이터로 구축하여 데이터베이스에 저장한 후 RFID 리더 장치에서 입력되는 Hex 타입의 태그 데이터를 해석하여 ALE 기반의 RFID 미들웨어 내부에서 활용되는 URN 타입으로 변환한다.

제안하는 방법은 RFID 리더 온톨로지의 재사용은 물론 향후에 새롭게 개발되는 RFID 리더의 접속정보 및 프로토콜 데이터에 대한 온톨로지만 추가함으로써 미들웨어를 추가 수정하지 않아도 리더 장치를 처리할 수 있음은 물론, RFID 리더 온톨로지 재사용을 통하여 미들웨어를 효과적으로 확장할 수 있다. 향후 미리 구축된 리더뿐만 아니라 구축되지 않은 리더를 온톨로지 추론과 퍼지 이론을 이용하여 처리할 수 있는 연구를 수행하고자 한다.

참고 문헌

[1] 유승화, "RFID기술현황 및 활용분야", 한국정보과학회지, 제23권, 제7호, pp.64-70, 2005.
 [2] EPCglobal, "The Application Level Event(ALE) Specification Version 1.0," EPCglobal Inc, 2005.

[3] EPCglobal, "Reader Protocol Standard Version 1.1," EPCglobal Inc, 2006.
 [4] JTCl/SC 31/WG 4, "Internet Standard 15961", ISO/IEC, 2004.
 [5] JTCl/SC 31/WG 4, "Internet Standard 15962", ISO/IEC, 2004.
 [6] 장수완, 김종덕, "차세대 항만물류를 위한 장치 독립형 RFID 미들웨어 구성요소의 설계 및 구현", 한국해양정보통신학회논문지, 제11권, 제1호, pp.124-130, 2007.
 [7] <http://www.w3.org/TR/owl-features/>
 [8] <http://www.w3.org/TR/owl-guide>
 [9] 박재홍, 임유정, 김도완, 박찬규, 조현규, "Semantic Web 환경에서의 자원발견", 한국정보처리학회 추계학술대회논문집, pp.713-716, 2002.
 [10] 하상범, 박영택, "온톨로지를 통한 추론형 시맨틱 검색 시스템에 관한 연구", 한국정보과학회 춘계학술대회논문집, 제31권, 제1호, pp.625-627, 2004.
 [11] 안준환, 안성우, 홍봉희, "다양한 RFID 리더를 지원하는 확장성 미들웨어 구조의 연구", 한국공간정보시스템학회 추계학술대회논문집, pp.97-103, 2006.
 [12] 안종민, 송하주(Hajoo Song), "이기종 리더의 통합접속 및 관리를 위한 RPManager의 설계와 구현", 한국정보과학회 2008 가을 학술발표논문집, 제35권, 제2호, pp.92-97, 2008.

저자 소개

노영식(Young-Sik Noh)

정희원



- 2004년 : 탐라대학교 컴퓨터공학과(공학사)
- 2007년 : 제주대학교 컴퓨터공학과(공학석사)
- 2009년 : 제주대학교 컴퓨터공학과(박사수료)

<관심분야> : 지식기반시스템, 시맨틱 웹, 지능형 컴퓨팅, 유비쿼터스 미들웨어

변 영 철(Yung-Cheol Byun)

정회원



- 1993년 : 제주대학교 정보공학과 (공학사)
 - 1995년 : 연세대학교 컴퓨터과학과(공학석사)
 - 2001년 : 연세대학교 컴퓨터공학과(공학박사)
 - 2001년 : 한국전자통신연구원 선임 연구원
 - 2002년 ~ 현재 : 제주대학교 컴퓨터공학과 교수
- <관심분야> : 패턴인식, 시맨틱 웹, 지능형 컴퓨팅, 유비쿼터스 미들웨어

이 동 철(Dong-Cheol Lee)

정회원



- 2002년 : 성균관대학교 산업공학과(공학박사)
- 2003년 ~ 현재 : 제주대학교 경영정보학과 조교수

<관심분야> : Agent, EC, MIS 응용