
온톨로지 기반 KKR 코드 변환

강민수* · 노영식** · 변영철*** · 이동철**** · 전계석*****

KKR code conversion based on ontology

Min- Soo Kang* · Young-Sik Noh** · Yung-Cheol Byun*** · Dong-Cheol Lee**** · Kye-Suk Jun*****

본 연구는 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 광역경제권 선도 산업
인재양성사업의 산학공동과제 연구결과임

요 약

국내 RFID 산업 활성화 및 RFID 서비스 간 상호 운용성 제고를 위하여 한국인터넷진흥원(현KISA)에서는 ISO/IEC 15459 국제 표준을 준수하는 KKR 코드 체계를 발표하였다. 본 논문에서는 EPCglobal의 ALE 표준 스펙을 기반으로 한 RFID 미들웨어에서 RFID 리더로부터 수집되는 다양한 유형의 RFID KKR 코드 데이터를 효율적으로 URN 코드로 변환하기 위하여 온톨로지 기반 코드 변환 방법을 제안한다. 실험 결과 다양한 유형의 RFID KKR 코드 데이터가 URN 코드로 성공적으로 변환됨을 알 수 있었으며, 향후에 새로운 형식을 갖는 KKR 코드가 제안되더라도 새로운 코드에 대한 온톨로지만을 추가함으로써 기존 RFID 미들웨어의 수정 없이 효과적으로 확장할 수 있다.

ABSTRACT

KISA announced KKR code system observing the international standard, ISO/IEC 15459, in order to activate the domestic RFID industry and improve mutual operation among RFID services. In this paper, we propose the way on code conversion based on ontology methods so as to effectively convert various types of RFID KKR code data collected from RFID reader into URN code in the middleware based on the ALE standard spec of EPC global. The results of experiment show that various types of RFID KKR code data could be converted into URN codes successfully. This means that even though new types of KKR code are added, it can be extended easily by adding ontology information without rebuilding the middleware.

키워드

RFID, 미들웨어, 온톨로지, KKR코드, URN코드

Key word

RFID, Middleware, Ontology, KKR, URN

* 정회원 : 제주관광대학 관광멀티미디어게임과(mskang@ctc.ac.kr)
** 정회원 : 제주대학교 컴퓨터공학과
*** 종신회원 : 제주대학교 컴퓨터공학과(교신저자)
**** 정회원 : 제주대학교 경영정보학과
***** 정회원 : 경희대학교 전자공학과

접수일자 : 2010. 12. 03
심사완료일자 : 2010. 12. 13

I. 서 론

RFID는 태그와 리더의 무선 통신을 이용하여 제품, 사람, 동물 등을 인식하고 추적하며 이 과정 중에 얻은 실시간 혹은 축적된 정보를 처리하여 과학적 혹은 상업적인 목적을 이룰 수 있도록 하는 기술이다. 상업적인 목적의 RFID 시스템은 상품에 대한 고유한 정보를 담고 있는 태그에 내장된 자료를 리더가 RF 채널을 통해 비접촉 방식으로 읽어오거나 태그에 자료를 저장하는 연산을 수행하고, 연산의 결과를 네트워크를 통해 전용 서버의 미들웨어가 필터링 등의 처리를 한 후, 다양한 상업적인 응용 프로그램에서 이를 저장하여 활용할 수 있는 구조로 되어있다[1].

RFID 관련 표준을 제정하고 있는 단체인 EPCGlobal에서는 RFID 기술을 위한 EPCGlobal 아키텍처 프레임워크를 제안하였고[2], 리더로부터 수집되는 상품 정보 코드(EPC)를 수집하고 정제(Filtering) 과정을 거쳐 상위 레이어로 전달하기 위한 미들웨어의 역할을 담당하는 구성요소인 Application Level Events(ALE) Specification 1.1을 포함한다[3].

한편, 국내에서 EPC 코드를 사용하기 위해서는 EPCGlobal 단체에 가입하여 EPC 코드를 할당받아야 할 뿐만 아니라 추가적으로 많은 비용과 시간이 소요된다[4]. 한편 한국인터넷진흥원(이전 NIDA가 KISA로 통합된 관계로 이하 KISA로 칭함)에서는 국내 RFID 산업 활성화 및 RFID 서비스 간 상호 운용성 제고를 위하여 ISO/IEC 15459 국제 표준을 준수하는 KKR 코드 체계를 제안하였다[5].

본 논문에서는 EPCGlobal의 ALE 표준 스펙을 기반으로 한 RFID 미들웨어에서 RFID 리더로부터 수집된 다양한 유형의 RFID KKR 코드 데이터를 효율적으로 처리하기 위해 온톨로지 기반의 URN코드 변환 방법을 제안한다.

본 논문의 구성은 다음과 같다. 우선 II장에서 ISO/IEC 15459 KKR, RFID 미들웨어 기술을 분석하고, III장에서는 RFID 미들웨어가 온톨로지를 이용하여 RFID KKR 코드 데이터를 효율적으로 변환 처리하는 방법에 대하여 설명한다. IV장에서는 제안하는 방법을 구현하여 실험한 후 결과를 분석하고, 마지막 V장에서는 본 연구의 결론에 대해 설명한다.

II. 관련 연구 및 기술

2.1. ISO/IEC 15459 KKR

표 1은 ISO/IEC 15459 KKR 코드 기본 체계의 구성이다. 발급기관코드(IAC)는 ISO/IEC 15459-2에서 'National Public Administration'을 위하여 첫 글자를 'K'로, 이후 두 글자는 ISO 3166에 정의된 국가코드 두 문자(대한민국: KR)가 오도록 정의하여 총 3문자(대한민국: 'KKR')를 사용하도록 한다. 하위영역은 IAC를 발급받은 기관인 Issuing Agency에 의해서 정의된다.

표 1. ISO/IEC 15459 KKR 코드 기본 체계
Table 1. The basic system of ISO/IEC 15459 KKR codes

구분	발급자코드(IAC)	하위영역
문자수	3	가변
기본구조	KKR	Issuing Agency가 정의

표 2는 정의된 ISO/IEC 15459 KKR 국가코드 체계이다. 기관코드는 kCode 체계를 이용하여 기관간의 유일한 식별을 제공하며 KKR 코드체계의 주요 적용 대상인 공공분야를 수용할 수 있는 세문자를 할당하였다. 사실상 기관 코드의 '000 ~ 9ZZ' 영역을 향후 확장을 위하여 예약하여 실질적으로 무한대로 사용 가능하도록 하였다. 단, 기관코드 'ZZ0' ~ 'ZZZ'는 자산관리 등 내부 영역을 위한 코드로 사용된다.

표 2. kCode 체계
Table 2. The system of kCode

구분	발급기관코드(IAC)	기관코드(CC: Company Code)	구분자(Prefix)	객체종류 식별코드(IC:Item Code)	객체단위 식별코드(SC:Serial Code)
문자수	3	3	1	가변	가변
세부 설명	KKR	000 ~ 9ZZ	RFU		
		A00 ~ ZZZ	표 3. 참조	기관별 자체정의	기관별 자체정의

구분자(Prefix)는 한 문자로 객체종류 식별코드의 문자수를 정의하며 세부 내용은 표 3과 같다. 객체종류 식

별코드는 가변 길이를 가지며 기관별로 관리하는 개체 간의 유일한 식별을 제공하며 구분자를 통하여 기관별 특수 상황에 맞게 길이의 정의가 가능하다.

표 3. 구분자에 따른 IC 문자수
Table 3. Character number of IC by prefix

구분자 (Prefix)	0	~ 9	A	B	C	D	E	F	G	중략	Z
IC의 문자수	RFU		1	2	3	4	5	6	7	중략	26

객체단위 식별코드도 가변 길이를 가지면서 IC간의 유일한 식별을 제공하며 인코딩시 코드 전체 길이를 나타내는 Object Length를 통하여 기관별로 자유롭게 SC의 길이를 정의 할 수 있다. NIDA는 국내 ISO/IEC 15459 KKR 코드에 대한 발급 권한을 가지고 있는 지식경제부 기술표준원으로 부터 관리권한을 위임받아 2006년 하반기부터 코드 발급을 하고 있다[5].

2.2. RFID 미들웨어

RFID 미들웨어는 리더 등의 장비와 ERP 등의 외부 응용시스템 중간에 위치하여, 리더를 관리하고 수집된 정보 중에서 응용프로그램이 요청한 의미 있는 태그 정보를 전달해 주는 역할을 수행한다.

미들웨어는 리더로부터 전달받은 태그 자료 중에서 중복 되거나 불필요한 자료를 필터링 하거나 그룹화 작업을 수행한다. 필터링이란 중복된 자료를 제거하는 작업 이외에 응용프로그램이 지정한 특정 태그 자료만 선별하여 전달하거나, 특정 위치의 태그 자료만 전달하는 기능을 포함한다. 또한 그룹화란 리더의 태그의 개수만 전달하거나 특정 제조회사의 제품에 부착된 태그의 수 혹은 태그 자료에 대한 그룹화 결과를 전달하는 기능을 의미한다[6].

미들웨어에서 논리적 리더란 관련된 물리적 리더(physical reader) 혹은 기존에 정의된 논리적 리더의 집합을 의미하며, 리더는 지정된 사이클 기간에 지정된 작업을 수행한다. 리더 사이클(Reader Cycle)이란 미들웨어와 리더 사이에서 자료를 주고받는 등의 상호작용을 하는 가장 작은 단위시간을 의미한다. 그 시간동안 리더는 자신이 읽은 하나 이상의 태그 자료를 미들웨어에게 전달할 수 있다.

반면 이벤트 사이클(Event Cycle)이란 미들웨어에게 태그 자료를 요청한 응용프로그램을 대신하여 미들웨어가 하나 혹은 그 이상의 리더와 상호작용하는 논리적인 단위 시간을 의미한다. 응용프로그램은 미들웨어에게 하나의 이벤트 사이클 동안에 어떤 일들을 수행해야 하는지 알려주어야 한다. 예를 들어 응용프로그램은 매 5초의 이벤트 사이클 동안 리더1과 리더2로부터 읽은 자료 중에서 A 회사 제품이 몇 개 입고되는지를 알리는 등의 작업을 지시할 수 있다.

ALE v1.1에서는 태그에 읽기와 쓰기가 가능하기 때문에, 미들웨어가 읽기와 관련된 API를 수행하는 경우 이를 이벤트 사이클 이라 하고 쓰기와 관련된 API를 수행할 경우 커맨드 사이클(Command Cycle) 이라 한다 [1, 7].

2.3. ALE 미들웨어의 URN 형식

ALE 기반 RFID 미들웨어는 RFID 리더에서 들어오는 코드 데이터를 공통의 URN 형식으로 변환처리 하여 처리 한다[9]. 따라서 ALE 기반 RFID 미들웨어에서 KKR 코드 처리를 위해서는 KKR 코드를 효율적으로 URN 형식으로 변환할 수 있어야 한다[10]. 한편 KISA에서 제안하고 있는 KKR 코드의 URN 형식은 다음과 같다[11].

urn:ods:iso-iec:15459:1:KKR,AAA,C,ROM,ABCD

그러나 KISA에서 제안하는 KKR 코드는 ODS(Object Discovery Service)를 위한 URN 으로서, ‘KKR,AAA,C,ROM,ABCD’와 같이 알파벳 대문자로 코드 데이터를 표현하므로 기존의 URN과 형식이 다르기 때문에 ALE 미들웨어는 KISA에서 제시하고 있는 URN 코드를 효과적으로 처리하기에 어려운 점이 있다[8, 11, 12].

이를 해결하기 위하여 NIDA 에서 제안하는 URN 형식을 따르면서 ALE 미들웨어 내에서도 처리할 수 있는 URN 코드가 제안되었다[10].

urn:ods:15459:1:11634.27457.2.130.1083492

URN 코드 부분 중 헤드(head) 부분인 urn:ods:15459:1는 ISO/IEC 15459-1 코드 체계임을 나타내며, URN 코드 중 몸체(body) 부분에 해당하는 11634.27457.2.130.

1083492는 KKR,AAA,C,ROM,ABCD를 십진수로 변환한 것이다. 알파벳 대문자를 십진수로 변환한 이유는 ALE 미들웨어에서 물리적인 코드 데이터를 URN 으로 변경할 때 URN의 몸체 부분을 십진수로 처리하기 때문이다[2, 3, 9].

2.4. KKR 코드 변환을 위한 고려 사항

위와 같이 관련 연구 및 기술의 내용과 문제점들을 바탕으로 본 논문에서 고려해야 내용을 살펴보면 다음과 같다. 기존 온톨로지를 기반으로한 RFID 코드 변환 연구는 EPC 코드에만 한정적이며[13], KKR 코드는 새롭게 제안되어 추가되는 코드체계로서 아직은 그에 대한 연구가 미미한 상황이다[10].

따라서 본 논문에서는 다음과 같은 사항을 고려하였다. 우선 KISA의 KKR 코드 데이터 변환 규칙을 바탕으로 표준을 준수해야 한다. 두 번째로, 유연하면서 다양한 유형의 KKR 코드를 변환 처리할 수 있는 공통 방법이 필요하다. 즉, 현재까지 제안되어 있는 KKR 코드 체계를 모두 처리할 수 있어야 한다. 세 번째로, 확장할 수 있는 KKR 코드 데이터 변환 방법이어야 한다.

즉, 향후 RFID 기반의 유비쿼터스 컴퓨팅 환경은 계속적으로 발전할 것이며, 이에 따라 새로운 형태의 RFID KKR 코드 체계가 제안될 수 있다. 따라서 새롭게 추가되는 데이터 형식을 효율적으로 처리할 수 있도록 쉽게 확장할 수 있어야 한다.

III. 제안하는 방법

3.1. KKR 코드 변환

ISO/IEC 18000-6C 지원 태그에 KKR 코드를 인코딩하고, RFID 리더 장치를 이용하여 태그를 읽어 ALE 기반 RFID 미들웨어에 보낸다[10]. 표 4는 통상적으로 가장 많이 쓰이는 ISO/IEC 18000-6C 지원 태그의 메모리 Bank 01의 세부 구조이다[12].

ISO/IEC 15459 KKR 코드 체계를 사용할 경우 PC의 Toggle 비트가 '1'로 설정되고 AFI 코드가 11로 할당된다. 할당된 이후에 RFID 리더 장치는 KKR 코드 데이터를 읽어서 미들웨어로 보낸다.

표 4. ISO/IEC 18000-6C 지원 태그의 메모리 Bank 01 구조
Table 4. Memory Bank 01 structure of ISO/IEC 18000-6C tag

구성요소		Bank01내의 주소	기능	
CRC-16		00 _h ~0F _h	태그와 리더 사이의 이동정보 에러 검사	
PC	Length	10 _h ~14 _h	'(PC + UII data)의 Word 길이' - 1	
	RFU	15 _h ~16 _h	향후 사용을 위해 예약	
	Toggle(T)	17 _h	'1'(Non-EPC 정보가 UII data 영역에 기록됨)	
	AFI	18 _h ~1F _h	태그 응용 분야 식별	
UII D A T A	DSFID	20 _h ~27 _h	데이터 저장 구조 및 형식 표기	
	Precursor	28 _h ~2F _h	ObjectID & Object의 형식 정의	
	Object Id	Length	30 _h ~	ObjectID의 byte 길이-1
		ObjectID		ObjectID 값
	Object	Length		Object의 byte 길이
		Object		RFID 코드

UII(Unique Item Identifier) DATA는 RFID 코드 데이터를 변환할 때 필요한 부분이다. 통상적으로 ISO 태그 지원 RFID 리더 장치는 미들웨어에게 UII DATA을 보낸다. 대다수 RFID 리더 장치가 미들웨어로 보내는 코드 데이터는 일반적으로 16진수 형태이다. 예를 들어 16진수의 코드 데이터 값이 '05 34 09 5A E5 AD 04 44 10 42 21 90'일 경우 코드 데이터 값을 2진수로 일괄 변환한다.

ISO/IEC 15459 코드 체계의 DSFID(Data Structure Formatted Identifier) 값은 2진수로 '0000 0101'이다. 리더 장치에서 들어오는 코드 데이터 중 앞의 8비트가 '05'이므로, 이 코드 데이터가 ISO/IEC 15459 코드 체계임을 알 수 있다.

다음으로 Precursor 값을 통해서 KKR 코드 체계인지 아닌지를 확인할 수 있다. 5비트 압축을 사용하는 KKR 코드 체계는 ISO 15459 코드 체계 중 15459-1 코드 체계와 15459-4 코드 체계로 나뉜다. 그러므로 KKR 코드 체계의 Precursor 값은 '0011 0001' 이나 '0011 0100'으로 기록되어야 한다. 리더 장치에서 보낸 데이터 중 Precursor에 해당되는 부분이 16진수로 '34'이고 이것을 2진수로

변환하면 '0011 0100'이므로 이 코드 데이터는 KKR 코드 체계를 따르고 있다는 것을 확인할 수 있다.

Object 요소의 Length는 Object 바이트 수를 2진수로 기술하는데, 예시로 든 코드 데이터의 값이 16진수로 '09'이어서 2진수로 변환하면 '0000 1001' 이므로 Object의 길이가 9 바이트로 총 72비트이다.

KKR 코드는 5비트 압축된 아스키 문자로 코드 데이터를 표현하므로 14개의 아스키 문자가 있다. 여기에서 남은 2비트는 아스키 문자가 아니라 단지 '0'을 붙인 것이다. KKR 코드 체계에서는 아스키 문자를 5비트 압축을 해서 사용하기 때문에 아스키 문자를 십진수로 변환하기 위하여 아스키 문자에 해당되는 Object 값을 각각 5비트씩 잘라서 '01'을 붙인다.

이와 같은 방식으로 2진수에서 KKR 코드 데이터의 바디 부분을 아스키 문자로 변환한다. 변환된 아스키 문자에서 7번째 자리인 Prefix 값을 얻어서, IC 자리수가 몇 자리인 지를 알아낸다. 마지막으로 IC 값을 제외한 나머지는 SC 이다. 앞서 언급하였듯이 아스키 문자는 ALE 미들웨어에서 처리하기가 어렵기 때문에 KKR 코드 체계의 IAC, CC, Prefix, IC, SC 값을 십진수로 변환하여 URN에 적용한다[10].

3.2. KKR 코드 온톨로지 구조 설계

그림 1은 KKR 코드 변환을 위한 온톨로지 구조를 설계한 모습을 보여준다. KKR 코드를 범용 URN 코드 형태로 변환하기 위하여 Body, Fragment, Partition 클래스 및 코드 변환에 필요한 속성들을 설계하였다.

온톨로지 구조를 살펴보면 Code 클래스는 urn:ods:15459:1:11634.27457.2.130.1083492에서 urn:ods:15459:1와 같은 URN 코드의 앞부분에 해당하는 코드 URN 정보, 코드 이름, ISO/IEC 15459와 같은 코드 멤버 정보, 코드의 크기 정보 속성을 관리하는 Additional 클래스를 갖고 있으며, 이를 관리하기 위하여 hasAdditional 객체 속성을 갖는다.

또한 RFID 리더로부터 수집된 0534095AE5AD044410422190과 같은 원시 16진 정보의 헤더 정보를 바탕으로 구축된 KKR 코드 온톨로지 정보에서 해당 코드 정보를 추출할 때 활용되는 KKR 코드 헤더 정보와 KKR 코드 인코딩 정보 속성을 관리하는 Class 클래스 갖고 있으며, 이를 hasClass 객체 속성으로 관리한다. 그리고 URN 코드의 뒷부분에 해당하는 11634.27457.2.130.1083492

와 같은 KKR 코드 구조 정보 속성을 관리하는 Body 클래스와 구체적으로 각 구조 구분 정보를 관리하는 Fragment 클래스와 Prefix 정보 기반 URN 코드로 변환할 때 활용되는 파티션 정보를 관리하는 Partition 클래스 등으로 구성된다.

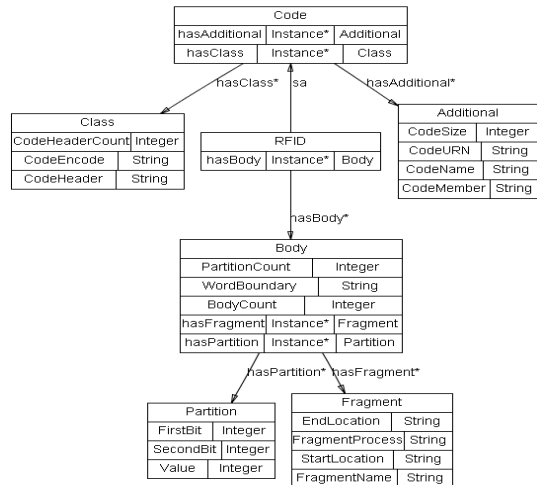


그림 1. KKR 코드 변환을 위한 온톨로지 구조도
Fig 1. Ontology structures for KKR code conversion

3.3. 온톨로지 기반 KKR 코드 변환

그림 2는 KKR 코드 변환을 위한 흐름도이다. 리더로부터 태그 데이터를 입력받아 원시 바이너리 데이터로 변환하여 코드 헤드 값에 해당되는 이진수를 추출한다.

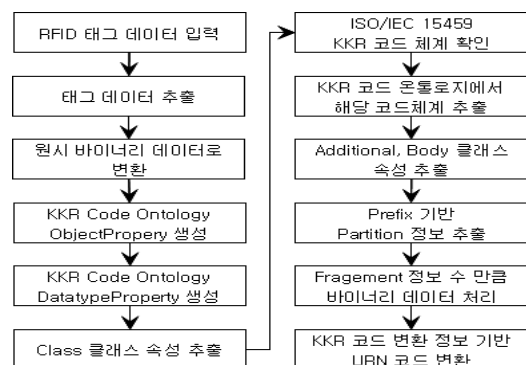


그림 2. KKR 코드 변환 흐름도
Fig 2. The sequence diagram for KKR code conversion

그림 1의 클래스와 속성을 연결하는 관계 정보인 ObjectProperty와 속성 정보에 해당하는 Datatype Property를 생성한다. 이후 추출된 헤더 값과 KKR 코드 온톨로지 데이터를 비교하여 ISO/IEC 15459 KKR 코드 체계를 확인하여 각 코드 체계에 따라 코드 변환을 수행한다.

IV. 구현 및 실험

4.1. 구현 및 실험 환경

온톨로지 설계 및 구현을 위하여 OWL, Protege를 이용하였으며, Jena API를 이용하여 KKR 코드 온톨로지 정보를 검색하였다. 테스트 데이터로 KKR 코드 3종을 이용하여 실험하였고, KKR 코드 온톨로지 정보를 검색, 활용하여 RFID 리더로부터 수집된 KKR 코드 태그 데이터의 헤더 정보를 분석하여 이에 맞는 URN 코드로 변환하는지에 대하여 실험하였다.

4.2. KKR 코드 온톨로지 구축

그림 3은 온톨로지 작성 툴인 Protege를 이용하여 그림 1의 KKR 코드 온톨로지 구조도를 바탕으로 클래스와 Object 및 Data 속성을 바탕으로 실제 ALE 기반 RFID 미들웨어에서 다양한 유형의 KKR 코드 변환을 처리할 때 필요한 Additional, Class, Body 온톨로지 인스턴스를 생성한 모습이다.

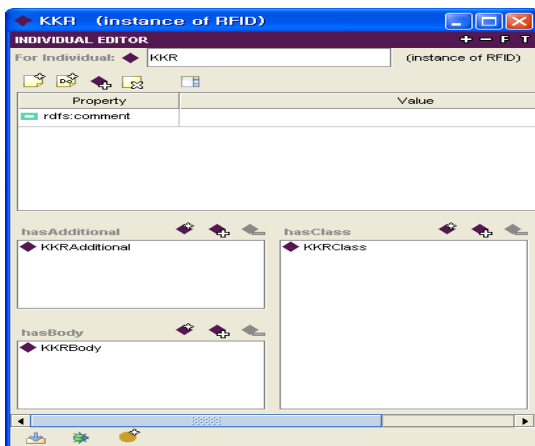


그림 3. KKR 코드 온톨로지 인스턴스
Fig 3. The ontology instance of KKR code

그림 4는 그림 3에서 생성한 인스턴스 중 Additional 인스턴스의 CodeMember Data 속성으로 ISO/IEC 15459, CodeName Data 속성으로 KKR, CodeSize Data 속성으로 96, CodeURN Data 속성으로 urn:ods:15459:1:을 실제 구축한 모습이다. 또한 실제 구축된 OWL은 다음과 같다.

```
<Additional rdf:ID="KKRAdditional">
  <CodeMember rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ISO/IEC 15459</CodeMember>
  <CodeSize rdf:datatype="http://www.w3.org/2001/XMLSchema#int">96</CodeSize>
  <CodeURN rdf:datatype="http://www.w3.org/2001/XMLSchema#string">urn:ods:15459:1:</CodeURN>
  <CodeName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">KKR</CodeName>
</Additional>
```

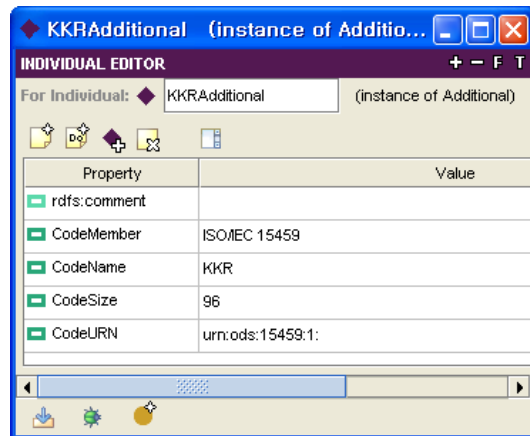


그림 4. Additional 온톨로지 인스턴스
Fig 4. The ontology instance of Additional

그림 5는 KKR 코드 변환을 위해 구축한 KKR 인스턴스의 전체 관계도이다. 즉, Protege를 이용하여 KKR 인스턴스 관계도 및 코드를 처리하기 위한 데이터와 연관되어 있는 타 클래스의 관계 정보 ObjectProperty 정보를 보여주며, 그림 1의 온톨로지 구조도를 바탕으로 구축하였다.

그림 6은 Protege의 TGVizTab 기능을 이용하여 그림 3에서 구축한 KKR 인스턴스 관계도로 클래스와 구축한 인스턴스들의 관계 정보를 보여준다. RFID 클래스의

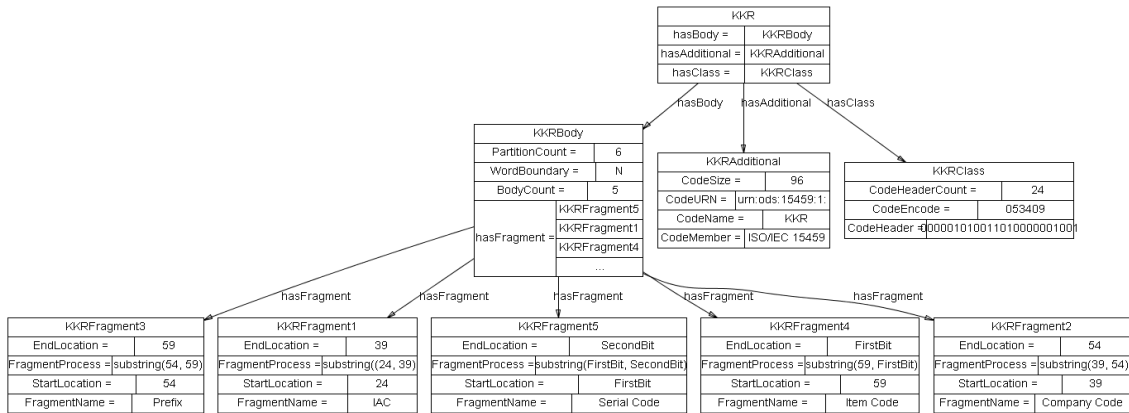
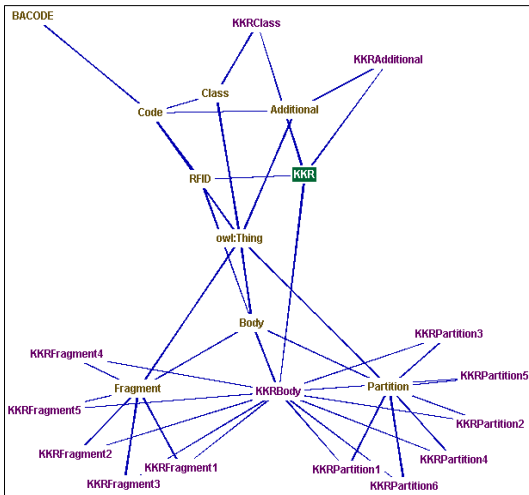


그림 5. KKR 인스턴스 관계도
Fig 5. A relation diagram of KKR instance

KKR 인스턴스를 중심으로 Additional 클래스의 KKR Additional 인스턴스, Class 클래스의 KKRCClass 인스턴스, Body 클래스와 연관된 Partition 클래스의 KKR Partition 인스턴스, Fragment 클래스의 KKRFragment 인스턴스를 바탕으로 KKR 코드 구조 정보를 관리한다.

0534095AE5AD044410422190이라는 데이터가 입력된 경우 이 데이터의 헤더 정보를 기반으로 구축된 온톨로지 정보와 비교하여 URN으로 어떻게 변환할 것인지 추출한다. 그리고 그 값에 의하여 이진 데이터 변경된 태 그 데이터를 이용하여 urn:ods:15459:1:11634.27457.2.130.1083492 이라는 URN 코드로 변환한다.



```

2010-10-23 14:17:04,109 INFO - === The code conversion start which uses Ontology ===
2010-10-23 14:17:04,265 INFO - BIN Header = 000001010011010000001001
2010-10-23 14:17:04,265 INFO - URN Header = urn:ods:15459:1:
2010-10-23 14:17:04,281 INFO - Body Count = 5
2010-10-23 14:17:04,281 INFO - Partition Count = 6
2010-10-23 14:17:04,281 INFO - WordBoundary = N
2010-10-23 14:17:04,281 INFO - PartitionValue = 2
2010-10-23 14:17:04,281 INFO - FirstBit = 10
2010-10-23 14:17:04,281 INFO - SecondBit = 25
2010-10-23 14:17:04,281 INFO - FragmentName = I&C
2010-10-23 14:17:04,281 INFO - StartLocation = 24
2010-10-23 14:17:04,281 INFO - EndLocation = 39
2010-10-23 14:17:04,281 INFO - FragmentName = Company Code
2010-10-23 14:17:04,281 INFO - StartLocation = 39
2010-10-23 14:17:04,281 INFO - EndLocation = 54
2010-10-23 14:17:04,281 INFO - FragmentName = Prefix
2010-10-23 14:17:04,281 INFO - StartLocation = 54
2010-10-23 14:17:04,281 INFO - EndLocation = 59
2010-10-23 14:17:04,281 INFO - FragmentName = Item Code
2010-10-23 14:17:04,281 INFO - StartLocation = 59
2010-10-23 14:17:04,281 INFO - EndLocation = 69
2010-10-23 14:17:04,281 INFO - FragmentName = Serial Code
2010-10-23 14:17:04,281 INFO - StartLocation = 69
2010-10-23 14:17:04,281 INFO - EndLocation = 94
2010-10-23 14:17:04,281 INFO - RFID TAG DATA = 0534095AE5AD044410422190
2010-10-23 14:17:04,281 INFO - BIN = [ 96 ] 0000010100110100000010010101010110010111
2010-10-23 14:17:04,281 INFO - URN = urn:ods:15459:1:11634.27457.2.130.1083492
2010-10-23 14:17:04,296 INFO - === The code conversion end which uses Ontology ===
    
```

그림 7. KKR 코드 Prefix B 변환 실험 결과
Fig 7. The result of code conversion for KKR code Prefix B

그림 6. KKR 인스턴스 관계도
Fig 6. A relation diagram of KKR instance

4.3. 실험 결과

그림 7은 KKR 코드 중 Prefix 값이 B인 코드를 변환 실험한 결과이다. KKR 코드 형식에 따라 인코딩된

그림 8은 KKR 코드 중 Prefix 값이 D인 코드를 변환 실험한 결과이다. KKR 코드 형식에 따라 인코딩된 0534095AE4318C94A483AD68이라는 데이터가 입력된 경우 이 데이터의 헤더 정보를 기반으로 urn:ods:15459:1:11634.3171.4.676417.27482 이라는 URN 코드로 변환

한다.

V. 결 론

```

2010-10-23 14:18:44,453 INFO - === The code conversion start which uses Ontology ===
2010-10-23 14:18:44,609 INFO - BIN Header = 000001010011010000001001
2010-10-23 14:18:44,609 INFO - URN Heaer = urn:ods:15459:1:
2010-10-23 14:18:44,609 INFO - Body Count = 5
2010-10-23 14:18:44,609 INFO - Partition Count = 6
2010-10-23 14:18:44,609 INFO - WordBoundary = N
2010-10-23 14:18:44,625 INFO - PartitionValue = 4
2010-10-23 14:18:44,625 INFO - FirstBit = 20
2010-10-23 14:18:44,625 INFO - SecondBit = 15
2010-10-23 14:18:44,625 INFO - FragmentName = IAC
2010-10-23 14:18:44,625 INFO - StartLocation = 24
2010-10-23 14:18:44,625 INFO - EndLocation = 39
2010-10-23 14:18:44,625 INFO - FragmentName = Company Code
2010-10-23 14:18:44,625 INFO - StartLocation = 39
2010-10-23 14:18:44,625 INFO - EndLocation = 54
2010-10-23 14:18:44,625 INFO - FragmentName = Prefix
2010-10-23 14:18:44,625 INFO - StartLocation = 54
2010-10-23 14:18:44,625 INFO - EndLocation = 59
2010-10-23 14:18:44,625 INFO - FragmentName = Item Code
2010-10-23 14:18:44,625 INFO - StartLocation = 59
2010-10-23 14:18:44,625 INFO - EndLocation = 79
2010-10-23 14:18:44,625 INFO - FragmentName = Serial Code
2010-10-23 14:18:44,625 INFO - StartLocation = 79
2010-10-23 14:18:44,625 INFO - EndLocation = 94
2010-10-23 14:18:44,625 INFO - RFID TAG DATA = 0534095AE4318C94A483AD68
2010-10-23 14:18:44,625 INFO - BIN = { 96 } 00000101001101000000100101010110010000
2010-10-23 14:18:44,625 INFO - URN = urn:ods:15459:1:11634.3171.4.676417.27482
2010-10-23 14:18:44,640 INFO - === The code conversion end which uses Ontology ===
    
```

그림 8. KKR 코드 Prefix D 변환 실험 결과
 Fig 8. The result of code conversion for KKR code Prefix D

표 5는 본 논문에서 제안한 KKR 변환 방법을 적용하여 Prefix 3종류의 KKR 코드를 URN 코드로 변환한 실험 결과로 기존 연구와 동일한 URN 코드로 변환한 결과이다. 그림 7과 그림 8의 실험 로그 정보에 출력된 바와 같이 코드 당 평균 변환 시간은 0.187초이다.

표 5. KKR 코드 변환 실험 결과
 Table 5. Experimental results of KKR code conversion

실험데이터	Prefix	변환결과	시간
0534095AE5AD044 410422190	B	urn:ods:15459:1:11634.27 457.2.130.1083492	0.187
0534095AE4318C9 4A483AD68	D	urn:ods:15459:1:11634.31 71.4.676417.27482	0.187
0534095AE410882 110863210	A	urn:ods:15459:1:11634.10 58.1.1.69307524	0.187
0534095AE411905 8C0443214	B	urn:ods:15459:1:11634.11 24.2.792.1117317	0.188
0534095AE4110C2 108864298	A	urn:ods:15459:1:11634.10 91.1.1.35754150	0.188

이와 같이 본 연구에서는 KKR 태그 에뮬레이터에 의해 각 세부 코드 타입 및 Prefix 별로 10개의 랜덤 데이터를 생성하여 변환 실험을 수행한 결과 모든 코드에 대하여 성공적으로 URN 코드로 변환함을 알 수 있었다.

본 논문에서는 EPCglobal의 ALE 표준 스펙을 기반으로 한 RFID 미들웨어에서 RFID 리더로부터 수집된 다양한 유형의 RFID KKR 코드 데이터를 효율적으로 처리하기 위한 방법을 제안한다. 현재 존재하는 RFID KKR 데이터뿐만 아니라 새롭게 정의되는 RFID KKR 데이터를 효과적으로 처리하기 위하여 온톨로지 기반 방법을 제안하였다. 제안하는 방법을 구현한 후 세부 코드 타입 및 Prefix 별로 실험한 결과 데이터를 성공적으로 변환할 수 있음을 알 수 있었다. 제안하는 방법은 KKR 코드 온톨로지의 재사용은 물론 향후에 새로운 형식을 갖는 KKR 코드가 제안되더라도 단지 KKR 코드 체계에 대한 온톨로지만 추가함으로써 미들웨어를 추가 수정하지 않아도 KKR 코드 및 그 외 코드를 처리할 수 있도록 쉽게 확장할 수 있다.

참고문헌

- [1] 강승석, 장은실, 박길주, "표준 RFID 시스템의 ALE v1.1 미들웨어 구현에 관한 연구", 한국정보기술학회논문지, 제7권 제4호, pp. 124-133, 2009.8.
- [2] EPCGlobal Inc., "The EPCGlobal Architecture Framework", March 2007.
- [3] EPCGlobal Inc., "The Application Level Events Specification, Version 1.1", February 27, 2008.
- [4] 김경호, 정한영, 이상훈, "국방 RFID 태그 코드 선정 및 ONS 구축방안", 정보과학회지 제25권 제9호, 2007. 9.
- [5] 한국 인터넷 진흥원, "RFID 코드 인코딩 지침서 V1.0", 2006. 10.
- [6] C. Floerkemeier, M. Lampe, RFID middleware design - addressing application requirements and RFID constraints, Gesellschaft fur Informatik (INFORMATIK 2005), pp. 277-281.
- [7] EPCglobal, EPCglobal Standards Overview, www.epcglobalinc.org/standards/
- [8] 홍연미, 변영철, "ALE 기반 RFID 미들웨어 시스템 설계 및 구현", 한국해양정보통신학회 논문지, 제 11권, 제4호, 2007. 4.

- [9] EPCglobal Inc., EPCglobal Tag Data Translation 1.0, 2006.
- [10] 변지웅, 변영철, 이동철, "ALE 미들웨어를 위한 KKR 코드 변환", 한국해양정보통신학회논문지, 제12권, 제10호, pp.1759-1766, 2008. 10.
- [11] 한국 인터넷 진흥원, "RFID 검색 시스템 구축 및 운영 지침서 V1.2", 2006. 12.
- [12] 한국 인터넷 진흥원, "사례제시를 통한 RFID 적용 분사, 지사간 자산출입관리 시스템 구축 가이드", 2006. 12.
- [13] 노영식, 변영철, "온톨로지 기반 EPC 코드 자동 변환 방법", 한국해양정보통신학회 논문지, 제12권, 제3호, pp.452-460, 2008. 10.



이동철(Dong-Cheol Lee)

2003년 2월 : 성균관대학교
산업공학과(공학박사)
2003년 5월~현재 : 제주대학교
경영정보학과 조교수

※관심분야 : Agent, EC, MIS 응용



전계석(Kye-Suk Jun)

1983년 2월 : 연세대학교
전자공학과(공학박사)
1979년 3월~현재 : 경희대학교
전자공학과 교수

※관심분야 : 초음파 및 마이크로파 통신시스템



강민수(Min-Soo Kang)

1997년 2월 : 경희대학교
전자공학과(공학석사)
1999년 2월 : 경희대학교
전자공학과(박사수료)

1999년 3월~현재 : 제주관광대학 관광멀티미디어
게임과 조교수

※관심분야 : 유비쿼터스 미들웨어, 시맨틱 웹, 네트워크
프로그래밍



노영식(Young-Sik Noh)

2007년 : 제주대학교
컴퓨터공학과 석사
2009년 : 제주대학교
컴퓨터공학과 박사수료

※관심분야 : 유비쿼터스 미들웨어



변영철(Yung-Cheol Byun)

1995년 : 연세대학교
컴퓨터과학과 석사
2001년 : 연세대학교
컴퓨터공학과 박사

2002년~현 : 제주대학교 컴퓨터공학과 교수

※관심분야 : 패턴인식, 지능형 컴퓨팅