

An RDB to RDF Mapping System Considering Semantic Relations of RDB Components

Hajung Sung[†] · Jangwon Gim^{**} · Sukhoon Lee^{***} · Doo-Kwon Baik^{****}

ABSTRACT

For the expansion of the Semantic Web, studies in converting the data stored in the relational database into the ontology are actively in process. Such studies mainly use an RDB to RDF mapping model, the model to map relational database components to RDF components. However, pre-proposed mapping models have got different expression modes and these damage the accessibility and reusability of the users. As a consequence, the necessity of the standardized mapping language was raised and the W3C suggested the R2RML as the standard mapping language for the RDB to RDF model. The R2RML has a characteristic that converts only the relational database schema data to RDF. For the same reasons above, the ontology about the relation data between table name and column name of the relational database cannot be added. In this paper, we propose an RDB to RDF mapping system considering semantic relations of RDB components in order to solve the above issue. The proposed system generates the mapping data by adding the RDFS attribute data into the schema data defined by the R2RML in the relational database. This mapping data converts the data stored in the relational database into RDF which includes the RDFS attribute data. In this paper, we implement the proposed system as a Java-based prototype, perform the experiment which converts the data stored in the relational database into RDF for the comparison evaluation purpose and compare the results against D2RQ, RDBToOnto and Morph. The proposed system expresses semantic relations which has richer converted ontology than any other studies and shows the best performance in data conversion time.

Keywords : RDB, RDF, Semantic Web, Ontology, R2RML

관계형 데이터베이스 구성 요소의 의미 관계를 고려한 RDB to RDF 매핑 시스템

성 하 정[†] · 김 장 원^{**} · 이 석 훈^{***} · 백 두 권^{****}

요 약

시맨틱 웹의 확산을 위해 관계형 데이터베이스에 저장된 데이터를 온톨로지로 변환하는 연구가 활발히 진행 중이다. 관계형 데이터베이스에 저장된 데이터를 온톨로지로 변환하기 위한 연구들은 관계형 데이터베이스의 구성 요소와 RDF 구성 요소를 매핑하는 방식인 RDB to RDF 매핑 모델을 주로 사용한다. 하지만 지금까지 제안된 매핑 모델들은 그 표현방식이 서로 다르며, 이는 사용자의 접근성과 재사용성을 떨어뜨린다. 이로 인해 표준화된 매핑 언어의 필요성이 대두되었으며, W3C에서는 RDB to RDF 모델의 표준 매핑 언어로서 R2RML을 제안하였다. R2RML은 관계형 데이터베이스 스키마 정보만을 RDF로 변환하는 특징을 가진다. 이와 같은 이유로 관계형 데이터베이스의 테이블 명, 컬럼 명 사이의 관계정보에 대한 온톨로지를 추가할 수 없다. 이 논문에서는 이러한 문제를 해결하기 위해 관계형 데이터베이스 구성 요소의 의미 관계를 고려한 RDB to RDF 매핑 시스템을 제안한다. 제안 시스템은 R2RML에서 정의한 관계형 데이터베이스의 스키마 정보에 RDFS 속성 정보를 확장하여 매핑 정보를 생성한다. 이러한 매핑 정보는 관계형 데이터베이스에 저장된 데이터를 RDFS 속성 정보가 포함된 RDF로 변환시킨다. 이 논문에서는 제안 시스템을 자바 기반의 프로토타입으로 구현하며, 비교 평가를 위해 관계형 데이터베이스에 저장된 데이터를 RDF로 변환하는 실험을 수행하고 결과를 D2RQ, RDBToOnto, Morph와 비교한다. 제안 시스템은 다른 연구들에 비해 변환한 온톨로지가 풍부한 의미관계를 표현하며, 데이터 변환 시간에서 가장 우수한 성능을 보인다.

키워드 : 관계형 데이터베이스, RDF, 시맨틱 웹, Ontology, R2RML

* 이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(Ns.2012M3C4A7033346).
† 준 회 원 : 고려대학교 컴퓨터·전자통신공학과 석사과정
** 준 회 원 : 한국과학기술정보연구원 선임연구원
*** 준 회 원 : 고려대학교 컴퓨터·전자통신공학과 박사과정
**** 종신회원 : 고려대학교 융합소프트웨어전문대학원 교수
논문접수 : 2013년 9월 27일
수정일 : 1차 2013년 12월 4일
심사완료 : 2013년 12월 11일
* Corresponding Author : Doo-Kwon Baik(baikdk@korea.ac.kr)

1. 서 론

웹 온톨로지 언어를 사용하는 시맨틱 웹의 영향으로 온톨로지의 사용이 증가하고 있다. 또한 이를 저장하고 관리하기 위한 온톨로지 저장소에 대한 연구가 진행되고 있다[1-7]. 온톨로지 저장소는 RDF 형태의 그래프 모델을 표현하기 위한

데이터 구조를 가진다. 이러한 데이터 구조는 대부분 관계형 데이터베이스에 저장된 데이터 구조와 다르다. 이로 인해 시맨틱 웹에서는 관계형 데이터베이스 기반의 웹 페이지에 저장된 데이터를 검색할 수 없는 문제를 가진다. 하지만 위와 같은 문제로 인해 시맨틱 웹에서 기존 웹 페이지에 저장된 데이터를 사용하지 않는 것은 시맨틱 웹의 확산을 막는 다른 문제를 발생시킨다. 따라서 시맨틱 웹에서도 기존 웹 페이지에 저장된 데이터를 사용하는 방법에 대한 연구의 필요성이 증가하였다.

이러한 연구들은 관계형 데이터베이스에 저장된 데이터를 온톨로지 형태로 변환하는 방법에 대한 것이 대부분이다. 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지 형태로 변환하기 위해서는 주로 매핑 방식을 이용한다[8,9]. 또한 매핑 규칙을 표현하기 위해 다양한 매핑 언어들도 제안되고 있다. DB2OWL은 매핑을 구현하기 위하여 R2O 언어를 개발하였으며[10], D2RQ의 경우에는 자체적으로 정의한 언어를 사용한다[11]. 하지만 다양한 매핑 언어의 사용은 매핑 언어에 대한 접근성과 재사용성을 떨어뜨린다.

위와 같은 문제를 해결하기 위해 W3C에서는 표준화된 매핑 언어인 R2RML을 제안 하였다[12]. Morph와 DB2Triple는 R2RML에서 정의한 매핑 규칙을 사용하여 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지 형태로 변환한다[13,14]. 하지만 R2RML은 관계형 데이터베이스의 스키마 정보만을 RDF 형태의 온톨로지 형태로 변환한다. 따라서 외부에 존재하는 테이블 및 컬럼과 내부의 테이블 및 컬럼 사이의 관계 정보를 표현하지 않는다. 또한 외래 키(Foreign Key)로 정의되지 않은 테이블 사이의 관계 정보에 대해서도 정의하지 못하여 온톨로지를 확장하는데 제약이 따른다. 이러한 문제점은 테이블 명과 컬럼 명 사이의 관계 정보를 온톨로지 형태로 표현하여 해결한다. 따라서 R2RML에 관계형 데이터베이스의 테이블 명, 컬럼 명 사이에 관계 정보를 표현하기 위한 매핑 규칙 정의와 확장이 필요하다.

이러한 문제점을 개선하기 위해 이 논문은 관계형 데이터베이스 구성 요소의 의미 관계를 고려한 RDB to RDF 매핑 시스템을 제안한다. 제안 시스템은 표준화된 매핑 언어인 R2RML을 사용하며 RDFS 속성 정보를 매핑 모델에 추가한다. 추가된 RDFS 속성 정보는 관계형 데이터베이스에 정의된 관계 정보를 표현한다. 제안 시스템은 최종적으로 RDF의 클래스 및 속성으로 변환된 온톨로지를 표현한다. 또한 변환된 온톨로지는 온톨로지 저장소에 저장되며 SPARQL 질의문을 통하여 온톨로지를 검색할 수 있다.

이 논문의 구성은 다음과 같다. 제2장에서는 관계형 데이터베이스를 RDF 형태의 온톨로지 형태로 변환하기 위한 RDB to RDF 변환에 관련된 연구 내용을 기술하며, 제3장에서는 제안 시스템의 구조를 설명한다. 제4장에서는 제안 시스템의 구현 환경 및 방법에 관하여 기술하며, 제5장에서는 제안 시스템 검증에 위한 실험 결과 및 기존 연구와의 비교 평가를 보인다. 마지막으로 제6장에서는 결론에 관해 기술한다.

2. 관련 연구

이 장에서는 관계형 데이터베이스를 RDF 형태의 온톨로지 형태로 변환하는 연구의 필요성과 방법에 대해서 기술한다. 또한 기존 연구에서 제안된 매핑 모델에 대해서 논하며 R2RML을 매핑 언어로 사용하는 연구에 관해서 기술한다.

2.1 RDB2RDF

기존의 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지 형태로 변환하기 위한 방식으로 관계형 데이터베이스의 구성요소와 RDF의 구성요소를 매핑 하는 연구가 이루어졌다. RDB2RDF는 관계형 데이터베이스의 테이블을 RDF의 클래스로 매핑 하며 컬럼 및 속성은 RDF의 속성으로 매핑 한다[8,9]. 이러한 매핑 규칙에 따라 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지 형태로 변환한다. RDB2RDF에서 매핑은 구현 방식에 따라 정적, 동적 구현으로 구분된다. 정적 구현 방식은 매핑 규칙을 사용하여 관계형 데이터베이스에 저장된 데이터를 RDF로 변환한다. 정적 구현 방식의 변환은 구현이 쉽지만, 변화가 많은 데이터에는 적합하지 않다는 문제를 가진다. 동적 구현 방식은 SPARQL 질의문을 SQL로 변환하는 방식을 통해 관계형 데이터베이스에 저장된 데이터를 검색한다[15,16,17]. 이는 관계형 데이터베이스에 저장된 데이터를 실시간으로 얻어올 수 있으므로 데이터의 최신성을 보장한다. 하지만 동적 방식은 질의문의 완전한 변환이 어려우므로 질의 처리 성능이 떨어지는 문제를 가진다.

2.2 D2RQ

D2RQ는 매핑을 이용하여 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지 형태로 변환하는 대표적인 연구이다. D2RQ는 자체적으로 개발한 매핑 언어를 사용한다. D2RQ의 매핑 언어는 D2R에서 얻은 경험을 토대로 생성되었다[18]. 매핑 언어에 따라 관계형 데이터베이스에 접근하여 스키마 정보에 대한 매핑 정보를 자동으로 생성한다. 자동으로 생성된 매핑 정보는 관계형 데이터베이스의 기본적인 정보만 가지고 있다. 따라서 사용자는 기본 키(Primary Key)와 외래 키(Foreign Key) 및 조인 관계에 대한 정보를 추가로 입력한다. 입력된 정보를 사용하여 풍부한 의미를 표현하여 자동 생성의 한계를 극복한다. D2RQ 서버는 생성된 매핑 정보를 참고하여 관계형 데이터베이스의 데이터를 RDF 형태의 온톨로지 형태로 변환한다. 변환된 데이터는 시스템의 메모리에 RDF 형태의 온톨로지 형태로 생성된다. 이렇게 생성된 데이터는 SPARQL을 통해 검색할 수 있다[19]. 또한 D2RQ는 매핑 정보를 참고하여 SPARQL을 SQL로 변환한다[20]. 질의문의 변환에 의한 매핑 모델은 데이터의 최신성을 보장하며 추가적인 트리플 저장소가 필요하지 않은 장점이 있다. 하지만 D2RQ의 질의문 변환은 조인 관계를 매핑 정보에 정의하지 않았을 때는 변환을 지원하지 않는다. D2RQ는 동일한 컬럼 명에 대한 URI를 테이블 별로 다르게

지정해야 한다. 따라서 두 개 이상 존재하는 동일한 컬럼명을 검색하는 SPARQL 질의문의 결과값이 하나의 컬럼명에 대한 결과값을 갖는 문제점이 있다. 마지막으로 D2RQ는 표준화된 매핑 언어를 사용하지 않아 매핑 언어에 대한 재사용성과 접근성이 낮다. 이는 매핑 언어의 학습에 들어가는 비용을 높인다.

2.3 RDBToOnto

RDBToOnto는 관계형 데이터베이스에 저장된 데이터를 온톨로지로 변환하는 연구로서 TAO 프로젝트에서 진행하였다[21]. 데이터 변환 방식은 관계형 데이터베이스의 테이블 및 컬럼에 대한 스키마 정보만을 온톨로지로 변환한다. RDBToOnto는 관계형 데이터베이스에 저장된 데이터의 의미를 표현하기 위해 RTAXON 학습 방법을 이용하여 자동 분류를 지원한다[22]. RTAXON에 정의된 규칙에 따라 카테고리 관련된 어휘를 가지고 있는 단어에 대해 subClassOf 관계를 자동으로 지원한다. 추가적으로, RDBToOnto는 사용자 정의에 의한 제약 조건을 추가, 데이터 중복을 방지를 위한 정규화 기능, 편리한 UI를 제공한다. RDBToOnto는 정적 구현 방식으로 데이터의 최신성을 보장하지 않는다. 그리고 데이터 변환을 위해 추가적인 변환 시간이 필요하므로 대용량의 데이터 처리에는 적합하지 않다. RDBToOnto는 표준화된 매핑 언어를 사용하지 않는다. 또한 관계형 데이터베이스의 테이블 명, 컬럼 명 사이에 관계 정보를 표현하기 위한 온톨로지로 추가하는 데는 제약을 가진다.

2.4 R2RML

W3C의 RDB2RDF 워킹 그룹에서는 매핑 언어의 표준화를 위하여 R2RML을 제안하였다. R2RML은 관계형 데이터베이스의 테이블, 뷰, SQL 질의문으로 이뤄진 데이터에 대한 매핑을 지원한다. R2RML은 트리플사상(LogicalTable)을 최상단 노드로 하며 하위 노드에 논리테이블(LogicalTable), 주어사상(SubjectMap), 술어목적어사상(PredicateObjectMap) 노드를 가진다. 각 테이블 정보는 트리플사상에 매핑되며 논리테이블 노드는 테이블, 뷰, SQL 질의문에 대한 정보를 표현한다. 주어사상은 RDF 변환된 데이터의 주어(Subject)를 표현하며 rr:template을 사용하여 주어의 속성을 나타낸다. 이 때 rr:template 값은 기본 키의 컬럼 명을 포함하여 표현한다. rr:template에 포함된 기본 키의 컬럼 명은 기본 키의 데이터 값으로 변환되어 주어를 구성한다. 술어목적어사상은 술어(Predicate)와 목적어(Object)의 값을 설정한다. 이 때 술어에는 RDF의 술어 정보를 표현하며 목적어사상(ObjectMap)에 RDF의 목적어 정보를 표현한다. 목적어사상은 컬럼 값을 표현하기도 하지만 rr:template을 사용하여 특정 컬럼 값을 표현할 수도 있다. 참조목적어사상(RefObjectMap)은 테이블간의 조인 관계를 표현한다. 즉, 참조목적어사상을 이용하여 조인 관계가 설정된 테이블간에 컬럼 값을 표현한다. 마지막으로 그래프사상(GraphMap)은 주어사상과 술어목적어사상에 포함되어 RDF의 그래프를 표현한다.

2.5 Morph

Morph는 W3C에서 제안한 R2RML을 구현하여 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지로 변환하는 연구이다[13]. Morph는 자바 기반 시스템으로 구성되어 있으며 수동으로 매핑 정보를 생성한다. 매핑 구현 방식은 정적 방식을 사용하며 매핑 규칙은 R2RML의 정의에 따른다. R2RML의 정의에 따라 트리플사상 노드는 관계형 데이터베이스의 테이블과 1:1 대응된다. 트리플사상 노드의 하위 노드인 논리테이블, 주어사상, 술어목적어사상 노드를 사용하여 트리플사상에 정의된 테이블 명과 컬럼 명 정보를 표현한다. 또한 참조목적어사상을 이용하여 테이블 간 조인 관계를 표현한다. Morph는 표준화된 매핑 언어를 사용한다. 따라서 매핑 언어에 대한 접근성과 재사용성이 높다. 하지만 Morph는 R2RML에 정의된 관계형 데이터베이스 스키마 정보만을 RDF 형태의 온톨로지로 변환한다. 따라서 관계형 데이터베이스에 테이블 명과 컬럼 명 사이의 관계 정보를 표현하는데 제약이 따른다.

2.6 DB2Triple

DB2Triple은 W3C에서 제안한 R2RML과 다이렉트 매핑 표준을 이용하여 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지로 변환한다[14]. DB2Triple은 RDB2RDF 워킹 그룹에서 검증을 받았으며 Antidot사에서 자바 기반의 오픈 소스 형태로 제공하고 있다. 매핑 정보는 수동으로 생성하며 R2RML에서 정의한 매핑 규칙을 따른다. 따라서 트리플사상을 이용하여 관계형 데이터베이스의 테이블 정보를 표현한다. 매핑 구현 방식은 정적 방식을 사용하며 이는 실시간 변경 점이 많은 데이터에 대한 사용에 제약을 가진다. DB2Triple은 표준화된 매핑 언어인 R2RML을 사용하여 시스템 독립적인 매핑 언어를 사용하는 장점이 있다. 하지만 R2RML은 관계형 데이터베이스의 스키마 정보만을 RDF 형태의 온톨로지로 변환한다. 따라서 DB2Triple은 관계형 데이터베이스의 테이블 명, 컬럼 명 사이의 관계정보는 변환하지 않는다. 또한 사용하기 편리한 UI를 제공하지 않아 자바 라이브러리 파일을 사용자가 추가해야 하는 문제점을 가진다.

관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지로 변환하는 연구의 필요성에 따라 많은 연구들이 진행되었다. D2RQ와 RDBToOnto는 RDB to RDF 매핑 모델을 사용하는 대표적인 연구이다. 하지만 기존의 연구들은 표준화된 매핑 언어를 사용하지 않아 매핑 언어의 접근성과 재사용성이 떨어진다. 이러한 문제를 해결하기 위해 R2RML이 제안되었으며 R2RML을 적용한 대표적인 연구는 Morph와 DB2Triple이 있다. R2RML을 적용한 연구들은 표준화된 매핑 언어를 사용하였으나 관계형 데이터베이스의 스키마 정보만을 RDF로 변환하는 문제점을 가진다. 따라서 관계형 데이터베이스의 테이블 명, 컬럼 명 사이의 관계정보는 변환하지 않는다. 위와 같은 문제를 해결하기 위해 이 논문은 관계형 데이터베이스 구성 요소의 의미 관계를 고려한 RDB

to RDF 매핑 시스템을 제안한다.

3. 제안 시스템

이 장에서는 관계형 데이터베이스 구성 요소의 의미 관계를 고려한 RDB to RDF 매핑 시스템을 제안한다. 3.1절에서는 제안 시스템을 간략하게 소개하고 3.1.1절에서는 시스템 상세 구조를 설명한다. 3.1.2절에서는 이 논문에서 사용하는 주요 표기에 대하여 정의한다. 3.1.3절에서는 RDB to RDF 매핑, 3.1.4절에서는 R2RML 분석기, 3.1.5절에서는 SQL 생성기, 3.1.6절에서는 RDB to RDF 변환기를 각각 설명한다. 마지막으로 3.2절에서 RDB to RDF 변환 프로세스를 기술하여 제안 시스템에 대한 설명을 마친다.

3.1 제안 시스템 개요

제안 시스템은 R2RML을 이용하여 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지로 변환한다. 또한 제안 시스템은 R2RML에 관계형 데이터베이스의 관계 정보들을 RDFS 속성으로 추가하여 변환된 온톨로지에 추가한다. 이를 위해 RDB to RDF 매핑, R2RML 분석기, SQL 생성기, RDB to RDF 변환기를 가진다. RDB to RDF 매핑은 RDB 구성요소와 RDF 구성요소를 매핑 하는 기능을 가진다. RDB to RDF 매핑에서 정의한 규칙을 바탕으로 R2RML 분석기는 R2RML 문서를 분석한다. R2RML을 분석한 정보는 SQL 생성기에 전달되어 SQL을 생성한다. 위에서 생성된 SQL 질의문을 사용하여 관계형 데이터베이스의 저장된 데이터에 접근하여 데이터를 가져온다. 마지막으로 RDB to RDF 변환기를 통해 SQL 질의문을 통해 가져온 데이터를 RDF 형태의 온톨로지로 변환한다.

1) 제안 시스템 구조

Fig. 1은 제안 시스템 구조를 보여준다. 제안 시스템은

Fig. 1과 같이 “R2RML 분석”, “SQL 생성”, “RDB to RDF 변환”의 세 부분으로 구성된다.

- R2RML 분석: 사용자가 입력한 R2RML 문서를 분석하여 관계형 데이터베이스 스키마 정보를 정의한다. R2RML은 트리플사상을 기반으로 논리테이블, 주어사상, 술어목적어사상을 하위 노드로 가진다. 트리플사상은 하나의 테이블에 매핑되며 논리테이블 정보를 사용하여 테이블 정보를 정의한다. 또한 주어사상을 통해 주어에 정의할 rr:template 정보를 정의한다. 마지막으로 술어목적어사상 정보를 사용하여 술어와 목적어 정보를 정의한다.

- SQL 생성: R2RML 분석 단계에서 정의한 정보를 기반으로 SQL질의문을 생성한다. 트리플사상에는 관계형 데이터베이스의 테이블 정보가 정의되어 있다. 논리테이블 정보를 사용하여 SQL 질의문의 From 절을 정의한다. 다음으로 주어사상에 정의된 기본 키 정보를 활용하여 테이블의 기본 키 정보를 Select 절에 추가한다. 마지막으로 술어목적어사상 정보를 바탕으로 컬럼 정보를 Select 절에 추가하여 SQL 질의문을 완성한다.

- RDB to RDF 변환: RDB에 저장된 데이터를 RDF로 변환하여 저장한다. R2RML을 기반으로 생성된 SQL 질의문을 통해 관계형 데이터베이스에 저장된 데이터를 질의한다. SQL 질의문을 통해 가져온 결과는 질의결과를 관리하는 자료구조에 저장된다. 질의 결과에 저장된 데이터는 R2RML에 저장된 정보에 사용하여 RDF로 변환된다. 이때 주어는 주어사상에 저장된 정보를 사용하여 정의된다. 다음으로 술어, 목적어는 술어목적어사상에 저장된 정보를 사용하여 정의된다. 술어에 저장된 정보에 따라 목적어의 생성은 다음의 방법을 따른다. 첫째 술어에 저장된 정보가 관계형 데이터베이스의 스키마 정보이면 목적어는 관계형 데이터베이스에 저장된 데이터가 된다. 둘째 술어에 저장된 정보가 RDFS 속성이면 목적어는 술어목적어사상에 정의된 정보를 따른다. 위에 방법에 따라 술어목적어사상의 정보는 술어, 목적어로

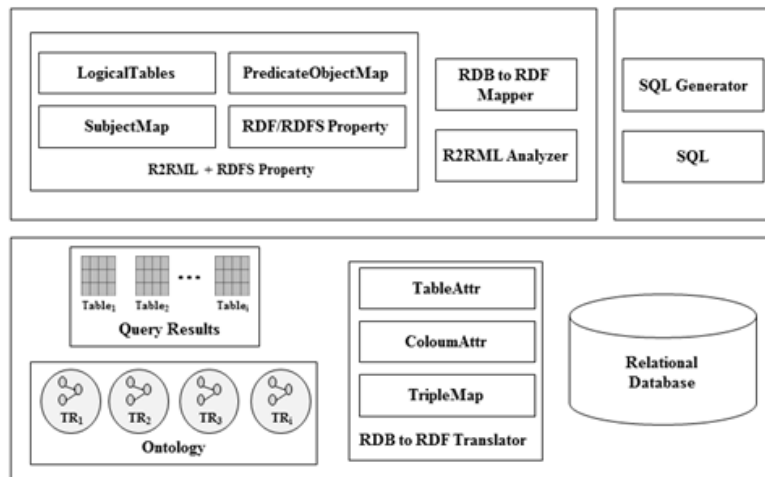


Fig. 1. System architecture

변환된다. SQL 질의문을 통해 생성된 질의 결과에 대한 변환이 완료되면 RDF 문서를 생성하여 변환을 완료한다.

2) 주요 표기 및 기호 정의

이 절에서는 이 논문에서 사용하는 주요 표기 및 기호 등에 대해 정의한다. 이 논문에서 사용하는 주요 표기 및 기호를 Table 1에 정리 한다.

3) RDB to RDF 매핑

제안 시스템에서 제시하는 RDB to RDF 매핑 규칙은 Table 2와 같다. 관계형 데이터베이스의 구성요소를 테이블 명, 뷰 명, 컬럼 명, 기본 키, 외래 키, 인스턴스로 정의한다. 또한 RDFS 구성요소는 총 17개의 속성을 매핑 대상으로 구성한다. RDFS의 속성 중 관계형 데이터베이스의 구성요소와 매핑 가능한 속성은 type, subclassOf, subPropertyOf, objectPropertyOf, domain, range이다. 다른 속성의 경우에는 관계형 데이터베이스의 구성요소와 매핑 시킬 수가 없다.

테이블 명과 뷰 명의 경우에는 type과 매핑하며 subclassOf 속성을 사용하여 테이블 간의 상 하위 관계를 나타낸다. 마지막으로 2개 이상의 테이블을 연결하기 위해 사용되는 테이블은 domain과 range로 표현될 수 있다. 따라서 domain과 range도 테이블 명과 뷰 명으로 매핑된다. 기본 키와 외래 키는 objectProperty를 구성하여 표현한다. 이때 술어는 컬럼 명이 되므로 술어와 컬럼 명을 매핑 시킨다. 또한 주어는 기본 키와 외래 키로 매핑되며 인스턴스는 목적어와 매핑된다. 컬럼 명의 경우에는 subProperty를 사용하여 컬럼 간의 상 하위 관계를 나타낼 수 있기 때문에 subProperty와 매핑된다. subProperty를 표현하기 위해 주어와 목적어에 컬럼 명이 사용된다. 따라서 컬럼 명은 주어와 목적어로 매핑 될 수 있다. 관계형 데이터베이스의 구성요소와 직접적으로 매핑 할 수 없는 11개의 RDFS 구성요소는 사용자의 정의에 따른다. 즉 관계형 데이터베이스의 인스턴스 정보는 사용자 정의에 따라 매핑 하는 규칙을 따른다.

Table 1. Definition of Notation and Symbol

Nataion/Symbol	Description
R2RML _d	R2RML Documnet
RDF _d	RDF Documnet
TA[]	A list of th definition table attributes
CA[]	A list of th definition column attributes
TM[]	A list of th definition TripleMap
RS	SQL Query results
SQL _s	A select statement in SQL
SQL _{from}	A from statement in SQL
SQL[]	A SQL query list
Parsing()	A function for extracting TM information through to analysis a R2RML document
SetTableAttr()	A function for definition of TA[]
AddTableAttr()	A function of adding table attributes to TA[] List
SetColumnAttr()	A function for definition of CA[]
AddColumnAttr()	A function of adding column attributes to CA[] List
SetFromstatement()	A function of defining a from statement in SQL
SetKeyStatement()	A function of defining a primary key
SetSelectStatement()	A function of defining a select statement in SQL
AddSQL()	A function of adding SQL to SQL list
SetTripleTemplete()	A function of confirming template information of a subjectMap
SetSubject()	a function of defining subject
SetPredicate()	a function of defining predicate
SetObject()	a function of defining object
AddRDF)	a function of adding defined RDF to RDF Document

Table 2. RDB to RDF Mapping rule

RDB Component	RDFS components
Table Name	type, subClassOf, domain, range
View Name	type, subClassOf
Column Name	subProperty, objectProperty
PK Name	object Property
FK Name	object Property
Instance	type, sub Class Of, sub Property Of, object Property, domain, range, seeAlso, labe, is DefineBy, comment, first, member, rest, values, subject, predicate, object

4) R2RML 분석기

Table 3은 R2RML을 분석기의 알고리즘을 나타낸다. R2RML 분석기는 입력 값으로 R2RML_d를 갖고 출력 값은 TA[], CA[]를 가진다. R2RML 분석기에 입력된 R2RML_d는 파싱 함수를 이용하여 R2RML 문서 내 트리플사상을 정의한다. 트리플사상은 논리테이블, 주어사상, 술어목적어사상으로 구성되며 R2RML 문서 내 정의를 따라 TM[]에 저장된다. TM[]은 트리플사상으로 정의되며 R2RML 문서에는 한 개 이상의 트리플사상이 정의되어 있다. 따라서 TM[]은 리스트 형태의 자료구조를 가진다. 파싱 함수를 통해 TM[]의 정의가 완료되면 테이블과 컬럼 정보를 정의한다. SetTableAttr 함수는 TM[]에 정의된 논리테이블 정보와 주어사상 정보를 이용하여 테이블의 이름과 rr:template 정보를 정의한다. 이렇게 정의된 테이블 정보는 TA[]에 저장되며 TM[]의 수만큼 정보를 가진다. 테이블 속성 정의가 완료되면 AddTableAttr 함수를 이용하여 테이블 속성 정보를 TA[] 리스트에 추가한다. TA[] 정보가 완료되면 컬럼 속성 정보를 정의한다. 컬럼 속성은 SetColumnAttr 함수를 통해 정의된다 이 때 컬럼 속성은 CA[]에 저장되며 컬럼 명과 컬럼 별칭을 정의하며 추가적으로 RDFS 속성 정보를 가진다. 컬럼 속성 정보의 정의가 완료되면 AddColumnAttr 함수를 이용하여 컬럼 속성 정보를 CA[] 리스트에 추가한다.

Table 3. R2RML Analyzer

[Algorithm] R2RML Analyzer
1: Input: R2RML _d
2: Output: TA[],CA[]
3: Initialize: a=0
4: Parsing(R2RML _d)
5: FOR ∀TM[a] ∈ TM DO
6: TA[a]= SetTableAttr(TM[a])
7: CA[a] = SetColumnAttr(TM[a])
8: INCREMENT a
9: ENDFOR

5) SQL 생성기

Table 4는 SQL 질의문을 생성하는 생성기의 알고리즘을 나타낸다. SQL 생성기는 TA[], CA[]를 입력 값을 갖고

SQL[]을 출력 값으로 가진다. TA[]에는 테이블 정보가 저장되어 있다. 따라서 SQL 질의문은 TA[]에 저장된 데이터를 기반으로 생성된다. TA[]에 저장된 테이블 명 정보는 SetFromStatement 함수를 통해 SQL 질의문의 From 절을 생성하는데 사용된다. 이때 생성된 SQL 질의문은 SQL_{from}에 저장된다. SQL 질의문의 From 절의 생성이 완료되면 테이블의 기본 키 값을 설정한다. 테이블의 기본 키 명은 TA[]에 저장되어 있으며 기본 키 명을 Select 절에 추가하여 기본 키 값을 가져온다. 이렇게 설정된 기본 키 값은 주어를 정의할 때 사용된다. 기본 키 명의 정의가 완료되면 TA[]에 속한 컬럼 명을 Select 절에 추가한다. 컬럼 명은 CA[]에 저장되어 있으며 TA[]에 저장된 테이블 명과 CA[]에 저장된 테이블 명이 일치하는 경우에만 Select 절에 추가한다. CA[]에 저장된 컬럼 명을 Select 절에 추가하여 Select 절의 생성을 완료한다. Select 절의 생성이 완료되면 SQL_s에 저장한다. 하나의 테이블 속성에 대한 SQL 질의문 변환이 완료되면 SQL_s와 SQL_{from}을 SQL[]에 저장한다. 각각의 테이블 속성마다 생성된 SQL 질의문은 SQL[]에 리스트 형태로 저장된다.

Table 4. SQL Generator

[Algorithm] SQL Generator
1: Input: TA[],CA[]
2: Output: SQL[]
3: Initialize: a=0,b=0
4: FOR ∀TA[a] ∈ TA[] DO
5: SQL _{from} = SetFromStatement(TA[a])
6: SQL _s = SetKeyStatement(TA[a])
7: FOR ∀CA[b] ∈ CA[] DO
8: IF CA[b].tableName=TA[a].tableName THEN
9: SQL _s = SQL _s + setSelectStatement(CA[b])
10: ENDIF
11: INCREMENT b
12: END FOR
13: SQL[a] = SQL _s +SQL _{from}
14: INCREMENT a
15: ENDFOR

6) RDB to RDF 변환기

Table 5는 RDB to RDF 변환기의 알고리즘을 나타낸다. RDB to RDF 변환기는 입력 값으로 TA[], CA[], SQL[]을 가지며 출력 값으로 RDF_d를 가진다. 입력된 SQL 질의문은 Query 함수를 통해 관계형 데이터베이스에 저장된 데이터에 질의한다. Query 함수는 모든 SQL 질의문에 대한 질의 결과를 RS에 저장한다. RS에 저장된 질의 결과는 각각의 저장 데이터별로 RDF로 변환된다. RS의 저장된 데이터를 RDF로 변환하는 과정은 TA[]에 저장된 데이터를 기반으로 한다. TA[]에는 테이블에 대한 정보가 저장되어 있다. 따라서 TA[]에 저장된 테이블 정보와 RS에 저장된 테이블 정보를 확인하여 일치하는 데이터를 찾는다. 테이블 정보가 일치하면 TA[]에 저장된 rr:template 정보를 참조하여 트리

플의 주어 정보를 저장한다. 다음으로 CA[]에 저장된 컬럼 정보를 사용하여 술어와 목적어 정보를 저장한다. 이때 술어의 정보가 RDFS 속성으로 설정되어 있으면 목적어 정보를 CA[] 설정된 테이터로 저장한다. 하나의 레코드에 대해 RDF 변환이 완료되면 RDF_d 문서에 추가한다. SQL 질의문 결과 데이터의 변환이 완료된다.

Table 5. RDB to RDF Translator

[Algorithm] RDB to RDF Translator	
1:	Input: TA[],CA[],SQL[],
2:	Output: RDF _d
3:	Initialize: a=0,b=0,c=0
4:	RS[] = Query(SQL[])
5:	FOR \forall RS[a] \in RS[] DO
6:	FOR \forall TA[b] \in TA[] DO
7:	IF RS[a].tableName = TA[b].tableName THEN
8:	SetTripleTemplete(TA[b])
9:	SetSubject(TA[b])
10:	FOR \forall CA[c] \in CA[] DO
11:	SetPredicate(CA[c])
12:	SetObject(CA[c])
13:	RDF _d =addRDF()/addRDFdocument
14:	INCREMENT c
15:	ENDFOR
16:	INCREMENT b
17:	ENDIF
18:	ENDFOR
19:	INCREMENT a
20:	ENDFOR

3.2 RDB to RDF 변환 프로세스

Fig. 2는 제안 시스템의 RDB to RDF 변환 프로세스를 나타낸다. 제안 시스템은 RDB to RDF 매퍼, R2RML 분석기, SQL 생성기, RDB to RDF 변환기를 가진다. 매퍼는 Generator 함수를 이용하여 관계형 데이터베이스 구성요소와 RDFS 구성 요소와의 매핑 규칙을 정의한다. R2RML 분석기는 정의된 매핑 규칙을 사용하여 파싱 함수를 통해 R2RML에 정의된 정보를 설정한다. 이때 SetTableAttr 함수와 SetColumnAttr 함수를 사용한다. 다음으로 AddTableAttr, AddColumnAttr 함수를 사용하여 관련 정보를 저장한다. R2RML에 정의된 정보의 분석이 끝나면 SQL 생성기는 SQL 질의문 생성을 시작한다. SQL 질의문을 Select, From 절로 구분하며 R2RML의 분석 정보를 이용하여 Select, From 절을 정의한다. 또한 R2RML에 정의된 기본 키, 외래 키 정보를 설정한다. 위와 같은 과정을 통해서 SQL 문을 생성하며 관계형 데이터베이스에 저장된 데이터를 질의한다. 마지막으로 RDB to RDF 변환기는 SQL 질의문을 통해 얻은 결과를 바탕으로 RDF를 생성한다. 이 때 테이블 명, 컬럼 명 사이의 관계정보가 R2RML에 정의되어 있으면 정의된 정보를 바탕으로RDF를 추가로 생성한다. R2RML에 정의된 모든 트리플사상의 변환이 완료되면 RDB to RDF 변환 프로세스를 마친다.

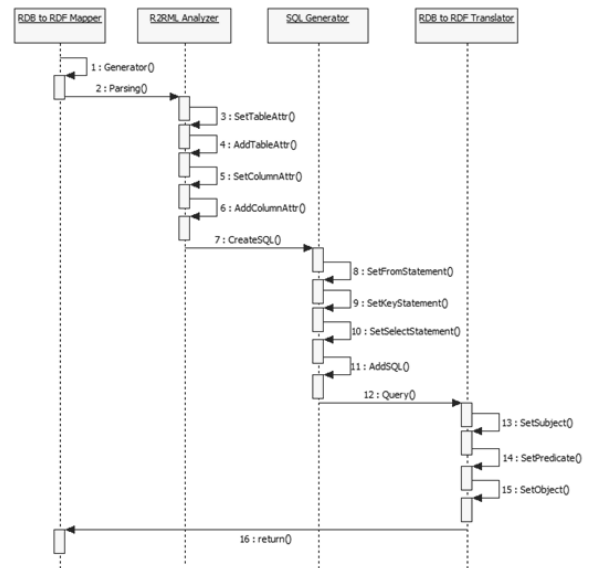


Fig. 2. Sequence Diagram for RDB to RDF Translation

4. 구현

이 장에서는 제안 시스템의 구현 환경과 구현 내용을 보인다. 제안 시스템은 Windows 7 Enterprise K, Inter(r) Core(TM)2 Duo 2.66Hz, 3.25 GB Memory, JRE 1.7.0_15, MySQL 5.613, Eclipse Indigo 로 구성된 환경에서 수행하였다. 제안 시스템을 실험하기 위해 온톨로지 생성도구 인 UBA(Univ-Benchmark Artificial Data Generator)를 이용하여 온톨로지 데이터를 생성한다[23]. 또한 UBA로부터 생성되는 온톨로지를 관계형 데이터베이스에 동일하게 생성한다. 이렇게 생성된 데이터는 제안 시스템과 기존 연구들의 성능을 비교하는데 사용된다.

4.1 R2RML 분석

Fig. 3은 제안 시스템에서 사용한 R2RML 예제를 나타낸다. R2RML은 관계형 데이터베이스 테이블을 트리플사상에 매핑하여 구현한다. 위에 정의한 매핑 규칙에 따라 관계형 데이터베이스에 존재하는 수만큼 트리플사상이 생성된다. 트리플사상 내 논리테이블에는 관계형 데이터베이스 테이블 명 정보가 존재한다. 따라서 논리테이블 정보를 사용하여 관계형 데이터베이스 테이블 명 정보를 얻는다. 다음으로 주어사상에는 주어를 표현할 rr:template을 정의한다. 주어사상에서 정의한 rr:template에 따라 주어는 표현된다. 또한 rr:template은 관계형 데이터베이스 테이블의 기본 키 정보를 가지고 있다. rr:template에 존재하는 {} 안에는 해당 트리플사상에서 사용할 기본 키 정보가 저장되어 있다. 테이블 명, 기본 키 명, rr:template정보와 테이블 명은 TA[]에 저장된다. TA[]는 트리플사상에 대응되기 때문에 관계형 데이터베이스 내 테이블 수만큼 테이블 속성 정보를 가진다. 마지막으로 술어목적어사상은 술어와 목적어 정보를 가지고

있다. RDF의 술어에 표현할 정보를 술어에 표현하며 실제 컬럼 명은 목적어에 rr:column 태그를 사용하여 표현한다. 따라서 술어에 표현된 정보가 실제 RDF의 술어에 표현되며 RDF의 목적어에는 컬럼 값이 표현된다. 제안 시스템은 관계형 데이터베이스 스키마에 RDFS 속성을 지원한다. 따라서 술어목적어사상에는 RDFS 속성을 추가할 수 있다. 예를 들어 rdfs:member 정보를 추가하기 위해서는 술어목적어사상에 해당 정보를 추가하면 된다. Fig. 3은 rdfs:member 정보를 추가한 예제를 나타낸다. 술어에 rdfs:member를 표현하면 제안 시스템은 해당 술어목적어사상은 RDFS 속성 정보를 표현한 것으로 판단하여 목적어를 컬럼 값이 아닌 R2RML의 목적어에 저장된 정보를 사용하여 표현한다. 이렇게 생성된 술어목적어사상 정보는 CA[]에 저장된다. CA[]는 R2RML에서 정의한 술어목적어 수만큼 정의되며 SQL 질의문 생성과 RDB to RDF 변환을 위한 정보로 사용된다.

```
<?xml version="1.0" encoding="euc-kr"?>
<TripleMapList xmlns:rr="http://www.w3.org/ns/r2rml#">
  <tripleMap>
    <rr:logicalTable>
      <rr:tableName>undergraduatestudent</rr:tableName>
    </rr:logicalTable>
    <rr:subjectMap>
      <rr:template>http://www.Department1.
        University1.edu{name}</rr:template>
      <rr:termType>rr:IRI</rr:termType>
      <rr:class>http://semantics.korea.ac.kr/
        Ontologies/UndergraduateStudent </rr:class>
    </rr:subjectMap>
    <rr:prediateObjectMap>
      <rr:prediate>name</rr:prediate>
      <rr:objectMap>
        <rr:column>name</rr:column>
      </rr:objectMap>
    </rr:prediateObjectMap>
    <rr:prediateObjectMap>
      <rr:prediate>http://www.w3.org/2000/01/rdf-
        schema#member</rr:prediate>
      <rr:objectMap>
        <rr:column>memberof</rr:column>
      </rr:objectMap>
    </rr:prediateObjectMap>
  </tripleMap>
</TripleMapList>
```

Fig. 3. An example of R2RML

4.2 RDB to RDF 변환

R2RML 정보를 바탕으로 만들어진 SQL 질의문을 사용하여 관계형 데이터베이스에 저장된 데이터에 접근한다. SQL 질의문을 통해 얻은 결과는 RDB to RDF 변환기 통해 RDF로 변환된다. 이때, TA[], CA[] 정보가 질의 결과를

RDF로 변환하기 위해 사용된다. 질의 결과의 하나의 레코드는 TA[]를 통해 주어를 정의한다. 하나의 레코드는 생성된 주어를 공유하며 술어와 목적어는 컬럼에 해당하는 CA[] 정보를 활용하여 정의한다. Fig. 3의 예제는 주어사상의 rr:template에 http://www.Department1.University1.edu{name}로 정의되어 있다. 따라서 주어는 위 정보를 따르며 해당 레코드의 기본 키 값이 name에 대치된다. 또한 술어목적어사상에 정의된 규칙에 따라 술어와 목적어를 정의한다. Fig. 5은 관계형 데이터베이스 스키마를 나타낸다. Fig. 4의 첫 번째 레코드의 주어는 http://www.Department1.University1.edu/UndergraduateStudent0으로 표현된다. 또한 rr:class 정보를 바탕으로 타입 정보를 설정한다. 각각의 컬럼 정보는 R2RML에 정의된 정보에 따라 술어와 목적어로 변환된다. Fig. 5는 R2RML 정보에 따라 변환된 결과를 나타낸다. 즉, Fig. 5는 Fig. 4의 데이터에 대한 온톨로지 변환 결과를 나타낸다.

emailaddress	memberof	name
UndergraduateStudent0@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent0
UndergraduateStudent0@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent0
UndergraduateStudent1@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent1
UndergraduateStudent1@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent1
UndergraduateStudent1@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent1
UndergraduateStudent10@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent10
UndergraduateStudent10@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent10
UndergraduateStudent100@Department1.University1.edu	http://www.Department1.University1.edu	UndergraduateStudent100

Fig. 4. Relational Database Schema

```
<http://www.Department1.University1.edu/UndergraduateStudent0<http://www.w3.org/2000/01/rdf-schema#type>
<http://semantics.korea.ac.kr/Ontologies/UndergraduateStudent?>.
<http://www.Department1.University1.edu/UndergraduateStudent0<http://www.w3.org/2000/01/rdf-schema#member>
<http://www.Department1.University1.edu>.
<http://www.Department1.University1.edu/UndergraduateStudent0<http://www.w3.org/2000/01/rdf-schema#label>
<UndergraduateStudent?>.
<http://www.Department1.University1.edu/UndergraduateStudent0<mailto:
UndergraduateStudent0@Department1.University1.edu>.
<http://www.Department1.University1.edu/UndergraduateStudent0<memberof>
'http://www.Department1.University1.edu'.
<http://www.Department1.University1.edu/UndergraduateStudent0<name>
'UndergraduateStudent0'.
```

Fig. 5. Extracted ontology from a relational database

4.3 구현 내용

제안 시스템은 사전에 R2RML 정보를 만들어야 한다. R2RML 정보는 접근하고자 하는 데이터베이스에 스키마 정보를 기반으로 만든다. 사전에 만들어진 R2RML 정보는 파싱 함수에 의해 TA[], CA[] 정보를 저장한다. TA[]와 CA[] 정보를 바탕으로 관계형 데이터베이스에 저장된 데이터를 가져오기 위해 SQL 질의문을 생성한다. SQL 질의문은 트리플사상을 기반으로 생성한다. 따라서 트리플사상의 수만큼 SQL 질의문이 만들어진다. 이렇게 만들어진 SQL 질의문을 통해 관계형 데이터베이스에 저장된 데이터에 접근한다. SQL 질의문의 결과는 RDF 변환을 위한 데이터로 사용된다. R2RML에 정의된 규칙에 따라 질의 결과는 주어, 술어, 목적어로 변환된다. 관계형 데이터베이스에 저장된 데이터의 RDF 변환 시간은 1/1,000초 단위로 제공하여 시간에 대한 성능을 평가한다.

5. 실험 및 평가

이 장에서는 RDB to RDF 변환 연구들에 대한 실험 결과를 기술한다. 또한 이 논문에서 제안한 시스템과 기존 연구들의 비교 분석을 통하여 정량적, 정성적 평가에 관해 기술한다.

정량 평가를 위하여 변환된 데이터의 크기와 데이터 변환 시간을 평가한다. 변환된 데이터의 크기는 제안 시스템과 다른 연구들의 변환 데이터의 크기 비교를 통하여 제안 시스템의 온톨로지 변환 양을 비교한다. 데이터 변환 시간은 관계형 데이터베이스의 데이터를 RDF로 변환하는 시간을 의미하며 제안 시스템의 성능을 평가한다. 제안 시스템과 비교하기 위한 평가 대상으로는 RDB에 저장된 데이터를 RDF로 변환하기 위한 연구인 D2RQ, RDBToOnto를 선정하였다. 또한 R2RML을 적용한 RDB to RDF 변환 연구인 Morph를 평가 대상으로 선정하였다. 실험을 위해 D2RQ, RDBToOnto, Morph, 제안 시스템은 동일한 관계형 데이터베이스에서 구현하였다. 데이터 생성은 UBA 온톨로지 생성기를 이용하였으며, 총 50,000개의 튜플을 관계형 데이터베이스에 저장하였다. 또한 모든 비교 실험은 윈도우 시스템에서 실행하였으며 관계형 데이터베이스는 MySQL을 사용하였다.

5.1 정량 평가

관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지로 변환하는 양의 크기를 비교하기 위해 RDB to RDF 변환 데이터의 크기를 비교한다. 실험을 위해 총 10개의 관계형 데이터베이스 저장소를 생성하였다. 각 데이터베이스에는 데이터 크기를 5,000개 튜플 단위로 추가하여 총 50,000개 튜플을 저장하였다. Fig. 6은 제안 시스템에 관계형 데이터베이스의 테이블 명, 컬럼 명 사이에 관계정보를 추가한 변환 결과를 나타낸다. D2RQ와 RDBToOnto의 경우에는 자동으로 추가하는 RDFS 속성 정보를 포함한 결과를 나타낸다. D2RQ와 RDBToOnto는 Morph보다 변환 데이터가 많았다. 관계형 데이터베이스에 저장된 데이터가 동일하지만 데이터 변환 결과가 다른 이유는 D2RQ는 레이블 속성을 변환 데이터에 자동으로 추가하기 때문이다. 또한 RDBToOnto는 RTAXON 학습 방법을 통해 지정된 단어의 경우 자체적으로 RDFS 속성을 추가하는 규칙을 가지고 있다. 따라서 관계형 데이터베이스의 스키마 정보만을 변환하는 Morph에 비해 D2RQ와 RDBToOnto는 많은 변환 결과를 보였다. 하지만 제안 시스템에 관계형 데이터 베이스의 관계 정보를 추가할 경우에는 제안 시스템이 가장 많은 양의 온톨로지를 생성하는 것을 보였다. 또한 제안 시스템은 매핑 정보에 추가 할 수 있는 RDFS 속성 종류가 많다. 따라서 제안 시스템에 관계형 데이터베이스의 관계 정보를 추가할 경우 다른 연구보다 많은 양의 변환 데이터를 생성하는 것을 보인다.

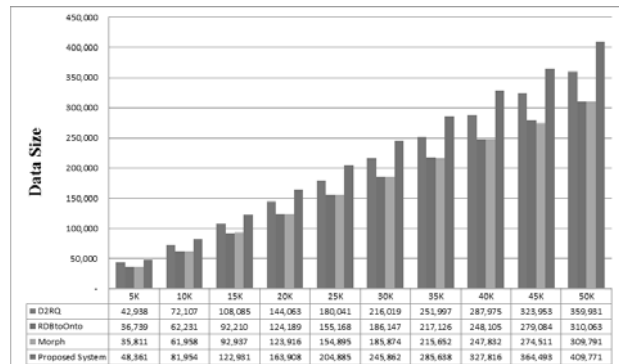


Fig. 6. Result of RDB to RDF Translation

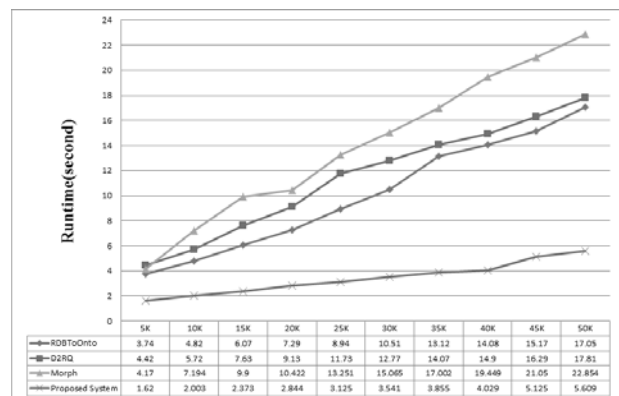


Fig. 7. Comparison of RDB to RDF Translation time

시스템의 성능을 평가하기 위해 RDB to RDF 변환 시간을 비교한다. 변환 시간은 매핑 파일을 작성하는 단계는 고려하지 않는다. Fig. 7는 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지로 변환하는 시간에 대한 비교 결과를 나타낸다. D2RQ, RDBToOnto, Morph, 제안 시스템을 이용하여 관계형 데이터베이스에 저장된 데이터를 RDF 형태의 온톨로지로 변환하는 데 걸리는 시간을 튜플의 개수에 따라 측정하였다. Fig. 7은 제안 시스템이 다른 연구에 비해 가장 빠른 처리 시간을 갖는 것을 보인다. 따라서 제안 시스템은 데이터 변환시간에서 다른 연구에 비해 우수한 성능을 갖는 것을 보인다.

5.2 정성적 평가

5.1절에서는 제안 시스템과 기존의 연구들의 RDB to RDF 변환 시간과 데이터 변환 결과를 비교하였다. 이 절에서는 기존 연구와 제안 시스템의 정성적 평가를 통해 성능 비교 평가를 수행한다. RDB to RDF 변환에 대한 기존 연구들과 제안 시스템의 비교 분석 결과는 Table 6에 나타낸다.

Table 6. A comparative view of RDB to RDF mapping model

Model	Speed of Mapping Generation	Reusability of Mapping Info	Accessibility of Mapping Info	Extension of ontology
D2RQ	high	middle	middle	high
RDBToOnto	high	low	low	middle
Morph	middle	high	high	low
Proposed System	middle	high	high	high

매핑 생성 속도는 매핑 정보 생성시 소요되는 시간을 구분하기 위한 평가 요소이다. 매핑 정보생성 방식은 자동, 반자동, 수동방식으로 구분되며 자동 생성 방식이 가장 우수하다. 하지만 관계형 데이터베이스 스키마 정보만을 변환하는 단점을 가진다. D2RQ와 RDBToOnto는 매핑 정보를 자동으로 생성하는 반면 제안 시스템과 Morph는 매핑 정보를 수동으로 생성한다. 따라서 D2RQ와 RDBToOnto가 매핑 정보 생성에서는 우수한 성능을 보인다. 매핑 언어의 재사용성은 시스템 독립적인 매핑 언어 사용 여부를 보여주는 평가 요소이다. 시스템 종속적인 매핑 언어를 사용하면 매핑 언어에 대한 이해가 필요하다. 따라서 시스템마다 다른 매핑 언어의 사용은 매핑 언어 사용의 재사용성을 떨어트린다. D2RQ와 RDBToOnto는 각각의 매핑 언어를 사용하며 Morph와 제안 시스템은 R2RML을 사용한다. 이와 같은 이유로 Morph와 제안 시스템은 매핑 언어의 재사용성이 높다. 매핑 언어의 접근성은 표준화된 매핑 언어 사용 여부를 보여주는 평가 요소이다. Morph와 제안 시스템은 표준화된 매핑 언어를 사용하여 매핑 언어의 접근성이 높다. 또한 표준화된 매핑 언어의 사용으로 매핑 언어에 학습에 들어가는 비용이 적다. 마지막으로 온톨로지 확장성은 변환 된 온톨로지에 관계형 데이터베이스의 테이블 명, 컬럼 명 사이에 관계 정보를 추가할 수 있는지 여부를 확인할 수 있는 평가 요소이다. 제안 시스템은 매핑 정보를 수정하여 관계형 데이터베이스의 테이블 명, 컬럼 명 사이에 관계정보에 대한 온톨로지를 추가하는 장점을 가진다. 이는 D2RQ와 동일한 기능이며 Morph에서는 지원하지 않는 기능이다. 따라서 제안 시스템은 D2RQ와 같은 우수한 성능을 보이며 표준화된 매핑 언어 사용을 통해 높은 매핑 정보의 접근성과 재사용성을 가진다.

6. 결 론

시맨틱 웹 환경에서 기존의 웹에 저장된 데이터에 접근을 위해 많은 연구가 이루어져 왔다. 기존 웹에 저장된 데이터 구조는 시맨틱 웹에서 사용하는 데이터 구조와 다르게 구성

되어 있다. 서로 다른 데이터 구조로 인해 시맨틱 웹에서는 기존 웹에 저장된 데이터를 이용할 수 문제점을 가진다. 하지만 기존의 웹에 저장된 데이터를 버리고 새로운 시맨틱 웹 데이터를 구축하는 것은 많은 시간과 비용을 초래한다. 따라서 기존 웹에 저장된 데이터를 시맨틱 웹 환경에 맞게 변환하는 방법의 필요성이 대두되었다. 대부분의 변환 방법은 매핑 방식을 이용하여 관계형 데이터베이스 구성요소와 RDF 구성요소를 연결하는 방식을 사용한다. 하지만 매핑 언어의 표준화가 이뤄지지 않아 시스템마다 다른 매핑 언어를 사용하였다. 따라서 W3C에서는 매핑 언어의 표준을 위해 R2RML을 제안하였다. 하지만 R2RML은 관계형 데이터베이스의 테이블명, 컬럼명, 저장된 데이터를 온톨로지로 변환하지는 않는다. 따라서 관계형 데이터베이스에 저장된 데이터를 단순히 RDF로 변환하는 것에 그친다. 풍부한 의미 정보가 있지 않은 온톨로지는 사용에 제한적이다.

이 논문에서는 관계형 데이터베이스 구성 요소의 의미 관계를 고려한 RDB to RDF 매핑 시스템을 제안하였다. 제안 시스템은 R2RML을 매핑 언어로 사용하며 R2RML에 저장된 정보를 이용하여 SQL 질의문을 생성한다. 위와 같이 생성된 SQL 질의문을 통해 관계형 데이터베이스에 저장된 데이터에 접근한다. SQL 질의문을 통해 얻은 관계형 데이터베이스에 저장된 데이터는 R2RML의 정해진 규칙에 따라 RDF로 변환된다. 제안 시스템은 추가적으로 R2RML에 RDFS 속성을 추가하여 데이터의 풍부한 의미 표현을 지원하였다. 제안 시스템에 대한 평가로 정량평가와 정성평가를 수행하였다. Morph, RDBToOnto, D2RQ를 비교 대상으로 선정하고, 변환된 데이터의 크기와 변환 시간을 측정하였다. 그 결과 제안 시스템은 신뢰성을 보장할 수 있으며 가장 빠른 변환 시간을 보여 우수한 성능을 보였다. 또한 정성적 비교 평가의 결과로 수동으로 매핑을 생성해야 하지만 표준화된 언어를 이용하여 매핑 정보에 높은 접근성과 재사용성을 보였다. 마지막으로 관계형 데이터베이스의 테이블 명, 컬럼 명 사이에 관계 정보를 추가하여 높은 온톨로지 확장성을 보였다.

참 고 문 헌

- [1] Jeen Broekstra, Arjohn Kampman, Frank van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," Springer-Verlag, Lecture Notes in Computer Science, Vol.LNCS 2342, pp. 54-68, 2002.
- [2] Jena-A Semantic Web Framework for Java, <http://jena.sourceforge.net>
- [3] Oracle, http://download.oracle.com/otndocs/tech/semantic_web/pdf/oradb_semantic_overview.pdf
- [4] SDB, <http://jena.hpl.hp.com/wiki/SDB>

[5] OWLJessKB : A Semantic Web Reasoning Tool, <http://edge.cs.drexel.edu/assemblies/software/owljesskb>

[6] Steve Harris, "SPARQL query processing with conventional relational database systems," Springer Verlag, Lecture Notes in Computer Science, Vol. LNCS 3807, pp. 235-244, 2005.

[7] Dongwon Jeong, Myoung-hoi Choi, Yang-Seung Jeon, Youn-Hee Han, Laurence T. Yang, Young-Sik Jeong, and Sung-Kook Han, "Persistent Storage System for Efficient Management of OWL Web Ontology," Springer-Verlag, Lecture Notes in Computer Science, Vol. LNCS 4611, pp.1089-1097, 2007.

[8] RDB2RDF, <http://www.w3.org/2001/sw/rdb2rdf/>

[9] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau, S. Auer, and J. Sequeda, A. Ezzat, "A Survey of Current Approaches for Mapping of Relational Databases to RDF," Technical Report, 2009.

[10] Cullot, N., Ghawi, R., Yetongnon, K., "DB2OWL: A Tool for Automatic Database-to-Ontology Mapping," In Proceedings of 15th Italian Symposium on Advanced Database Systems, pages 491-494, Torre Canne, Italy, 2007.

[11] Bizer C., Cyganiak, R., "D2RQ-Lessons Learned," Position paper for the W3C Workshop on RDF Access to Relational Databases, Cambridge, USA, 2007.

[12] R2RML, <http://www.w3.org/TR/r2rml/>

[13] Morph, <http://github.com/jpcik/morph>

[14] DB2Triples, <http://github.com/antidot/db2triples>

[15] A. Chebotko, S. Lu, H.M. Jamil, F. Fotouhi, "Semantics preserving SPARQL-to-SQL query translation for optional graph patterns," Technical Report TR-DB-052006-CLJF, Wayne State University, 2006.

[16] R. Cyganiak, "A relational algebra for SPARQL," Technical Report HPL-2005-170, Hewlett-Packard Laboratories, 2005.

[17] S. Harris, N. Shadbolt, "SPARQL query processing with conventional relational database systems," in Proceedings of the International Workshop on Scalable Semantic Web Knowledge Base Systems, pp. 235-244, 2005.

[18] Bizer, C. "D2R MAP-A Database to RDF Mapping Language," The twelfth international World Wide Web Conference, WWW2003, Budapest, Hungary, 2003.

[19] Prud'Hommeaux, Eric, and Andy Seaborne. "SPARQL query language for RDF," W3C recommendation 15, 2008.

[20] Bizer, C., Seaborne, "D2RQ-Treating Non-RDF Databases as Virtual RDF Graphs," in Proceedings of the Third International Semantic Web Conference, 2004.

[21] Fabrid Cerbah, "Learning Highly Structured Semantic Repositories from Relational Databases-The RDBToOnto Tool," in Proceedings of the 5th European Semantic Web

Conference, Tenerife, Spain, 2008.

[22] Cerbah. F. "Mining the Content of Relational Databases to Learn Ontologies with Deeper Taxonomies," in, Sydney, Australia, 2008.

[23] Guo, Y., Pan, Z., and Heflin. J., "An Evaluation of Knowledge Base Systems for Large OWL Datasets," Springer Verlag, Third International Semantic Web Conference, Vol. LNCS 3298, pp. 274-288, 2004.

성 하 정



e-mail : octoom@korea.ac.kr
 2006년 인하대학교 컴퓨터 공학과(학사)
 2012년~현 재 고려대학교 컴퓨터·전과
 통신공학과 석사과정
 관심분야: 온톨로지, 시맨틱 웹, 메타데이터
 레지스트리

김 장 원



e-mail : jangwon@kisti.re.kr
 2005년 상명대학교 소프트웨어공학과(학사)
 2005년 한국과학기술연구원(KIST) 위촉연
 구원
 2008년 고려대학교 컴퓨터학과(이학석사)
 2012년 고려대학교 컴퓨터·전과통신공학과
 (공학박사)
 2013년~현 재 한국과학기술정보연구원 선임연구원
 관심분야: 온톨로지, 시맨틱 웹, GIS, 데이터베이스, 메타데이터,
 지식적 분석 등

이 석 훈



e-mail : leha82@korea.ac.kr
 2009년 고려대학교 전자 및 정보공학부
 (학사)
 2011년 고려대학교 컴퓨터·전과통신공학
 과(공학석사)
 2011년~현 재 고려대학교 컴퓨터·전과
 통신공학과 박사과정
 관심분야: 온톨로지, 데이터마이닝, 메타데이터 레지스트리, 자율
 컴퓨팅 등



백 두 권

e-mail : baikdk@korea.ac.kr

1974년 고려대학교 수학과(학사)

1977년 고려대학교 산업공학과(석사)

1983년 Wayne State Univ. 전산학과(석사)

1985년 Wayne State Univ. 전산학과(박사)

1986년~2013년 고려대학교 컴퓨터·전파

통신공학과 교수

1989년~1991년 고려대학교 전산학과 학과장

1990년~1991년 미국 Arizona대학교 객원 교수

1991년~2013년 ISO/IEC JTC1/SC32 전문위원회 위원장

1993년~1999년 한국과학기술원 객원책임연구원

1993년~1999년 한국DB진흥센터 표준연구위원

1996년~1997년 고려대학교 컴퓨터과학기술연구소(초대소장)

1997년~1998년 고려대학교 정보전산원 원장

1998년~1999년 한국정보과학회 전산교육연구회 운영위원장

1999년~2001년 정보통신진흥협회 데이터기술위원회 의장

2002년~2004년 고려대학교 정보통신대학(초대학장)

2002년~2003년 한국시물레이션학회 회장

2003년~현 재 정보통신부 컴퓨터프로그램보호위원회 위원

2004년~2005년 한국정보처리학회 부회장

2005년~2008년 한국소프트웨어진흥원 이사

2009년~2010년 고려대학교 정보통신대학 학장

2013년~현 재 고려대학교 융합소프트웨어전문대학원 교수

2013년~현 재 ISO/IEC JTC1/SC32 전문위원회 전문위원

관심분야: 메타데이터, 소프트웨어공학, 데이터공학, 컴포넌트기
반 시스템, 메타데이터 레지스트리, 프로젝트 매니지
먼트 등