

# Jena 저장소 구조를 TRIPLES 구조로의 사상을 위한 뷰 생성 도구 개발

지종진<sup>1</sup> 손지성<sup>2</sup> 정동원<sup>1</sup>

군산대학교 정보통계학과<sup>1</sup> 고려대학교 컴퓨터·전파통신 공학과<sup>2</sup>

e-mail: {coolgo, djeong}@kunsan.ac.kr<sup>1</sup>, redfunky07@korea.ac.kr<sup>2</sup>

## A Development of View Generation Tool for Mapping the Jena Storage Structure to TRIPLES Structure

Jongjin Ji<sup>1</sup> Jiseong Son<sup>2</sup> Dongwon Jeong<sup>1</sup>

Dept. of Informatics & Statistics, Kunsan National University<sup>1</sup>

Dept. of Computer and Radio Communication Engineering, Korea University<sup>2</sup>

e-mail: {coolgo, djeong}@kunsan.ac.kr<sup>1</sup>, redfunky07@korea.ac.kr<sup>2</sup>

### 요 약

웹 온톨로지 언어의 개발에 따라 웹 온톨로지를 저장하기 위한 다양한 저장소들이 개발되었으며 이러한 웹 온톨로지 저장소는 관계형 데이터베이스 모델을 기반으로 하고 있다. 또한 웹 온톨로지 데이터를 검색하기 위한 질의 언어로서 W3C에서 SPARQL을 제안하였다. 웹 온톨로지 저장소로서 관계형 데이터베이스가 활발히 개발되고 SPARQL의 이용이 증가함에 따라 SPARQL을 SQL로 변환 하는 알고리즘의 필요성이 대두되었다. 지금까지 제안된 변환 알고리즘들은 SPARQL-to-SQL 변환 알고리즘이 저장소 구조에 종속적이라는 문제점이 있다. 이를 해결하기 위하여 관계형 뷰를 기반으로 저장소에 독립적인 변환 알고리즘을 활용할 수 있는 모델이 제안되었다. 그러나 제안된 모델에서 변환 알고리즘을 독립적으로 활용하기 위해서는 사전에 사용자가 SQL 문을 작성하여 수동적으로 뷰를 생성해 주어야 한다. 이 경우 수동적인 SQL 작성으로 인한 불편함과 사용자의 실수로 인한 SQL문의 오류가 생길 수 있다는 문제점이 있다. 따라서 이 논문에서는 이러한 문제점을 해결하기 위하여 다양한 웹 온톨로지 저장소 모델 중 OWL 관련 시스템 개발에 가장 많이 사용되고 있는 Jena 저장소를 이용하여 독립적 저장소 활용을 위한 뷰 생성 도구인 Jena-to-TRIPLES 변환 도구를 제안한다.

키워드 : Jena, TRIPLES, 사상 도구, 관계형 데이터베이스, 웹 온톨로지

### 1. 서론

현재 웹의 정보 이용자는 자신이 원하는 형태로 정보를 제공받기 원한다. 지금까지의 거의 무한으로 확장되고 필요한 정보와 불필요한 정보가 혼재된 인터넷 환경에서 사용자가 원하는 정보만을 검색하기 위하여 시맨틱 웹이 정의 되었다

[1]. 시맨틱 웹에서 중요시 되는 기술은 웹 온톨로지 기술로 웹 상에 존재하는 자원들을 정의하고 자원들 사이의 관계를 나타내는 기술이다. 이 웹 온톨로지 정보들은 컴퓨터가 이해할 수 있는 형태로 표현되는데 이러한 표현을 위해 사용되는 것이 웹 온톨로지 언어이며 대표적으로 RDF(Resource Description Framework)[2], RDF Schema(Resource Description Framework Schema)[3], OWL(Web Ontology Language)[4] 등이 제안되었다.

또한 웹 온톨로지 데이터를 저장하기 위하여 다양한 저장

\* 이 연구에 참여한 연구자는 'BK 21 2단계 사업'의 지원을 받았음.

소들이 제안되었다. 제안된 저장소들 중 대부분은 관계형 데이터베이스를 기반으로 하고 있다(5-10).

웹 온톨로지 저장소로서 관계형 데이터베이스가 활발히 개발되고 SPARQL의 이용이 증가함에 따라 SPARQL을 SQL로 변환하는 알고리즘이 요구되었으며 기존에 제안된 SPARQL-to-SQL 변환 알고리즘은 각각의 알고리즘이 종속적이라는 문제점을 가지고 있었다. 따라서 다양한 웹 온톨로지 저장소에 대하여 관계형 뷰를 기반으로 변환 알고리즘을 독립적으로 유지시키는 모델이 제안되었다(11).

그러나 이 모델에서는 뷰를 생성할 때 사용자가 SQL 문을 작성하여 수동적으로 생성한다는 단점이 있다. 이는 SQL 작성 시 사용자 실수에 의한 오류를 초래한다.

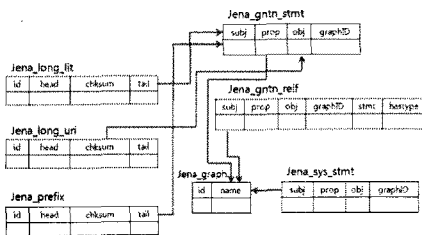
이 논문에서는 앞 절에서 제시된 독립적인 변환 알고리즘을 사용할 때 사용자가 보다 용이하게 뷰를 생성할 수 있도록 여러 OWL 관련 시스템에서 사용되고 있는 Jena 저장소 구조를 이용하여 물리적 공간인 저장소와 변환 알고리즘을 독립적으로 유지하기 위해 사용되는 TRIPLES 뷰를 생성하는 도구를 제안한다.

## II. 관련 연구

### 1. 웹 온톨로지 저장소

#### (1) Jena 저장구조

그림 1은 기본적으로 Jena 저장소(12)를 구성하는 총 7개의 테이블로 구성된 구조를 보여준다.



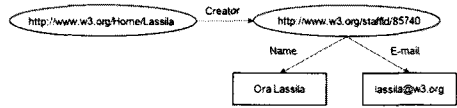
〈그림 1〉 Jena 저장소 구조

실질적으로 데이터를 저장하는 테이블은 jena\_gntn\_stmt이며 jena\_gntn\_stmt 테이블에 저장될 값들 중 256byte 이상의 데이터가 들어갈 경우 리터럴과 URI 정보가 저장되는 별도 테이블인 jena\_long\_lit와 jena\_long\_uri를 구성하고 이를 참조 하는 형태로 테이블이 구성되어 있다. jena\_long\_prefix 테이블은 공통적으로 사용하는 접두사에 대해서 저장하고 이를 참조하는 테이블이다.

#### (2) TRIPLES 저장소

그림 2와 같이 RDF 문서는 주어(subject), 술어(predicate), 목적어(object)의 TRIPLE 구조를 갖는 문장들로 이루어지며, TRIPLES 저장소는 웹 온톨로지 데이터를 주어, 술어,

목적어로 나누어 저장하는 저장소를 말한다(13).



Subject	Predicate	Object
http://www.w3.org/staff/id/85740	Creator	http://www.w3.org/staff/id/85740
http://www.w3.org/staff/id/85740	Name	Ora Lassila
http://www.w3.org/staff/id/85740	E-mail	lassila@w3.org

〈그림 2〉 웹 온톨로지 그래프와 저장

### 2. SPARQL-to-SQL 변환 알고리즘

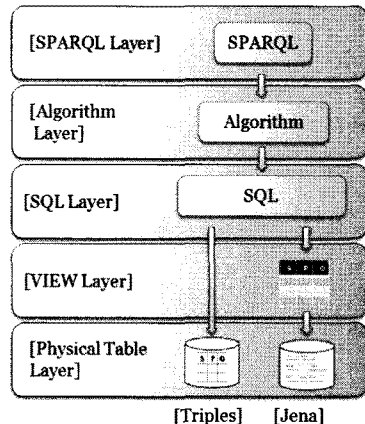
SPARQL의 이용이 증가하고 대부분의 관계형 데이터베이스 기반의 저장소를 이용함에 따라 SPARQL을 SQL로 변환하는 알고리즘 개발이 대두되었으며 Chobotko 알고리즘(14), Jena의 sparql2sql(15), Harris 알고리즘(16) 등이 있다.

#### (1) Chobotko 알고리즘

Chebotko 알고리즘은 중첩 OPTIONAL 절을 SQL로 변환하는 과정에서 기존에 제안된 변환 알고리즘 중 높은 기능성을 제공하지만 SPARQL의 UNION, FILTER 구문에 대한 변환 기능은 제공하지 않는다. 이 알고리즘은 다른 여러 구조의 저장소들 중에서 가장 간단한 구조인 TRIPLES 저장소를 사용한다(17).

#### (2) Jena의 sparql2sql과 Harris 알고리즘

SPARQL의 UNION과 FILTER 구문 변환을 지원하지 않고 중첩 OPTIONAL 절 변환에 한계가 있다. 또한 각각의 알고리즘은 특정 저장소 구조에 대하여 종속적이다.



〈그림 3〉 저장소와 독립적인 알고리즘 모델

### 3. 저장소에 독립적인 변환 알고리즘 활용모델

그림 3은 기존의 종속적인 저장소 알고리즘이 아닌 여러 웹 온톨로지 저장소 모델 중 저장소에 종속적이었던 변환 알고리즘

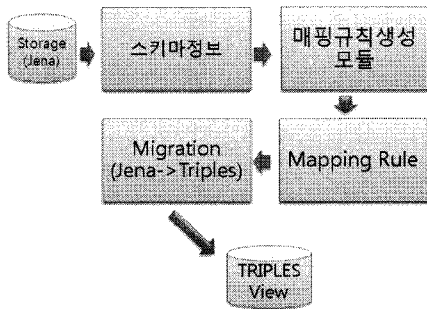
이 아닌 독립적인 알고리즘을 적용하기 위한 모델이다. Alogrithm 계층은 독립적 변환 알고리즘 계층으로 Chebotko 알고리즘을 사용하였다. SQL 계층은 Chebotko 알고리즘을 통해 생성된 SQL이며 View 계층은 변환 알고리즘과 저장소를 독립적으로 유지시켜준다. 그림에서 보는 바와 같이, TRIPLES 구조가 아닌 다른 저장소에 TRIPLES 구조의 뷰를 생성해준다. 따라서 저장소의 구조가 수정되어도 알고리즘의 수정 없이 질의를 실행할 수 있다. 하지만 View 계층에 생성된 뷰는 사용자가 SQL을 이용하여 수동적으로 생성해주어야 한다.

### III. View 생성 도구 설계

이 장에서는 앞서 언급한 수동적인 뷰 생성시 발생하는 문제점을 보완하기 위한 뷰 생성도구에 대하여 기술한다. 이 논문에서는 여러 온톨로지 저장소 중 Jena 저장소 구조를 TRIPLES 구조의 뷰로 사상을 위해 필요한 규칙과 주요 알고리즘 등에 대하여 기술한다.

#### 1. 전체적인 프로세스

그림 4는 Jena 저장소의 물리적 테이블 구조를 TRIPLES 구조의 뷰로 사상할 때의 처리 과정을 보여준다.



〈그림 4〉 Jena저장소를 위한 뷰 생성 도구 구조

우선 물리적 저장소인 Jena 구조의 스키마 정보를 도구에 나타낸다. 나타낸 Jena 저장소의 인스턴스 값들과 TRIPLES 구조와 사상될 값들을 모듈에 넣는다. 모듈에 넣은 값들을 도구에서 정의한 사상 규칙에 의하여 Jena 구조를 TRIPLES 구조의 뷰로 생성한다.

#### 2. 사상규칙정의

사상 규칙을 정의하기 전에 이와 관련된 Jena 구조에 대하여 상세하게 기술하면 다음과 같다.

먼저, 입력으로 주어진 OWL 파일을 분석하여 분석된 데이터 정보들을 SUBJ, PROP, OBJ의 칼럼으로 구성된 하나의 jena\_gntn\_stmt 테이블에 분류·저장한다. 단, jena\_long\_lit, jena\_long\_uri, jena\_long\_prefix 테이블에는 데이터 값이 항상

저장되지 않으므로 데이터를 추출할 때의 조건으로 데이터의 길이가 256바이트가 넘을 경우에만 jena\_long\_lit, jena\_long\_uri, jena\_long\_prefix 테이블에서 데이터를 추출한다.

그림 5는 Jena 저장소의 실질적인 데이터를 저장하는 Jena\_gntn\_stmt 테이블의 필드 값을 읽어 TRIPLES 구조의 뷰에 사상하는 과정을 보여준다.



〈그림 5〉 사상 규칙에 따른 뷰 생성 과정

위와 같은 Jena 저장 구조와 TRIPLES 구조 특성 분석 결과를 바탕으로 뷰 생성을 위한 규칙을 다음과 같이 정의할 수 있다.

```
Jena_gntn_stmt.SUBJ -> TRIPLES.Subject
Jena_gntn_stmt.PROP -> TRIPLES.Predicate
Jena_gntn_stmt.OBJ -> TRIPLES.Object
```

#### 3. 주요 알고리즘

이 절에서는 연구의 가장 핵심 알고리즘의 사상 및 뷰 생성 알고리즘에 대하여 기술한다.

```
1. DefaultTableModel model = (DefaultTableModel)
   table.getModel();
2. int cc = model.getColumnCount();
3. int rr = model.getRowCount();
4. for(int i = 0; i<rr; i++){
5.   for (int ii = 0; ii<cc; ii++){
6.     String fasd = (String) model.getValueAt(i, ii);
7.     vmt.add(fasd:);}
8. int vs= vmt.size();
9. ddl = (String)vmt.get(0);
10. StringTokenizer d1 = new StringTokenizer(dd1,",");
11. while (d1.hasMoreToksens()) {
12.   dv1.add(new String(d1.nextToken()));}
13. dd1 = (String) dv1.get(1);
14. tt = (String)dv1.get(0);
15. dv1.clear();
16. StringTokenizer d11 =
   new StringTokenizer(dd1,",");
17. while (d11.hasMoreToksens()){
18.   dv1.add(new String(d11.nextToken()));}
19. dd1=(String)dv1.get(0);
20. String sql = "CREATE VIEW TRIPLES
   (subject, predicate, object) " +
21. "AS SELECT " + dd1 + "," + dd2 + "," + dd3 +
   " FROM " + tt;
22. stmt.executeUpdate(sql);
```

위 알고리즘에서, 1~3은 도구에 추가된 TRIPLES 구조로 사상될 필드명과 같은 메타정보를 추출해 되는 연산을 수행한다.

4~19는 테이블의 열의 크기와 같은 메타정보를 읽어와 그 중 첫 번째 열을 필드명을 변수 dd1에 저장한다. 이러한 동일한 연산을 통해 두 번째와 세 번째 열에 대한 메타정보를 읽어들이 변수 dd2와 dd3에 각각 저장한다. 테이블명 또한 변수 tt에 저장한다.

20~21은 TRIPLES 구조의 뷰를 생성하기 위한 SQL 질의문을 작성하는 부분이며, 마지막으로 22는 질의문을 실행하여 뷰를 생성한다.

#### IV. 구현 및 평가

이 장에서는 앞 장에서 기술했던 사상 규칙 등을 적용하여 Jena 구조를 TRIPLES 구조와 동일한 구조로 된 뷰를 생성하는 도구 개발 내용 및 평가에 대하여 기술한다.

##### 1. 구현

그림 6은 Jena 저장소 구조, 매핑구조, TRIPLES 구조로 구성되어 있다. 데이터베이스 구조는 ①번과 ②번으로 구성되어 있으며 ①번은 데이터베이스에서 Jena 저장소 구조를 나타내고 ②번은 Jena 테이블의 인스턴스 값들을 보여준다. 매핑구조의 ③번은 ①번장에서 사상할 테이블명과 필드명을 추가하는 창이고 ④번과 ⑤번의 창에서 사상될 TRIPLES 구조의 테이블과 필드를 추가한다. ⑥번은 TRIPLES 뷰 생성을 위한 SQL 문을 생성하기 전 사상할 값들을 ⑥번에 추가하기 위한 버튼이다.

⑥번은 테이블 구조로 만들어져 있으며 사상된 필드 값을 확인하기 위한 테이블이다. ⑦번은 사상된 필드 값들을 사상규칙에 의해 SQL 문으로 변환하여 TRIPLES 구조의 뷰를 생성한다.

그림 7은 사상 도구 실행 초기 화면을 보여준다. Jena의 저장소 구조를 읽은 후 Jena\_glt1\_stmt 필드명과 인스턴스 값을 보여주는 화면이다.

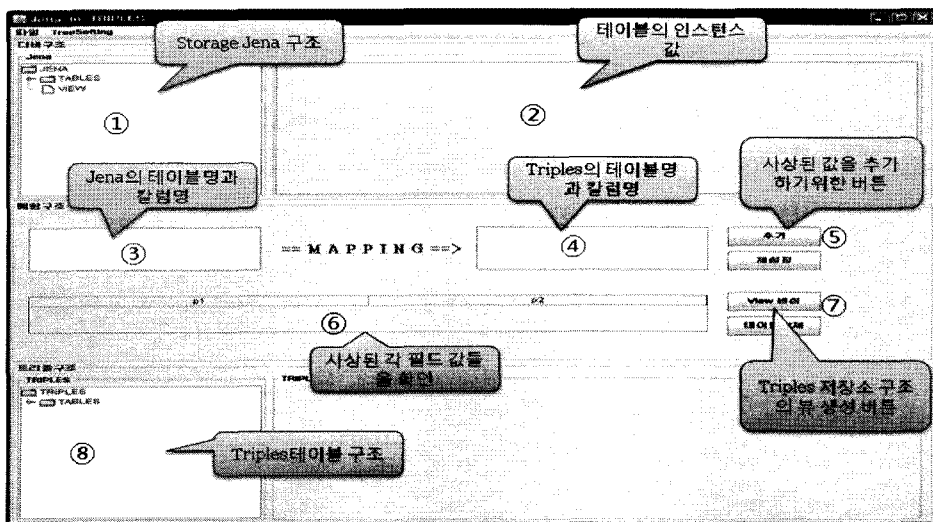
그림 8은 Jena 저장소 구조를 TRIPLES 구조의 뷰로 변환하기 위하여 정의한 사상 규칙을 보여준다. Jena 저장소 구조를 읽은 후 Jena\_glt1\_stmt의 필드명인 obj와 TRIPLES의 object를 확인하여 사상을 위해 디스플레이 된 화면을 보여준다.

그림 9는 "View생성" 버튼을 실행 후 TRIPLES 저장 구조로 생성된 뷰를 보여주는 화면으로 사상을 위해 가져온 테이블명과 필드명의 확인을 위해 테이블로 사상될 테이블명과 필드명을 보여준 후 "View생성" 버튼을 클릭 생성된 TRIPLES 뷰를 데이터베이스에서 읽어와 필드명과 사상된 인스턴스 값을 보여준다.

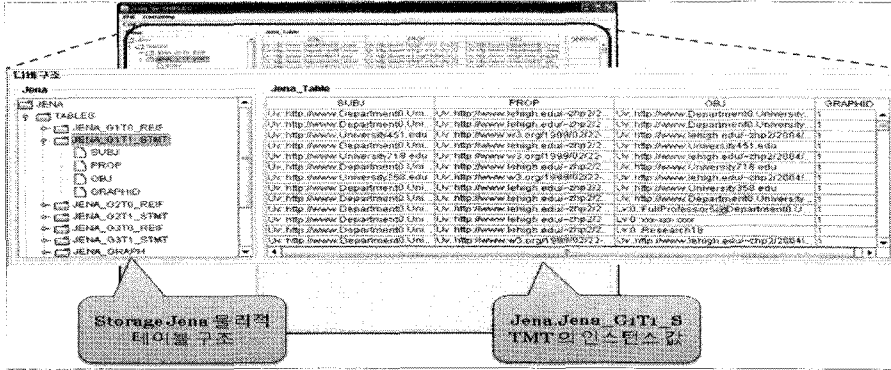
##### 2. 검증 및 평가

뷰 생성 도구의 검증을 위하여 도구의 사상 규칙에 의해 생성된 TRIPLES 뷰의 인스턴스 값이 기존의 Jena 저장소의 인스턴스 값과 동일한지를 확인하고 또한 TRIPLES의 뷰를 역변환하여 기존의 Jena 저장소 구조와 인스턴스 값이 동일한지를 확인하였다. 이러한 검증 방법을 통하여, 제한한 도구를 사용할 경우 데이터의 손실 없이 뷰를 생성할 수 있는 것을 알 수 있었다.

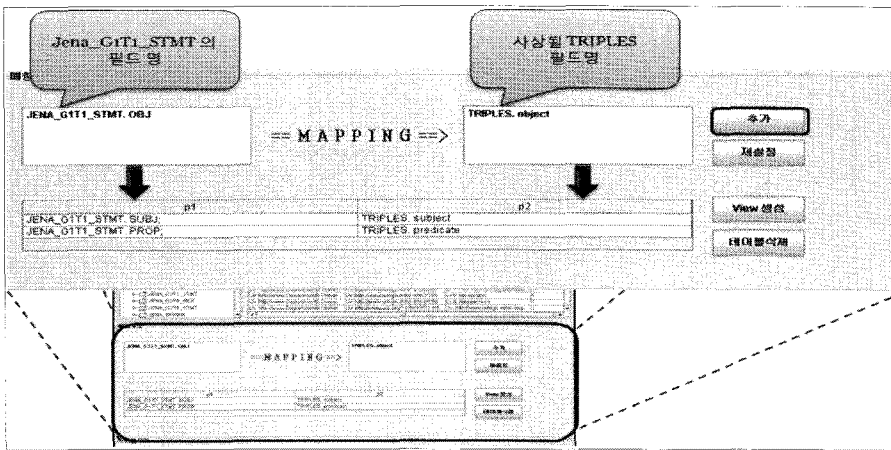
기존의 방법으로 변환 알고리즘을 활용하기 위하여 뷰 생성할 때에는 수동으로 사용자가 직접 SQL 문을 작성해야 하는 번거로움이 있었다. 그러나 제안한 도구를 이용하여 사상된 규칙에 따라 뷰를 자동적으로 생성할 경우 사용자가 직접 SQL 문을 작성해야 하는 번거로움과 사용자의 실수로 인한 SQL 오류를 줄일 수 있었다. 따라서 제안 도구는 사용자에게 뷰 생성의 편의성을 제공한다.



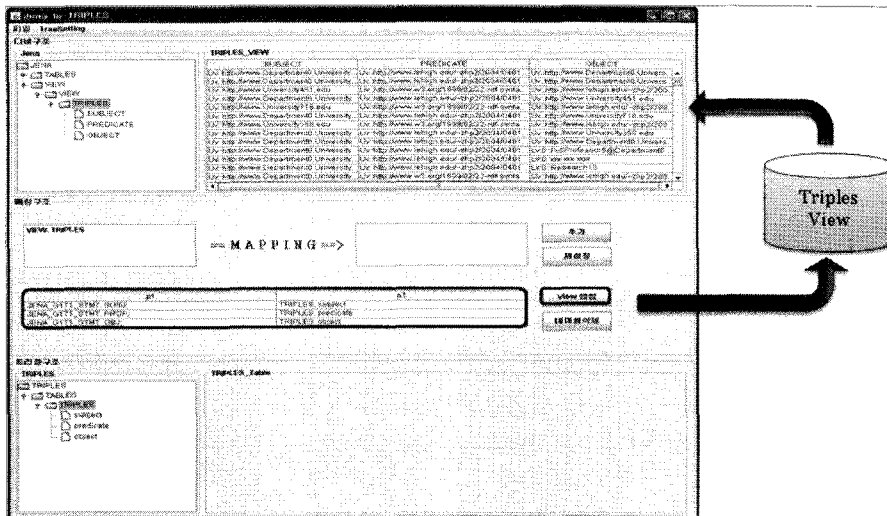
<그림 6> 뷰 생성 도구 초기 화면



<그림 7> 뷰 생성 도구 실행 초기 화면



<그림 8> 저장구조 Mapping하는 화면



<그림 9> 사상규칙에 의해 생성된 뷰를 보여주는 화면

## V. 결론

이 논문에서는 기존에 제안된 SPARQL-to-SQL 변환 알고리즘의 저장소 독립적 활용을 위한 시스템 모델[11]에서 사용자가 수동적으로 뷰를 생성하는 것이 아닌 자동적으로 TRIPLES 구조의 뷰를 생성해주는 도구를 제안하였다. 이 도구에는 사상 규칙이 정의되어 있어 사용자가 뷰를 생성함에 있어 편의성을 제공한다.

그러나 현재 제안한 도구는 다양한 저장소 구조 중 하나의 저장소 구조에 대해서만 적용한 도구이기 때문에 다른 저장소 구조에서는 사용할 수 없다. 따라서 향후 연구로써 다양한 저장소 구조에 대하여 사상 규칙을 정의한 도구 개발이 요구된다.

## 참고문헌

- [1] Tim Berners-Lee, James Hendler, and Ora Lassila, "The Semantic Web", Scientific American, Vol.284, No.5, pp.33-43, May 2001.
- [2] RDF/XML Syntax Specification, <http://www.w3.org/TR/rdf-syntax-grammar>, Feb.2004.
- [3] RDF Vocabulary Description Language 1.0: RDF Schema, <http://www.w3.org/TR/rdf-schema>, Feb.2004
- [4] OWL Web Ontology Language Reference, <http://www.w3.org/TR/owl-ref>, Feb. 2004.
- [5] Sesame: RDF schema querying and storage, <http://www.openrdf.org/>.
- [6] Zhengxiang Pan, Jeff Heflin, "DLDB: Extending relational databases to support Semantic Web queries, "Workshop on Practical and Scalable Semantic Web Systems, ISWC 2003, pp. 109-113, 2003.
- [7] Steve Harris, "SPARQL query processing with conventional relational database systems," Springer-Verlag, Lecture Notes in Computer Science (LNCS), Vol. LNCS 3807, pp. 235-244, 2005.
- [8] OWLJessKB: A Semantic Web Reasoning Tool, <http://edge.cs.drexel.edu/assemblies/>.
- [9] Jena Semantic Web Framework, <http://jena.sourceforge.net/>.
- [10] Dongwon Jeong, Myounghoi 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 (LNCS), Vol. LNCS 4611, pp.1089-1097, July 2007.
- [11] Jiseong Son, Dongwon Jeong, Jinhyung Kim, Doo-Kwon Baik, "System Model for Storage-Independent Use of SPARQL-to-SQL Translation Algorithm", 한국정보과학회 2007 가을 학술발표 논문집, Vol. 34, No. 2, pp. 3-4, October 2007.
- [12] K. Wilkinson, C. Sayers, H. Kuno, D. Reynolds, Efficient RDF Storage and Retrieval in Jena2, Proc. of the 1st International Workshop on Semantic Web and Databases, pp. 131-151, 2003
- [13] Tae-Whi Lee, Ki-Sung Kim, Sang-Won Yoo, Hyoung-Joo Kim, "an Incremental Update Algorithm on RDF Documents in Triple Stores", The Korean Institute of Information Scientists and Engineers(KIISE), Journal of KISE: Databases, Vol. 33, No. 5, pp. 476-485, 2006.
- [14] Artem Chebotko, Shiyong Lu, Hasan M. Jamil, and Farshad Fotouhi, "Semantic Preserving SPARQL-to-SQL Query Translation for Optional Graph Patterns", Technical Report TR-DB-052006-CLJF, May 2006, Revised, November 2006.
- [15] spatql2sql - a query engine for SPARQL over Jena triples stores, <http://jena.sourceforge.net/>.
- [16] SPARQL Query Language for RDF, W3C, 4 October 2006.
- [17] Yannis Theoharis, vassilis Christophides, and Grigoris Karvounarakis, "Benchmarking database representation of RDF/S stores, "Springer-Verlag, Lecture Notes in Computer Science(LNCS), Vol. LNCS 3729, pp. 658-701, 2005.