

RDF 데이터 관리를 위한 효율적인 질의 처리에 관한 연구

A Study on Querying Method for RDF Data in XML Database

남궁 황(Namgoong-Hwang)*

김 용(Kim-Yong)**

〈 목 차 〉

- | | |
|-----------------------------|--------|
| I. 서론 | IV. 실험 |
| II. 관련기술 및 기존 시스템 분석 | V. 결론 |
| III. 제안된 RDF 데이터 저장 및 검색 방법 | |

초록

시맨틱 웹상에서는 정보 자원들이 서로 의미적으로 연결되어 있으므로 컴퓨터가 이를 처리할 수 있다. RDF(Resource Description Framework)는 이러한 의미적 연결성을 제공한다. 시맨틱 웹이 발전하기 위해서는 RDF 데이터를 효율적으로 관리할 수 있는 방법이 매우 중요하다. 이에 따라 본 연구에서는 RDF 데이터를 XML 데이터베이스 시스템에 저장하고 이를 검색하는 기법을 제안하였다. XML 데이터베이스 시스템을 사용함으로써 XML 데이터와 RDF 데이터를 통합적이고 효율적으로 관리할 수 있다. 본 연구에서 제안한 저장 및 검색기법을 기반으로 새로운 시스템을 구현하고, 이를 기존 시스템과 비교·평가 하였다. 평가 결과에 의하면 제안한 검색 기법이 기존 시스템 보다 성능이 향상되었음을 확인할 수 있었다.

주제어: 시맨틱 웹, 자원기술모형

ABSTRACT

The semantic web was proposed as the next generation web technology. In the environment of the semantic web, resources as well defined and related with each other semantically, the RDF supports this basic mechanism. To establish and develop the semantic web, the basic technologies related to RDF data must be pre-established. In this research, we develop methods for storing and querying RDF data using an XML database system. As using XML database, XML data and RDF data can be integrated and efficiently managed. We construct and evaluate a system applying the proposed method to store and search data, we compared the query processing performance on our system with that of an existing system. The experiment result show that our system processes queries more efficiently.

Key Words: XML, RDF, RQL, XPath,

* 국방부 합동참모본부(nam0181@hanmail.net)(제1저자)

** KT BcN 본부(yongkim@kt.co.kr)(공동저자)

• 접수일: 2006년 8월 28일 • 최초심사일: 2006년 9월 8일 • 최종심사일: 2006년 9월 21일

I. 서론

인터넷과 정보 기술의 발전에 따른 정보의 폭발적인 증가와 정보 형태의 다양화는 도서관에서 정보자원의 기술 및 접근 방법에 많은 변화를 요구하고 있다. 네트워크상의 자원은 인쇄자료와는 달리 그 성격이 동적이기 때문에 전통적인 목록과 같은 방법을 통하여 해당 정보나 주제에 접근이 상대적으로 어렵다. 이러한 문제점을 해결하기 위해 네트워크 환경에서 디지털 정보에 대한 기술과 접근을 용이하게 하는 수단으로서 메타데이터 개념이 적용되고 있다. 그러나 상이한 형식의 난립으로 메타데이터 상호간의 호환성 문제에 직면하게 되어 정보의 공유를 달성하는데 저해요인이 되고 있다.¹⁾

이에 따라 상이한 메타데이터를 효율적으로 통합하고 이용할 수 있도록 하기 위해 어의나 구문 및 구조 등의 측면을 지원하는 메타데이터 관련 요소들의 공통적 규정이 있어야 하는데, RDF(resource description framework)는 이를 지원하기 위해 새롭게 개발된 기술 구조이다. 이러한 RDF는 다양한 메타데이터 형식과의 연계를 통한 통합기술모형이라고 할 수 있다.²⁾

최근 차세대 웹으로 시멘틱 웹(semantic web)이 부각되고 있다. 기존의 월드 와이드 웹과는 달리 시멘틱 웹상에서는 정보 자원들의 의미가 정의되어 있고, 이들 간의 의미적 연결을 지원한다. 시멘틱 웹에서 이런 의미적 연결성을 지원하기 위해 RDF는 보다 효과적인 도구로서 이용되고 있다. 이러한 시멘틱 웹이 기존의 월드 와이드 웹을 대체하는 새로운 차세대 웹으로 자리 잡기 위해서는 RDF 데이터를 효율적으로 처리할 수 있는 기술 및 관련 제반 기술들에 대한 연구가 필요하다³⁾. 특히, 방대한 양의 RDF 데이터를 효율적으로 저장하고 검색하는 기법은 폭발적으로 증가하는 정보의 처리와 제공에 있어서 매우 중요하다. RDF에서 채택하고 있는 표현 구문은 어의의 일반적인 표현을 위하여 XML을 사용하고 있는데, XML의 문헌유형 선언부는 사용하지 않고, XML의 정형화된 규정을 이용하고 있다. 본 연구에서는 이러한 관계성을 기반으로 RDF 데이터 및 RDF 스키마 데이터(RDF schema data)를 XML 데이터베이스에 저장하고, RDF 질의 언어인 RQL을 이용하여 검색하는 방법을 모색하였다. 기존의 연구와 달리 RDF 데이터를 XML 데이터베이스에 직접 저장하여 XML 데이터와 이의 메타데이터인 RDF 데이터를 같은 위치에서 저장 및 관리를 수행함으로써 해당 데이터를 통합적이고 효율적으로 관리할 수 있다. 또한 RDF 데이터를 XML 형태로 표현할 수 있기 때문에 XML 데이터베이스에 RDF 데이터를 별도의 처리과정 없이 XML 형태 그대로 저장할 수 있다.

-
- 1) 김이경, 김태수. "메타데이터 연계성을 위한 RDF 응용스키마 설계에 관한 연구," 한국정보관리학회지, 제17권, 제1호(2000. 3), pp.21-47.
 - 2) Manola, Frank and Eric Miller. RDF Primer, 2004.
<<http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>> [cited 2006. 8.23].
 - 3) McIlraith, Sheila A., Tran Cao Son, Honglei Zeng. "Semantic Web Services." IEEE INTELLIGENT SYSTEMS, Vol 16, No. 2(Mar/Apr 2001), pp.46-53.

그러나 해당 데이터에 대한 검색에 있어서 RDF 데이터와 XML 데이터의 모델이 다르기 때문에 XML로 표현된 RDF 데이터를 XML 질의 언어인 XPath 질의 언어로 검색하는 것은 매우 비효율적이라고 할 수 있다. RDF 데이터는 실제 XML 데이터베이스 시스템에 저장되어 있으므로 RQL 질의가 입력되면, 이를 XML 질의 언어인 XPath로 변환하는 작업이 필요하다.

따라서 본 연구에서는 RDF 데이터의 저장 및 해당 데이터의 검색시 RQL 질의를 XML 질의 언어인 XPath 질의로 변환하여 처리할 수 있는 효율적인 방법을 제안하였다. 본 연구에서 제안한 저장 및 검색기법을 기반으로 시스템을 구축하고, 이를 기존 시스템과 비교·평가하였다.

II. 관련 기술 및 기존 시스템 분석

1. 자원기술모형(RDF)

시멘틱 웹의 궁극적인 목적은 웹에 있는 정보를 컴퓨터가 좀 더 잘 이해할 수 있도록 도와 주는 표준과 기술을 개발하여 시멘틱 검색, 데이터 통합, 작업의 자동화 등을 지원하는 것이다. 세부적으로는 정보를 검색할 때 더욱 정확한 결과를 가져오고 서로 다른 이질적인 정보를 통합하고 비교한다. 이를 위하여 요구되는 기술요소들이 존재한다. 대표적으로 웹상의 정보자원을 기술하는데 메타데이터가 요구되는데, 메타데이터는 데이터에 관한 데이터라고 정의할 수 있다. 이러한 메타데이터는 웹에 있는 방대한 자원들을 종류와 특성에 따라 간단하게 기술하며 원하는 자원에 쉽게 접근할 수 있고 정확한 검색에 유용하게 쓰이고 있다. 메타데이터의 핵심은 간결성이다. 현재 대표적인 메타데이터로는 더블린 코어(dublin core)가 있다. 더블린 코어형식은 네트워크에 있는 자원의 기술에 필요한 요소를 정하고 쉽고 정확한 검색을 목적으로 1995년에 합의된 메타데이터로서 15개의 핵심 기술요소로 이루어져 있다.⁴⁾ 이들 기술요소들은 자원의 내용, 다른 자원과의 관계, 자원의 속성, 형식, 시간 등을 나타낸다. 더블린코어를 기초로 다른 분야에서도 확장하여 사용하는 것이 가능하다. 메타데이터의 목적은 정보검색의 처리과정을 줄여 주고 원하지 않는 데이터를 걸러주어 관련성이 있는 정보의 발견 가능성을 증가시켜 검색능력을 향상시키는 데에 있다.

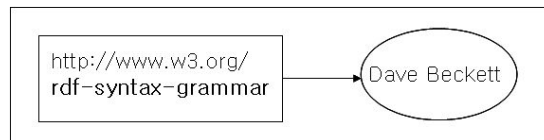
한편, RDF는 메타데이터를 처리하기 위한 기초로서, 현재 W3C의 표준으로 정해져 있으면서 기계가 이해할 수 있는 정보를 교환하는 응용프로그램 간의 상호 운용성을 제공하며 자원의 자동적인 처리를 하는데 편리하다. 특히, 자원검색에 있어서는 개선된 검색능력을 제공하고 목록분야에서는 콘텐츠나 콘텐츠 간의 관계를 기술할 수 있다. RDF는 자원의 속성과 값을 XML의 방식으로

4) 조윤희, 이두영. "RDF기반 인터넷 자원 메타데이터 설계에 관한 연구." 한국정보관리학회지, 제17권, 제3호 (2000. 9), pp.147-170.

표현하고 데이터에 의미를 부여한다. 이러한 RDF가 적용될 수 있는 응용분야는 매우 다양하다.

기본적인 RDF 데이터 모델은 자원, 속성, 선언문으로 구성되어 있으며, 객체중심이 아닌 속성중심의 구조를 가지고 있다. 자원은 속성을 가지며 웹에서 URI(uniform resource identifiers)로 식별되는 모든 객체를 가리킨다. 웹 페이지의 전체나 일부가 될 수도 있고 웹 페이지들의 집합을 나타낼 수도 있다. 속성은 자원의 기술을 위해 사용된 특징, 관계를 말한다. 선언문은 자원과 특성 그리고 값을 말하며 각각 주부, 술부, 객체라고 한다. 객체는 다른 자원일 수도 있고 문자열을 가리키기도 한다.

〈그림 1〉은 간단한 RDF 데이터 모델로서, 이 모델의 의미는 “Dave Beckett는 자원 〈http://www.w3.org/RDF-syntax-grammar〉의 작성자이다.”라는 뜻이다.



〈그림 1〉 기본적인 RDF 데이터 모델

〈표 1〉은 〈그림 1〉의 RDF 데이터 모델을 주부, 술부, 객체로 나눈 것이며, 이것은 각각 자원, 속성, 선언문에 대응된다.

〈표 1〉 RDF 데이터 모델의 주부, 술부, 객체

Subject	http://www.w3.org/RDF-syntax-grammar
Predicate	Author
Object	"Dave Beckett"

〈그림 2〉는 〈표 1〉의 데이터 모델을 RDF/XML 구문으로 표현한 것이다.

```

<RDF:RDF>
<RDF:Description
  RDF:about="http://www.w3.org/RDF-syntax-grammar">
  <ex:author>Dave Beckett<ex:author>
</RDF:Description>
<rdf:Description
  rdf: about="http://www.w3.org/rdf-syntax-grammar">
  <dc:title>RDF/XML Syntax Specification(revised)
  </dc:title>
</RDF:Description>
</RDF:RDF>
    
```

〈그림 2〉 RDF/XML 구문 표현

〈그림 3〉은 RDF/XML 구문에 대해 단축 구문 방법을 적용하여 표현한 것이다. 이와 같은 표현 방법은 본 논문에서 제안하고 있는 저장 및 검색방법에 적용하였다.

```
<RDF:RDF>
<RDF:Description
  rdf: about="http://www.w3.org/rdf-syntax-grammar">
  ex:author>Dave Beckett<ex:author
  dc:title>RDF/XML Syntax Specification(revised)
</RDF:Description>
</RDF:RDF>
```

〈그림 3〉 RDF/XML 구문 표현에 대한 단축 구문

2. RDF 스키마

RDF 스키마는 클래스(class)와 하위 클래스(subclass), 자원(resource)의 개념을 나타내고 있다.⁵⁾ 클래스는 사각형으로 표시되며, 자원은 여러 개의 점으로 표시된다. 화살표는 각 계층에서 나타내는 자원을 표시하는데 사용되며, 하위 클래스는 상위 계층 안에 포함되는 구조를 이룬다.

RDF 스키마는 특정 RDF에서 어의적인 속성 유형의 집합이라 할 수 있는 어휘집을 선언하는데 사용된다. RDF 스키마는 속성 유형의 값의 특성이나 제한 뿐만 아니라 주어진 RDF 기술에서의 유용한 속성들을 정의한다. 즉, 속성의 특징이나 그에 상응하는 속성값을 규정한다. RDF 스키마를 식별하기 위해 XML namespace 메커니즘이 사용되며, URI를 이용하여 RDF 스키마를 인간과 기계가 동시에 처리할 수 있도록 기술한다.⁶⁾

기본적인 스키마의 유형은 ‘Property’, ‘Property Type’, ‘Instance Of’, ‘SubType Of’, ‘Domain’, ‘Range’ 등이 있으며 이러한 스키마의 유형은 RDF에서 정의한 구문과 데이터 모형으로 표현이 가능하다.

3. RQL(RDF Query Language)과 XPath

현재 XML 데이터에 대한 질의 언어로서 XQuery⁷⁾ 질의 언어가 표준으로 제정되어 있다. 즉, XQuery 질의 언어의 기본이 되는 XML 데이터 경로를 탐색하기 위한 질의 언어가 XPath 질의

5) Brickley, D and R. V. Guha. 2004. RDF Vocabulary Description Language 1.0: RDF Schema, 2004. <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/> [cited 2006. 08. 21].

6) Beckett, D. 2002 RDF/XML Syntax Specification (Revised), 2004. <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/> [cited 2006. 07. 28].

7) Boag, Scott, Dan Chamberlin, Mary F. Fernández, Daniela Florescu, Jonathan Robie, and Jérôme Siméon. 2006. XQuery 1.0: An XML Query Language, 2006. <http://www.w3.org/TR/2006/CR-xquery-20060608/> [cited 2006. 08. 17].

언어이다.⁸⁾ XPath를 통해 XML 데이터의 최상위 노드(root node) 또는 임의의 노드로부터 특정 경로에 해당하는 노드와 특정 조건을 만족하는 경로의 노드를 탐색할 수 있다.

일반적으로 RDF 데이터는 표준화된 XML로 표현할 수 있으므로 XPath 질의 언어를 이용하여 RDF 데이터에 대한 검색은 가능하다. 그러나 이러한 검색기법은 RDF 데이터 모델의 특성을 살리지 못할 뿐만 아니라 비효율적이라고 할 수 있다.

RDF 질의 언어는 현재까지 표준으로 제정된 것은 없으며 여러 가지의 다양한 RDF 질의 언어가 표준언어로서 제안되어 있는 상태이다. 본 연구에서 적용한 RDF 질의 언어는 RQL 질의 언어⁹⁾로서 RDF 데이터를 검색하는데 있어서 가장 일반적으로 이용되는 질의 언어라고 할 수 있다.

4. RDF 데이터 저장 및 검색 시스템

현재 시멘틱 웹에 대한 연구 분야로서 RDF 데이터 저장 및 검색에 대한 연구들이 다양하게 진행되고 있으나 아직은 매우 미미한 상황이라고 할 수 있다. 이러한 연구들 중에 대표적으로 RQL 질의 언어를 지원하는 것으로서 ICS-FORTH의 RDFSuite¹⁰⁾와 On-to-Knowledge 프로젝트의 일부인 Sesame¹¹⁾ 등이 있다.

RDFSuite 시스템은 모두 RDF 데이터를 객체 지향형 데이터베이스 시스템(ORDBMS)에 RDF 스키마 정보를 이용하여 데이터베이스 스키마를 구성하고 이에 맞게 데이터를 저장한다. 즉, 저장 시에는 RDF 스키마 정보를 이용하여 데이터베이스 스키마를 구성하고 스키마에 맞게 RDF 데이터를 조작하여 저장하며, 저장된 데이터에 대한 검색은 RQL 질의가 들어오면 이를 SQL로 변환하여 결과를 가져온다. 이러한 방법은 질의 처리의 대부분이 해당 질의 처리 시스템에서 직접적으로 이루어지도록 하는 방법이다. 또한 해당 시스템은 객체관계형 데이터베이스를 사용하고 있기 때문에 저장시에 스키마에 맞게 데이터를 구성하여 저장해야 하며 검색과정에서는 생성된 테이블 간의 조인연산(join)의 수행을 통하여 결과를 추출해 낸다.

8) Clark, James and Steve DeRose. XML Path Language (XPath) Version 1.0. 1999.
<<http://www.w3.org/TR/1999/REC-xpath-19991116>> [cited 2006. 08. 11].

9) Karvounarakis, Gregory, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL: A Declarative Query Language for RDF. The proceeding of 11th International World Wide Web Conference, 2002.

10) Alexaki, Sofia, Vassilis Christophides, Gregory Karvounarakis, Dimitris Plexousakis, Karsten Tolle. "The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases." The Proceeding of 2nd International Workshop on the Semantic Web (SemWeb '01), in conjunction with Tenth International World Wide Web Conference(2001), pp.1 -13.

11) Broekstra, Jeen, Arjohn Kampman, and Frank van Harmelen. "Sesame: An Architecture for storing and querying RDF Data and Schema Information". In Semantics for the WWW, edited by D. Fensel, J. Hendler, H. L., and Wahlster, W. Cambridge, Mass: MIT Press, 2001.

한편, Sesame 시스템은 RDFSuite 시스템과는 달리 특정 데이터베이스에만 적용되지 않고 다양한 데이터베이스에 적용되도록 고안되어져 있다. 따라서 Sesame 시스템은 이전의 RDFSuite 시스템에 비하여 보다 범용적으로 사용할 수 있는 장점이 있다.

5. 기존 방법의 문제점

RDF 데이터의 저장 및 검색에 대한 대표적인 RDFSuite 시스템과 Sesame 시스템에서 적용하고 있는 방법에 대하여 간단히 알아보았다. 그러나 이 두 가지 방법은 실제 적용에 있어서 일부 문제점을 지니고 있다고 할 수 있다. RDFSuite 시스템에서는 저장시에 RDF 스키마 정보를 이용하여 데이터베이스 스키마를 구성하고 스키마에 맞게 RDF 데이터를 저장한다. 이러한 방식은 질의 처리 작업의 대부분이 질의 처리 시스템에서 이루어지기 때문에 데이터베이스 시스템에 최적화된 질의 처리기를 사용하는 다른 시스템에 비해 상대적으로 검색 성능이 떨어진다. 또한 해당 시스템은 객체관계형 데이터베이스를 사용하고 있으므로 저장시 데이터베이스 스키마에 맞게 데이터를 재구성하여 저장해야 한다. 검색과정에서는 여러 테이블 간의 조인연산을 수행함으로써 해당 작업에 따른 오버헤드가 존재한다.

한편, Sesame 시스템은 RDFSuite 시스템과는 달리 특정 데이터베이스에 국한되지 않는 장점이 있으나 각각의 개별적인 데이터베이스에 대한 최적화된 질의 처리기를 사용할 수 없기 때문에 데이터의 검색에 있어서 성능의 저하를 가져올 수 있다.

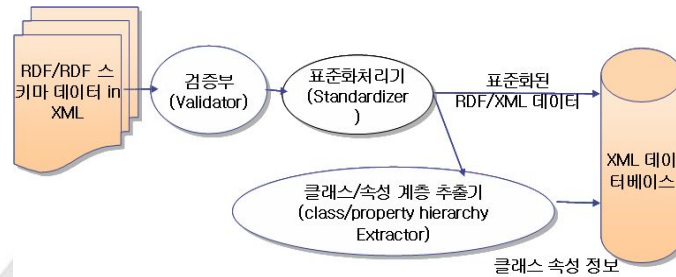
따라서 본 연구에서는 기존 시스템에 대한 문제점을 해결하기 위하여 새로운 두 가지 방법을 제시하였다. 첫 번째, XML 데이터베이스 시스템을 사용하여 데이터의 저장시에 별도의 처리 없이 데이터를 그대로 저장하면서 XML 데이터와 이의 메타 데이터인 RDF 데이터를 같은 시스템에 저장하고 관리함으로써 두 데이터 간의 공간적 차이를 줄여 통합적이고 효율적인 관리가 가능하도록 하였다. 두 번째, RDF 데이터에 대한 질의 처리를 가능한 동일한 데이터베이스에서 수행하도록 하였다.

Ⅲ. 제안된 RDF 데이터 저장 및 검색 방법

1. 데이터 저장 방법

본 연구에서 제안한 RDF 데이터에 대한 저장 과정은 <그림 4>와 같다. 먼저, RDF 기술 데이터와 RDF 스키마 데이터가 입력이 되면 해당 데이터를 검증부(validator)에서 구문과 의미에 대한

검증을 수행한다. 이와 같은 검증과정이 마무리 되면 표준화 처리기(standardizer)에서 여러 형태의 RDF/XML 구문을 저장 공간의 효율성과 검색의 용이성을 위해 하나의 구문으로 통합하는 작업을 수행한다. 이렇게 통일된 형태의 표준화된 RDF/XML 데이터가 XML 데이터베이스에 저장된다. 이와 병행하여 클래스/속성 계층 추출기(class/property hierarchy extractor)에서 RDF 스키마 데이터의 클래스와 속성에 대한 계층 구조 및 속성의 도메인, 범위 제약사항에 대한 정보를 추출한다. 추출된 정보는 별도의 XML 문서로 저장한다. 별도로 저장된 정보들은 검색의 성능을 향상시키기 위해 활용된다.



〈그림 4〉 RDF 데이터 저장 과정

2. 데이터 통합 및 변환

RDF 기술 데이터 및 RDF 스키마 데이터는 RDF/XML 구문을 이용하여 여러 형태의 XML 데이터로 표현될 수 있다. 또한, XML 질의 언어인 XPath 질의는 XML 구문에 의존적이기 때문에 입력된 RDF/XML 데이터를 통일된 형태의 RDF/XML 데이터로 변환할 필요가 있다. 본 연구에서는 검색의 용이성과 저장의 효율성을 위해 RDF/XML 데이터의 여러 단축(abbreviation) 구문 중에서 두 가지 구문만을 적용한 형태로 모든 RDF 데이터를 통일하였다. 즉, 기존의 RDF/XML 구문 보다 단축된 형태로서 저장에 따른 공간의 효율성 및 자원에 대한 속성과 속성값을 통일된 형태로 표현하였으며, 또한 데이터에 대한 검색은 XPath 질의로 변환이 가능하도록 구현하였다. 예를 들어 〈그림 2〉와 〈그림 3〉은 같은 RDF 데이터를 XML 형태로 표현한 것이다. 이때 'http://www.w3.org/rdf-syntax-grammar' 자원의 'title' 속성의 속성값을 알고 싶은 경우에 〈그림 2〉의 XML 데이터에는 다음과 같은 XPath 질의로 검색할 수 있다.

/rdf:Description[rdf:about = http://www.w3.org/rdf-syntax-grammar]/dc:title

〈그림 3〉의 XML 데이터에서 같은 결과를 얻기 위해서는 다음과 같은 XPath 질의로 검색이

가능하다.

`/rdf:Description[rdf:about=http://www.w3.org/rdf-syntax-grammar]@dc:title`

〈그림 2〉의 XML 데이터의 경우에는 'title' 속성이 기술(description) 엘리먼트의 자식노드(child node)로 표현되었으나, 〈그림 3〉에서는 어트리뷰트(attribute)로 표현되어 있으므로 같은 결과를 얻기 위해서는 XML 표현형태에 따라 다른 질의가 필요하다. 따라서 RDF 데이터에 대한 검색에 있어서 입력된 RDF/XML 구문을 통한 데이터의 형태와 관계없이 RQL 질의를 일정한 XPath 질의로 변환하기 위해서는 하나의 공통된 형태의 RDF/XML 데이터로의 변환이 요구된다.

RQL은 RDF 스키마 클래스, 속성에 대한 도메인, 범위 등을 제한할 수 있는 질의가 가능하다. 본 연구에서는 클래스, 속성의 계층 구조, 속성의 도메인, 범위 정보 등을 추출하고 XML 형태로 데이터베이스에 저장하여 해당 정보의 질의를 간단한 XPath 질의로 수행하도록 하였다. 〈그림 5〉은 RDF/XML 형태의 RDF 스키마 데이터의 예를 제시하고 있다.

```

<rdf:Description rdf.ID="MotorVehicle">
  <rdf.type rdf.resource=http://www.w3.org/2000/01/rdf-schema#Class/>
  <rdfs:subClassOf rdf.resource=http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdf:Description>
<rdf:Description rdf.ID="PassengerVehicle">
  <rdf.type rdf.resource=http://www.w3.org/2000/01/rdf-schema#Class/>
  <rdfs:subClassOf rdf.resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf.ID="Van">
  <rdf.type rdf.resource=http://www.w3.org/2000/01/rdf-schema#Class/>
  <rdfs:subClassOf rdf.resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf.ID="miniVan">
  <rdf.type rdf.resource=http://www.w3.org/2000/01/rdf-schema#Class/>
  <rdfs:subClassOf rdf.resource="#Van"/>
  <rdfs:subClassOf rdf.resource="#PassengerVehicle"/>
</rdf:Description>
    
```

〈그림 5〉 RDF/XML 형태의 RDF 스키마 데이터의 예

〈그림 5〉에서의 RDF 스키마 데이터에 대한 모든 하위 클래스를 추출하기 위해 RQL 질의를 XPath 질의로 변환한 결과를 〈그림 6〉에서 보여 주고 있다.

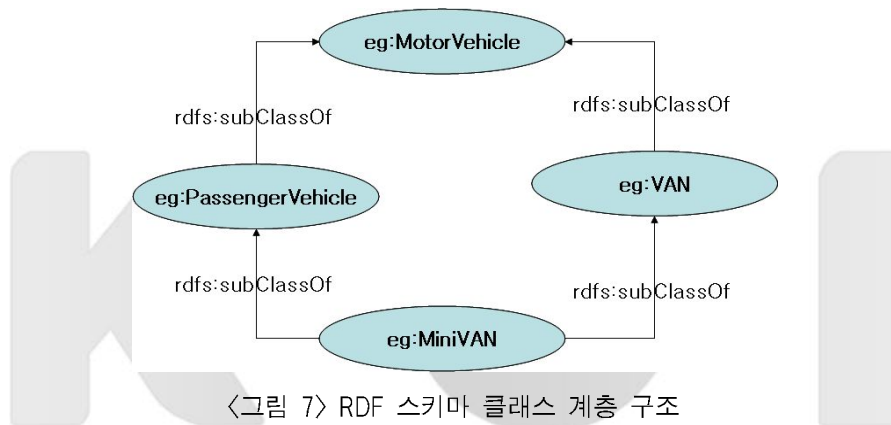
```

R= {/rdf:Description[/rdfs:subClassOf/@rdfs:resource="MotorVehicle"]@rdf:ID}
∃ r ∈ R, /rdf:Description[rdf:ID = r]/rdfs:subClassOf/@rdfs:resource
    
```

〈그림 6〉 RQL 질의에 대한 XPath 질의로의 변환의 예

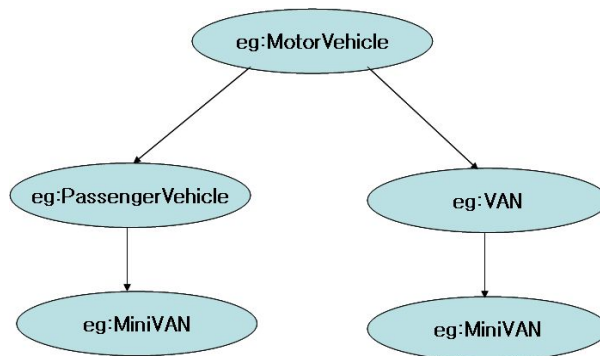
〈그림 6〉에서 보는 바와 같이 MotorVehicle의 하위 클래스를 구하고, 각각의 하위 클래스의 하위 클래스를 검색하는 재귀적 과정을 통하여 결과를 추출할 수 있다. 이로 인해 클래스의 계층 관계가 깊어질수록 더 많은 XPath 질의가 필요하게 된다

〈그림 7〉은 추출된 RDF 스키마 클래스 계층구조를 보여 주고 있다, 이러한 과정은 속성에 대한 계층관계를 추출하는데 있어서 동일하게 적용된다. 따라서 본 연구에서 제안하고 있는 방법과 같이 클래스 및 속성의 계층관계 데이터를 미리 추출하고 이를 데이터베이스에서 관리함으로써 질의에 대한 검색 속도를 향상 시킬 수 있다. 뿐만 아니라 RDF 스키마 데이터의 양이 많은 경우에는 도메인(domain) 및 범위(range) 속성 데이터를 미리 추출하여 관리함으로써 검색 속도를 개선시킬 수 있다.



〈그림 7〉 RDF 스키마 클래스 계층 구조

〈그림 8〉은 〈그림 5〉의 RDF 스키마에서 클래스 계층구조를 추출하여 XML의 부모 엘리먼트 (parent element)와 자식 엘리먼트(child element)의 관계를 표현한 것이며, 이를 텍스트 형태로 표현하면 〈그림 9〉와 같다.



〈그림 8〉 클래스 계층구조에 대한 그래프형태의 XML 데이터

```

<eg:MotorVehicle>
  <eg:PassengerVehicle>
    <eg:MiniVan>
  </eg:PassengerVehicle>
  <eg:Van>
    <eg:MiniVan/>
  </eg:Van>
</eg:MotorVehicle>

```

〈그림 9〉 RDF 스키마 클래스 계층구조에 대한 텍스트형태의 XML 데이터

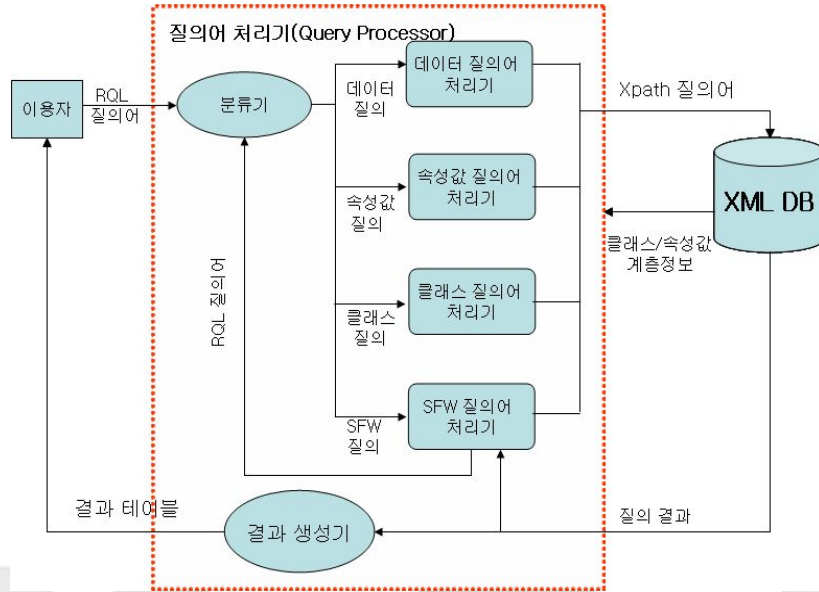
RDF 데이터 모델은 노드와 간선에 레이블이 할당된 그래프의 형태로서, 노드에만 레이블이 있는 트리형태의 XML 데이터 모델과는 다른 형태를 취하고 있다. 따라서 RDF 스키마 데이터 중에서 클래스 및 속성에 대한 하위 클래스(subClassOf) 속성으로 연결된 데이터만을 추출하여 간선의 레이블을 없애고, 이를 상위 클래스에 대한 하위 클래스로 각각 분리한다. 그리고 해당 데이터를 XML 데이터 형식으로 표현하여 RQL 질의를 간단한 하나의 XPath 질의로 변환시켜 그 결과를 추출한다. 이와 같은 방법을 통하여 속성에 대한 계층구조, 도메인, 범위의 제한사항 정보도 동일한 방법으로 XML 데이터로 변환하여 저장함으로써 RQL 질의를 보다 효율적으로 처리할 수 있다.

이 방법은 기존 RDF 스키마 데이터에 대한 XPath 질의 보다 훨씬 간단하게 질의가 수행됨으로 해당 데이터에 대한 검색성능을 높일 수 있다. 또한 속성에 대한 계층 구조를 표현하는데 있어서도 동일한 방법을 적용하여 보다 효율적인 저장과 검색이 이루어 질 수 있다.

3. 저장 데이터 검색 방법

가. 질의 처리 과정

본 연구에서 제안한 RQL 질의를 XPath로 변환하여 질의를 처리하는 전체 과정은 〈그림 10〉과 같다. 먼저 이용자로부터 RQL 질의가 입력되면, 이를 분류기(parser)에서 분류(parsing)한 후 질의의 분류에 따라 해당 질의 처리기가 RQL 질의를 XPath 질의로 변환하여 XML 데이터베이스에 보내어 결과를 가져 온다. 그리고 결과 생성기가 결과를 테이블 형태로 변환하여 사용자에게 보내 주게 된다. 특히 해당 시스템에서는 질의를 처리하는 과정에서 RDF 데이터를 저장할 때 개별적으로 추출되어 데이터베이스에 저장된 RDF 스키마 클래스와 속성에 대한 계층 정보를 참조하여 질의를 효율적으로 처리한다.



〈그림 10〉 RQL 질의의 XPath로의 변환을 통한 질의 처리 과정

이러한 RQL 질의는 형태상 크게 Non SFW(select-from-where) 질의와 SFW(select-from-where) 질의로 분류할 수 있다.

나. Non SFW 질의어 변환

〈표 2〉는 Non-SFW query의 분류에 대해 각각에 해당되는 RQL 질의의 예와 XPath 질의로 변환한 예를 보여 주고 있다. 각각의 질의식에서 위 부분은 질의의 예이고 아래 부분은 이를 XPath 질의로 변환한 형태이다. 변환된 질의에서 rdf_data.xml은 RDF 데이터가 저장되어 있는 XML 문서의 이름을 의미하고 있으며, classHierarchy.xml은 클래스 계층 구조 정보가 저장되어 있는 XML 문서의 이름이고, propertyHierarchy.xml은 속성 계층 구조가 저장되어 있는 XML 문서의 이름을 가리킨다. 이러한 Non-SFW 질의는 데이터의 저장단계에서 미리 추출된 정보를 이용하여 간단한 XPath 질의로 변환할 수 있으므로 저장 및 검색의 성능을 향상시킬 수 있다.

〈표 2〉 Non-SFW Query의 XPath로의 변환

클래스 질의 (Class Query)	TypeOf	<code>typeof(www.culture.net #picasso132)</code> <code>document("rdf_data.xml")//rdf:Description[@rdf:about = " www.culture.net #picasso132"]/rdf:type/@rdf:resource</code>
	SuperClassOf	<code>SuperClassOf(http://www.icom.com/schema1.rdf #Painter)</code> <code>document(" classHierarchy.xml")//NS:Painter/ancestor::*</code>
	Domain	<code>domain(http://www.icom.com/schema1.rdf #creates)</code> <code>document(" propertyHierarchy.xml")//NS:creates /@rdf:domain</code>
	URI	<code>http://www.w3.org/2000/01/rdf-schema #Class</code> <code>document(" classHierarchy.xml")//*</code>
속성 질의 (Property Query)	SuperPropertyOf	<code>SuperPropertyOf(http://www.icom.com/schema1.rdf #sculpts)</code> <code>document(" propertyHierarchy.xml")//NS:sculpts/ancestor::*</code>
	SubPropertyOf	<code>SubPropertyOf(http://www.icom.com/schema1.rdf #creates)</code> <code>document(" propertyHierarchy.xml")//NS:creates/descendant::*</code>
	URI	<code>http://www.w3.org/1999/02/22-rdf-syntax-ns #Property</code> <code>document(" propertyHierarchy.xml")//*</code>
	InstanceOf	<code>^http://www.icom.com/schema1.rdf #Painter</code> <code>document(" rdf_data.xml")//rdf:Description[rdf:type/@rdf:resource = " http://www.icom.com/schema1.rdf #Painter"]/@rdf:about</code>
데이터 질의 (Data Query)	URI	<code>http:// www.icom.com/schema1.rdf #Artist</code> <code>cSubClassOf(http://www.icom.com/schema1.rdf #Artist),</code> <code>document(" rdf_data.xml")//rdf:Description[rdf:type/@rdf:resource = c]/@rdf:about</code>

다. SFW 질의 처리 방법

RDF 기술 데이터나 RDF 스키마 데이터를 검색하는데 있어서 SFW 질의 방법은 매우 유용하다고 할 수 있다. 특히 SFW 질의 방법은 RDF 기술 데이터와 RDF 스키마 데이터를 동시에 검색하는 기능을 제공한다. SFW 질의 방법은 RDF 데이터 모델을 N-Triples 형태의 검색방법을 제공한다. SFW 질의 처리 과정을 간단하게 살펴보면, 우선 SFW 질의를 select, from, where 부분으로 분리한 다음 from절의 경로 표현식을 단순 경로식들로 분리한다. 그 다음으로 where절에 있는 조건(condition)을 and로 연결된 형태(conjunctive form)로 바꾼 후, and를 기준으로 분리한다. 이어서 from절의 각각의 경로식을 XPath 질의로 변경한다. 이때, where절의 조건 중에서 같이 처리할 수 있는 것들은 XPath 질의의 프리디캣(predicate)으로 전달하여 XPath 질의에 대한 결과를 가져올 경우 조건에 맞는 데이터만을 가져오게 한다.

〈그림 11〉은 RQL SFW의 질의식에 대한 간단한 예를 보여 주고 있다.

```
select X, $W, Y, Z, $V, U
from {X: $W} creates {Y}, exhibited {Z: $V},
      {X}fname{U}
wherenot( $W <= singeror Z like )
```

〈그림 11〉 RQL SFW 질의식의 예

〈그림 11〉의 질의식에서 From 절의 경로 표현식 X, Y 등의 변수는 RDF 기술 데이터 변수이며, '\$' 표식이 붙은 '\$W' 등의 변수는 RDF 스키마 클래스 변수이다. From 절의 첫 번째 경로 표현식 'from {X:\$W} creates {Y}'는 속성이 'creates'인 트리플(triple)의 서브젝트 자원 'X'와 해당 자원이 포함된 클래스인 '\$W', 그리고 오브젝트(object) 'Y'를 의미한다. 또한 두개의 경로를 ';'로 연결하여 조인연산을 표현하고 있다. Where 절에는 변수에 대한 조건을 명시함으로써 만족하는 데이터만을 검색할 수 있다.

(1) 경로 표현식의 분류

〈표 3〉은 본 연구에서 제안하고 있는 단순 경로 표현식에 대한 분류와 각각에 해당되는 XPath 질의를 보여 주고 있다.

단순 경로 표현식은 RDF 기술 데이터의 경로를 표현하는 데이터 경로(data path), RDF 스키마 데이터의 경로를 표현하는 스키마 경로(schema path) 그리고 두 가지의 경로 모두를 표현하는 혼합 경로(mixed path)가 있다. 경로 표현식에서 'c'는 질의에서 주어진 RDF 스키마 데이터의 특정 클래스를 의미하며, 'p'는 RDF 스키마 데이터의 특정 속성을 나타내고, 'Class'는 RDF 스키마에 정의된 속성(property) 클래스를 의미한다. 변환된 XPath 질의에서 'C'는 RDF 스키마 데이터의 모든 클래스의 집합을, 'P'는 모든 속성들의 집합을 나타낸다.

〈표 3〉 RQL SFW 질의의 경로 표현식 분류 및 해당 XPath 질의

유형	경로 표현식	변환된 XPath 질의
데이터 경로 (Data path)	c{X}	c'SubClassOf(c), <rdf:Description[type/@resource = c']/@about>
	\$X{Y}	cC, <c, //rdf:Description[rdf:type/@rdf:resource = c]/@rdf:about>
	{X}p{Y}	p'SubPropertyOf(p), V1={/rdf:Description[p']/@rdf:about}, v1V1, <v1, //rdf:Description[@rdf:about = v1]/p'/@rdf:resource>
	{X}@P{Y}	pP, V1={/rdf:Description[p']/@rdf:about}, v1V1, <v1, p, //rdf:Description[@rdf:about = v1]/p/@rdf:resource>
스키마 (Schema)	Class{X}	cC, <c >
	Property{P}	pP, <p >
	c: { \$C }	c'SubClassOf(c), <c' >
	\$X{ : \$Y }	cC, <c, SubClassOf(c) >
	{ : \$X }p{ : \$Y }	c1SubClassOf(Domain(p)), c2SubClassOf(Range(p)), <c1, c2>
{ : \$X }@P{ : \$Y }	pP, c1SubClassOf(Domain(p)), c2SubClassOf(Range(p)), <c1, p, c2 >	
혼합형식 (Mixed)	c{X: \$C}	c'SubClassOf(c), <rdf:Description[rdf:type/@rdf:resource = c']/@rdf:about, c'>
	{X: \$Z}p{Y: \$W}	p'SubPropertyOf(p), V1={/rdf:Description[p']/@rdf:about}, v1V1, C1={/rdf:Description[@rdf:about = v1]/rdf:type/@rdf:resource}, V2={/rdf:Description[@rdf:about = v1]/p'/@rdf:resource}, v2V2, C2={/rdf:Description[@rdf:about = v2]/rdf:type/@rdf:resource}, c1C1, c1SubClassOf(Domain(p')), c2C2, c2SubClassOf(Range(p')), < v1, c1, v2, c2 >

(2) Where절의 조건(condition) 처리

조건절에 대한 연산은 불리안 연산(boolean), 비교 연산(comparative), 부정 연산(not) 으로 구성된다. <표 4>에서는 조건의 비교 연산과 not연산에 대해 XPath 프리디킷으로 변환이 가능한지 그리고 가능하다면 어떻게 변환할 수 있는지를 보여 주고 있다. 이처럼 가능한 조건을 XPath 질의의 프리디킷으로 표현하여 조건에 대한 연산을 데이터베이스 시스템에서 처리하도록 함으로써, 데이터베이스 시스템의 질의 처리기를 최대한 활용할 수 있고, 데이터베이스에서 주고 받는 데이터의 양을 줄일 수 있어 RQL 질의 처리를 효율적으로 수행하도록 하였다.

<표 4> 조건 연산의 분류에 따른 XPath 질의의 프리디킷으로의 변환

조건 연산자	<, <=, >, >=	=, !=, not	like
XPath 프리디킷	X	=, !=, not	contains()

(3) 조인연산 처리

조인연산 처리는 각각의 경로 표현식의 결과를 순서대로 둘씩 조인을 수행한다. 두개의 결과가 조인이 필요 없는 경우는 두 결과를 프로덕트하고, 조인이 필요한 경우는 공통의 변수에 대해 sort-merge 조인을 수행한다 이러한 방법으로 데이터베이스 시스템에 대한 접근을 최소화 하였고, 각각의 결과에 대한 조인 횟수를 줄여 조인 연산의 오버헤드를 최소화할 수 있다

IV. 실험

본 연구에서 제안한 방법에 대한 시스템의 구현 및 평가는 Windows XP Professional 운영 체제와 Pentium 3.0GHz/800MHz/2MB 환경에서 XML 데이터베이스 eXcelen XIS 3.1, Visual Studio .NET, C++/STL 등을 이용하여 수행 하였다. 그리고 질의 성능 평가는 <표 5>에서 제시한 네 종류의 질의를 사용하였다.

<표 5> 실험에 적용된 RQL 질의

스키마 질의(Q1)		subClassOf(http://168.21.133.74:8080/RDF/KTC/Examples/music.rdfs#pop)
데이터 질의(Q2)		http://168.21.133.74:8080/RDF/KTC/Examples/music.rdfs#singer
SFW 질의	Q3	select X, Y from {X} http://168.21.133.74:8080/RDF/KTC/Examples/finance.rdfs#song {Y}
	Q4	select \$W, \$Z from {: \$W} http://168.21.133.74:8080/RDF/KTC/Examples/finance.rdfs#song {: \$Z}

첫 번째 질의(Q1)는 RDF 스키마 데이터를 검색하는 스키마 질의 중에서 하위 클래스들을 검색하는 질의이고, 두 번째 질의(Q2)는 RDF 기술 데이터를 검색하는 데이터 질의 중에서 주어진 클래스 URI의 인스턴스를 검색하는 질의이다. 세 번째(Q3)와 네 번째 질의(Q4)는 SFW 질의로서, RDF 기술 데이터에 대한 경로를 포함하는 질의와 RDF 스키마 데이터 경로를 포함하는 질의이다. 성능 평가 실험은 크게 자체적인 성능평가 실험과 기존의 유사한 시스템인 RDFSuite 시스템과의 비교·평가를 수행하였다.

1. 질의 처리 성능

본 실험에서는 <표 5>의 질의를 RDF 기술 데이터의 사이즈가 500K, 1M, 2M인 데이터베이스를 대상으로 수행하였다. 각각의 데이터의 RDF 스키마는 동일하다. <표 6>은 실험 결과를 보여 주고 있다.

<표 6> RQL 질의에 대한 처리 시간 (단위: 초)

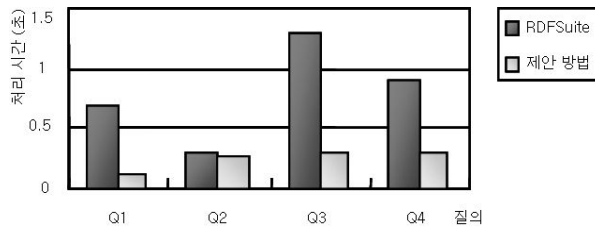
질의	사이즈	500K	1M	2M
Q1		0.137	0.134	0.133
Q2		1.802	2.874	4.447
Q3		44,384	154,863	610,918
Q4		0.438	0.441	0.437

실험 결과에서 Q1과 Q4에 대한 수행 시간은 데이터의 사이즈와 상관없이 일정하게 나타나고 있다. 이는 Q1과 Q4가 RDF 스키마 데이터에 대한 질의이고, 실험에 쓰인 데이터의 스키마 데이터가 일정하기 때문이다. Q2와 Q3의 경우는 RDF description 데이터의 사이즈가 클수록 질의 처리 시간이 길어짐을 볼 수 있다. 특히, Q3은 데이터의 사이즈가 커짐에 따라 처리 시간이 4배 정도씩 길어지고 있음을 알 수 있는데, 이는 RDF description 데이터에 대한 SFW 질의를 XPath 질의로 변환할 때에 여러 XPath 질의들로 변환되고, 이에 따른 조인이 많이 필요하기 때문에 그만큼 시간이 길어진 것이다.

2. RDFSuite 시스템과의 성능 비교

<그림 12>는 동일한 데이터에 대한 질의 처리 시간을 RDFSuite 시스템과 비교한 결과이다. Q2는 RDFSuite 시스템과 질의 처리 시간이 거의 비슷하지만, 다른 질의의 경우에는 본 연구에서 구현한 시스템이 더 나은 성능을 보여 주고 있다. 특히, Q3의 SFW 질의의 경우, 앞 절에서는 데이터

크기가 커짐에 따라 수행 성능이 나빠짐을 알 수 있었지만, 이 시스템과 비교했을 때에는 오히려 성능이 향상되었다. 이것은 본 연구에서 제안한 새로운 시스템이 보다 효율적으로 조인 연산을 수행하고 있기 때문이다.



<그림 12> RDFSuite 시스템과의 성능 비교

V. 결론

본 연구에서는 RDF 기술 데이터와 RDF 스키마 데이터를 XML 데이터베이스에 저장하고, 이를 검색하는 방법을 제안하였다. 시멘틱 웹이 차세대 웹으로 자리 잡기 위해서는 가장 먼저 RDF 기반 기술들이 정립되어야 하며, 이들 기반 기술들 중에서 가장 중요하고 시급한 문제는 RDF 데이터를 저장하고 검색하는 기술이다. 따라서 본 연구에서는 RDF 데이터를 XML 데이터베이스 시스템에 저장함으로써, 시멘틱 웹의 주된 데이터인 XML 데이터와 이에 대한 메타 데이터인 RDF 데이터를 통합적이고 효율적으로 다룰 수 있는 새로운 방법을 제시하였다. 또한, RDF 질의 언어인 RQL 질의 언어를 통해 RDF 기술 데이터 뿐만 아니라 RDF 스키마 데이터도 검색할 수 있도록 하여 RDF 데이터 모델에 맞게 원하는 데이터를 쉽게 검색할 수 있게 하였다. 그리고 저장할 때에 RDF 스키마의 클래스와 속성의 계층 구조를 저장함으로써 보다 쉽고 신속하게 이 데이터에 대한 RQL 질의를 처리하도록 하였다. 검색시에는 최대한 데이터베이스 시스템에서 질의를 처리하도록 하였음은 물론 보다 효율적인 조인 기법을 수행하게 하였다. 이에 따라 본 연구에서 제안한 시스템 성능 평가에서는 기존의 RDFSuite 시스템 보다 질의 처리 성능이 향상되었음을 확인할 수 있었다.

현재 정립되지 않은 RDF 데이터 저장 및 검색 방법을 XML 데이터베이스 시스템을 사용하여 제안함으로써 향후 RDF 데이터를 다루는 연구 분야에 새로운 방향을 제시한 데에 본 연구의 의의가 있다고 할 수 있다. 본 연구에서 제안한 저장 및 검색 기법들은 지속적인 연구를 통해 보다 효율적인 방법으로 발전할 수 있을 것이다.

<참고문헌은 각주로 대신함>

к с і