

# XML의 RDF 변환과 저장 및 질의 처리에 관한 연구

김연희\*, 김병곤\*\*, 이재호\*\*\*, 임해철\*

\*홍익대학교 컴퓨터공학과

\*\*부천대학 e-비즈니스과

\*\*\*경인교육대학교 컴퓨터교육과

{kyh, lim}@cs.hongik.ac.kr, bgkim@bc.ac.kr, jhlee@gin.ac.kr

## The Study of Storing and Query Processing Strategy based on Transition of XML to RDF

Youn Hee Kim\*, Byung Gon Kim\*\*, Jaeho Lee\*\*\*, Hae Chull Lim\*

\*Dept. of Computer Engineering, University of Hong Ik

\*\*Dept. of e-Business, College of Bucheon

\*\*\*Dept. of Computer Education, Gyeongin National University of Education

### 요 약

웹 상의 데이터 표현 및 교환의 표준으로 각광받는 XML은 논리적 구조와 내용 정보를 이용하여 보다 정확한 검색이 가능하다. 그러나 더욱 빠른 속도로 증가하는 많은 양의 데이터에 대해 보다 정확하고 풍부한 검색을 하기 위해서 메타데이터를 활용하는 방법이 고려되었고, RDF와 같은 메타데이터 기술 언어들에 대한 연구가 많이 이루어지고 있다. RDF는 XML의 문법 구조를 이용하여 작성되므로 XML 문서를 RDF 형태로 작성한다면, 약간의 수정을 통해 기존 XML 문서를 RDF 형태로 변환하는 것이 가능하다. XML의 RDF 변환은 여러 이점 때문에 앞으로 활성화될 가능성이 크기 때문에 RDF의 특성을 고려한 저장 및 검색에 대한 연구가 필요하다. 따라서 본 논문에서는 XML을 기본적인 RDF 형태로 변환하는 기본적인 규칙을 소개하고 변환된 RDF 문서를 위한 저장 구조를 제안한다. 제안한 저장 구조는 기존 웹 애플리케이션과의 쉬운 연동을 위하여 관계형 데이터베이스를 기반으로 구성되며, 리소스/속성/값의 RDF 기본 구조를 고려한 세 종류의 테이블로 구성된다. 또한 본 논문에서는 RDF 문서에 대한 키워드 질의 처리를 고려하여, 질의 처리 결과의 단위를 리소스로 정의한다. 그리고 주어진 키워드들에 대한 질의 처리 결과로 반환된 리소스들 간의 중요도를 평가하기 위하여 키워드간의 근접도, 키워드 내포 정도, 다양한 속성 관계를 맺고 있는 다른 리소스들을 고려한 랭킹 평가 기법을 제안한다.

### 1. 서 론

최근 웹 기반 애플리케이션 개발 시 데이터 표현 및 교환의 표준으로서 XML(eXtensible Markup Language)이 각광받고 있기 때문에, 많은 웹 상의 데이터가 XML의 형태를 갖추고 있다. XML은 HTML과는 달리 데이터의 내용과 구조를 모두 표현하는 자기 기술적(self-describing) 특성을 때문에 논리적 구조 및 내용 정보를 이용하여 보다 정확한 검색이 가능하다[1]. 그러나 XML의 논리적 구조는 사용자에게 따라 다양하게 표현될 수 있고, 그 태그 정보 또한 같은 내용에 대해 사용자마다 다른 이름으로 표현할 수 있다. 그러므로 같은 내용의 XML 문서라도 서로 다른 논리적 구조와 의미의 표현을 가지고 있을 수 있기 때문에, 검색 시 서로 다른 문서로 취급되어 정확하지 않은 검색 결과를 유도할 수 있다. 또한 웹 상의 데이터가 계속해서 기하급수적으로 증가하면서 사용자의 요구에 대해 보다 정확하고 풍부한 데이터를 검색하기 위한 확장된 검색 방법이 필요하다. 예를 들면, 실제 문서 내에 표현되어 있지는 않지만 보다 정확한 검색에 활용할 수 있는 저자, 요약 내용, 만든 날짜, 핵심 키워드 등의 메타데이터를 이용한 확장 검색 방법을 고려해볼 수 있다.

메타데이터는 데이터에 대한 데이터로서, 데이터의 컨텐츠, 필요 조건, 특성 등을 의미한다. 또한 메타데이터는 데이터간의 의미적인 관계를 기술하여 데이터 내에 숨겨져 있는 정보를 유도해내는 역할을 담당한다[2]. 그러므로, 이러한 메타데이터의 특성을 이용하면 사용자의 요구에 대한 보다 정확하고 풍부한 데이터 검색이 가능하다. 따라서 메타데이터에 대한 중요성이 점차 부각되고 있으며, 차세대 웹 환경인 시멘트 웹에

서의 메타데이터 역할에 관해 많은 기대가 모아지고 있다[3].

RDF(Resource Description Framework)는 웹 상의 데이터에 대한 메타데이터를 기술하기 위해 W3C에서 제안한 표준으로서, 데이터의 의미(semantic)와 추론을 위해 필요한 정보를 기술하는데 그 목적을 두고 있다[4]. RDF는 메타데이터의 기술을 위하여 XML의 문법 구조를 이용한다. 따라서 RDF 문서도 XML 형태의 문서로서 간주될 수 있고, 간단한 변환 규칙을 적용하여 기존의 XML 문서를 RDF 문서로 쉽게 변환할 수 있다. RDF 형태의 문서는 리소스/속성/값이라는 반복된 구조로 구성되어 있다. 또한 표준화된 용어의 사용과 데이터간의 의미적 관계에 대한 기술이 가능하기 때문에 XML을 RDF 형태로 작성할 경우 많은 이점을 얻을 수 있다.

따라서 본 논문에서는 첫째, 기존의 XML 문서를 RDF 형태로 변환하기 위한 간단한 변환 규칙을 소개한다. 둘째, 변환된 RDF 문서를 관계형 데이터베이스에 저장하기 위한 저장 구조를 제안한다. 제안한 저장 구조는 RDF 문서의 리소스/속성/값의 구조에 기반하여 3개의 테이블로 구성되어 있기 때문에 RDF의 특성을 고려한 데이터의 저장이 가능하다. 마지막으로, RDF 문서에 대한 키워드 질의 처리 시에 적용되는 검색 결과의 랭킹 평가 기법을 제안한다.

### 2. 관련 연구

RDF 형태로 표현된 데이터의 저장 및 검색을 위한 시스템 개발에 관한 연구가 최근 몇몇 외국 연구 기관에서 이루어지고 있다. 이러한 연구의 대표적인 결과물로 RDFSuite, Sesame 등이 있다. RDFSuite는 RDF 문서의 검증을 위한 파서인 VPR, RDF의 저장 구조인 RSSDB, 객

본 연구는 한국과학재단 기초과학연구사업(과제번호: F01-2002-000-00080-02002)의 지원을 받아서 작성되었음

체-관계 데이터베이스에 기반한 RDF의 질의 처리의 집합체이다[3]. RDFSuite의 경우, 객체-관계 데이터베이스에 기반한 저장 구조와 질의 처리 모듈을 개발하였다. 최근 객체-관계 데이터베이스에 대한 관심이 높아지고 이를 이용한 애플리케이션의 개발이 많이 이루어지고 있으나, 여전히 많은 애플리케이션이 관계형 데이터베이스를 기반으로 개발되고 있으므로, RDF를 관계형 데이터베이스에 저장하기 위한 구조의 개발이 필요하다. Sesame는 RDF와 RDF 스키마 정보를 위한 저장 시스템으로서, 어떠한 하부 저장 구조와도 호환이 가능하도록 RAL(Repository Abstraction Layer)라는 미들웨어와 RDF 처리를 위한 3가지 기능성 모듈을 개발하였다[5]. Sesame 시스템은 대부분의 저장 구조와의 호환이 가능한 장점이 있으나, 이를 위하여 모든 처리 과정은 하부의 저장 구조에 맞게 변환하는 과정이 반드시 요구되므로 그에 따른 오버헤드가 크다.

또한 RDFSuite와 Sesame 시스템은 RDF를 위한 질의어로 제한된 RQL을 통하여 RDF에 관한 질의를 처리한다. RQL은 RDF 스키마 정보를 이용하여 클래스간의 관계, 또는 속성간의 관계와 관련한 질의나 일반적인 RDF 관련 질의를 표현할 수 있으나 RDF에 관한 키워드 기반 질의를 지원하지 않는다. 따라서 RDF에 관한 키워드 질의와 관련한 처리 기법이 요구된다. 특히 XML 문서를 RDF 형태로 작성하거나, 기존의 XML 문서를 RDF로 변환하는 경우, 이와 같은 키워드 질의에 대한 고려가 필요하다.

### 3. XML의 RDF 변환 규칙

RDF가 메타데이터 기술을 위한 표준이지만 일반 데이터의 표현이 가능하고, XML의 기본 문법 구조에 따라 작성되기 때문에 XML 문서를 RDF 형태로 작성하거나 약간의 수정을 통해 기존의 XML문서를 RDF 형태로 변환하는 것이 가능하다[6]. XML 문서는 논리적 구조를 가지고 있고, 사용자가 정의한 태그의 활용이 가능하다는 장점이 있으나, 비정규적인 논리적 구조와 같은 의미를 사용자마다 다르게 표현하는 문제로 인해 원하지 않는 검색 결과를 초래하거나, 중요한 결과를 놓치게 되는 경우가 발생할 수 있다. 이와 같은 문제를 리소스/속성/값이라는 정규적인 구조로 표현되고, 표준화된 용어를 사용하는 RDF 형태를 이용하여 해결하려는 노력이 필요하다.

RDF 문서에서 리소스나 속성 모두 XML 문서의 엘리먼트처럼 표현이 되지만, 리소스/속성/값이라는 기본 구조를 반드시 갖추도록 변환해야 한다. 또 속성은 그 값으로 반드시 일반 데이터 값이나 리소스만을 가질 수 있다. RDF 형태의 표현은 RDF 작성 규칙을 엄격하게 지키는 정규형과 어느 정도의 융통성을 허용하여 보다 쉽게 표현하는 축약형으로 나눌 수가 있다. 본 논문에서는 변환되기 전의 XML 문서와 보다 유사한 형태로 변환하기 위해 축약형을 기준으로 하여, 변환시 약간의 수정만을 필요로 하도록 하였다. XML을 RDF로 변환하기 위한 간단한 변환 규칙에는 다음과 같은 내용이 있다.

- (1) 엘리먼트 중에서 루트 엘리먼트는 적절한 식별자를 부여하고 리소스로 변환한다.
- (2) XML 문서 내의 기본 네임스페이스에 대해 ID 속성 값을 가지고 있는 엘리먼트(태그)정보는 리소스로 변환한다.
- (3) 엘리먼트 내에 실제 데이터 값을 표현하고 있는 경우, 해당 엘리먼트는 속성으로 변환한다.
- (4) 또 다른 엘리먼트를 내포하고 있는 엘리먼트는 리소스로 변환한다.
- (5) 변환 후 리소스가 속성을 통하지 않고 바로 다른 리소스와 관계를 맺는 경우, 즉 다른 리소스를 바로 포함하고 있는 경우는 두 리소스간의 적절한 관계인 속성을 추론하여, 속성을 거쳐서 다른 리소스를 포함하도록 변환한다.
- (6) 속성이 속성의 값으로 다른 속성들을 포함하는 경우는 rdf:Bag, rdf:Seq, rdf:Alt 이나 임의의 리소스(anonymous resource)를 생성하여 속성들을 포함하도록 한다.

<그림 1>은 도서관에 대한 데이터를 기술하는 XML 문서로서 도서관의 위치, 도서관 내의 책에 대한 정보를 담고 있다.

<그림 2>는 <그림 1>의 XML 문서를 본 논문에서 소개한 변환 규

칙에 따라 간단하게 RDF 형태로 변환한 예를 보여준다.

```
<?xml version="1.0"?>
<library xmlns="http://www.zvon.org/library#">
  <location>
    <address>Kyobo Building</address>
  </location>
  <author uri="http://www.zvon.org/library#RD">
    <firstName>Rohms</firstName>
    <lastName>OhK</lastName>
    <catNo1>877-0000</catNo1>
    <books>
      <name id="Mallida"/>
      <name id="TheBFG"/>
    </books>
  </author>
  <book name="Mallida">
    <pages>240</pages>
    <publisher>Kyobok</publisher>
    <data>2003-08-19</data>
    <cost>8120</cost>
    <copy>
      <catNo>M124</catNo>
      <catNo>M125</catNo>
    </copy>
  </book>
  <book name="TheBFG">
    <pages>300</pages>
    <publisher>Information Culture</publisher>
    <data>2003-07-23</data>
    <cost>8180</cost>
    <copy>
      <catNo>T450</catNo>
    </copy>
  </book>
</library>
```

그림 1. 도서관 정보를 표현한 XML 문서

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:lib="http://www.zvon.org/library#">
  <lib:Description rdf:about="http://www.zvon.org/library#location">
    <lib:address>Kyobo Building</lib:address>
  </lib:Description>
  <lib:Description rdf:about="http://www.zvon.org/library#RD">
    <lib:firstName>Rohms</lib:firstName>
    <lib:lastName>OhK</lib:lastName>
    <lib:catNo1>877-0000</lib:catNo1>
    <lib:books>
      <lib:Bag>
        <lib:_1 rdf:resource="http://www.zvon.org/library#Mallida"/>
        <lib:_2 rdf:resource="http://www.zvon.org/library#TheBFG"/>
      </lib:Bag>
    </lib:books>
  </lib:Description>
  <lib:Description rdf:about="http://www.zvon.org/library#Book1">
    <lib:type rdf:resource="http://www.zvon.org/library#Book1">
      <lib:pages>240</lib:pages>
      <lib:publishMyBo</lib:publish>
      <lib:data>2003-08-19</lib:data>
      <lib:cost>8120</lib:cost>
      <lib:copy>
        <lib:Seq>
          <lib:_1 rdf:resource="http://www.zvon.org/library#M124"/>
          <lib:_2 rdf:resource="http://www.zvon.org/library#M125"/>
        </lib:Seq>
      </lib:copy>
    </lib:Description>
  <lib:Description rdf:about="http://www.zvon.org/library#TheBFG">
    <lib:type rdf:resource="http://www.zvon.org/library#Book1">
      <lib:pages>300</lib:pages>
      <lib:publishInformation Culture</lib:publish>
      <lib:data>2003-07-23</lib:data>
      <lib:cost>8180</lib:cost>
      <lib:copy rdf:resource="http://www.zvon.org/library#T450"/>
    </lib:Description>
  </lib:RDF>
```

그림 2. XML의 RDF 변환 결과

### 4. RDF의 저장 구조

본 논문에서는 관계형 데이터베이스에 기반한 RDF 정보의 저장 구조를 제안한다. 본 논문에서 제안하는 RDF 저장 구조는 리소스 테이블, 속성 테이블, 값 테이블로 구성되어 있다. 제안한 저장 구조는 RDF의 기본 구조를 고려하였기 때문에, RDF 문서의 구성 요소에 관한 일반적인 질의 처리가 가능하다.

리소스 테이블은 RDF 문서 내에 표현되어 있는 웹 상의 리소스에 대한 네임스페이스 정보를 저장한다. 리소스를 표현하는 네임스페이스에는 각 리소스에 대해 고유하게 정의된 아이디가 함께 표현되어 있기 때문에 네임스페이스에 대한 정보를 테이블에 유지함으로써 아이디 값을 이용한 리소스 정보에 대한 검색이 가능하다.

속성

source	title	target	target_type
1	address	1	Value
2	type	3	Resource
2	firstName	2	Value
2	lastName	3	Value
2	catNo1	4	Value
2	lib	5	Value
3	books	4	Resource
4	type	5	Resource
4	lib1	6	Resource
4	lib2	7	Resource

  

리소스

ID	uri
0	http://www.w3.org/1999/02/22-rdf-syntax-ns#
1	http://www.zvon.org/library#location
2	http://www.zvon.org/library#RD
3	http://www.zvon.org/library#lib
4	anonymous resource1
5	lib:Bag
6	http://www.zvon.org/library#Mallida
7	http://www.zvon.org/library#TheBFG
8	anonymous resource2
9	http://www.zvon.org/library#Book
10	lib:Seq
11	http://www.zvon.org/library#M124
12	http://www.zvon.org/library#M125
13	http://www.zvon.org/library#T450

  

ID	type	data
1	string	Kyobo Building
2	string	Rohms
3	string	OhK
4	string	877-0000
5	string	lib
6	string	240
7	string	Kyobo
8	string	2003-08-19
9	currency	8120
10	string	M124
11	string	Information Culture
12	string	2003-07-23
13	currency	8180

그림 3. RDF 문서의 실제 저장 예

속성 테이블은 리소스와 리소스간의 관계 또는 리소스와 값과의 관계를 기술하는 속성 정보를 저장한다. 속성 테이블은 속성을 포함하고

있는 시작 리소스의 키 값, 속성의 타이틀, 속성에 대한 값의 역할을 담당하는 타겟 리소스 또는 실제 데이터 값에 대한 키 값으로 구성되어 있다. 그리고, 속성의 값으로 리소스나 실제 데이터와 같은 두 가지 타겟 정보를 가지게 되므로 이를 구분할 수 있는 타겟의 타입 정보를 포함한다. 속성 테이블은 시작 리소스의 키 값을 기준으로 정렬하기 때문에, 특정 리소스의 모든 속성 정보에 대한 검색이 용이하다.

값 테이블은 RDF 문서 내의 실제 데이터 값에 대한 정보를 저장하고 있으며, 데이터의 실제 내용과 그 타입 정보, 키 값으로 구성된다.

<그림 3>은 <그림 2>의 RDF 문서를 제안한 저장 구조에 저장한 실제 예를 보여준다.

### 5. 키워드 질의 처리를 위한 랭킹 평가 기법

사용자가 제시한 키워드와 관련한 문서의 부분이나 또는 전체 문서를 검색해서 결과를 반환하는 것이 키워드 검색의 일반적인 처리 과정이다. RDF 문서에 대한 키워드 검색이 가능하지만 RDF에 대한 키워드 검색은 XML이나 HTML과는 조금 다른 특성을 지닌다. RDF 문서는 웹 상의 리소스에 대한 정보를 기술한 것이므로 특정 태그를 포함한 부분이나, 전체 문서보다는 해당 정보를 포함하는 리소스를 질의 결과의 반환 단위로 고려해야 한다. 또한, 만약 많은 리소스가 결과로서 반환되었다면, 각 리소스의 중요도를 파악하여 중요한 리소스부터 사용자에게 반환할 수 있는 랭킹 평가 기법이 필요하다.

본 논문에서는 RDF 문서와 관련한 키워드 검색 시 적용할 수 있는 랭킹 평가 기법을 다음과 같이 정의한다.

- (1) 사용자로부터 주어질 여러 개의 키워드간의 거리를 가깝게 유지하는 리소스를 더 높은 결과의 정확도를 가지는 것으로 간주한다. RDF 문서를 그래프 형태로 모델링 한 후 모든 키워드를 방문하는데 소요되는 총 비용을 이용해 키워드의 근접 정도를 평가한다. 즉, 총 방문 비용이 적은 리소스는 근접도가 크다고 할 수 있으므로, 높은 랭킹을 부여해야한다. 예를 들어, "Ronald"와 "KyoBo"가 사용자로부터 주어졌다고 가정하면, <그림 2>의 문서에서는 9번째 줄에서 "Ronald"를 포함하고 23번째 줄에서 "KyoBo"를 포함하는 "http://www.zvon.org/library#RD" 리소스에 대한 부분과 5번째 줄에서 "KyoBo"를, 9번째 줄에서 "Ronald"를 포함하는 전체 RDF 기술 문서가 결과 값으로 반환될 것이다. 두 개의 키워드를 모두 포함하고 있지만, RDF 그래프 관점에서 "http://www.zvon.org/library#RD" 리소스에 대한 기술 부분에서의 키워드 근접도가 더 크다고 할 수 있으므로, 이 부분에 더 높은 랭킹을 부여한다.
- (2) 모든 키워드를 직접 포함하고 있는 리소스는 간접적으로 포함하고 있는 리소스보다 더 높은 랭킹을 갖는다. 예를 들어, 사용자가 "Information"과 "Culture"를 질의 조건으로 하는 경우에 <그림 2>의 문서에서는 "http://www.zvon.org/TheRFG" 리소스에 대한 기술 부분과 "http://www.zvon.org/library#RD" 리소스에 대한 기술 부분이 결과 값으로 반환된다. 그러나 "http://www.zvon.org/TheRFG" 리소스에 대한 기술 부분이 모든 키워드를 직접적으로 포함하고 있으므로 더 높은 결과의 정확도를 가지도록 높은 랭킹을 부여한다.
- (3) 만약 어떤 리소스 내부에 다양한 속성 관계로 포함되어 있는 리소스들이 모두 높은 랭킹을 가지고 있다면, 이 리소스 또한 어느 정도 높은 랭킹을 가지도록 고려해야 한다. 반대로 어떤 리소스가 키워드들에 대해 높은 랭킹을 가지고 있다면, 그 리소스와 여러 속성 관계로 관계를 맺고 있는 다른 리소스들도 높은 랭킹의 영향을 받아야 한다.

본 논문에서는 웹 페이지에 대한 랭킹을 평가하는 페이지 랭크 합수를 응용하여, 키워드 질의에 대해 RDF 문서 내 각 리소스의 중요성을 평가하기 위한 *ResourceRanks* 합수를 다음과 같이 정의한다[7,8].

$$ResourceRanks(r) = (1 - d_1 - d_2) + d_1 \times \left( \frac{ResourceRanks(r_1)}{Out(r_1)} + \dots + \frac{ResourceRanks(r_n)}{Out(r_n)} \right) + d_2 \times \left( \frac{ResourceRanks(r'_1)}{In(r'_1)} + \dots + \frac{ResourceRanks(r'_m)}{In(r'_m)} \right)$$

위의 식에서,  $r$ 는 RDF 문서 내에 기술된 리소스를 의미한다.  $r$  리소스로부터 다양한 속성 관계를 통해  $r_1 \dots r_n$  리소스를 포함하는 관계를 가지고 있다고 가정한다. 또,  $r$  리소스는 다양한 속성 관계를 통해  $r_1 \dots r_m$  리소스로부터의 포함관계를 가지고 있다고 가정한다.  $d_1$ 은  $r$  리소스를 통해서 내부의 리소스들을 방문하게 되는 확률을 의미한다.  $d_2$ 는 외부의 리소스들을 통해서  $r$  리소스를 참조하게 되는 확률을 의미한다. *Out* 함수는 각 리소스가 포함하고 있는 내부 리소스의 개수를 계산하기 위한 함수이고, *In* 함수는 속성 관계를 통해 해당 리소스를 포함하고 있는 리소스들의 개수를 계산하기 위한 함수이다.

여러 개의 키워드가 주어질 경우에는, 모든 키워드에 대한 리소스의 랭크 정보의 합뿐만 아니라 키워드들의 근접도 정도를 고려하여 랭크를 평가하게 된다. 키워드들의 근접도 정도는 RDF를 그래프로 표현했을 때, 모든 키워드를 방문하는데 소요되는 총 비용과 반비례하는 것으로 평가한다.

### 6. 결론

본 논문에서는 메타데이터의 기술을 위한 표준인 RDF로 데이터 표현하는 것의 장점들을 분석하고, 웹 상의 데이터에 대한 RDF 표현을 일반화시키기 위한 방법으로, 기존 XML 문서에 대한 RDF 변환 규칙을 간단히 소개하였다. 그리고, RDF의 리소스/속성/값이라는 기본 구조를 고려하여 관계형 데이터베이스에 기반한 RDF 저장 구조를 제안하였다. 제안한 저장 구조는 리소스 테이블, 속성 테이블, 값 테이블로 구성되고, RDF 문서 내의 리소스에 대한 정보, 리소스와 리소스간의 관계 또는 리소스와 데이터간의 관계에 대한 정보, 실제 데이터 값에 대한 정보를 표현한다. 따라서 제안한 저장 구조를 활용하여 RDF 문서 내의 일반적인 구성 요소에 대한 검색이 가능하다. 또한 RDF 문서에 대한 키워드 질의 처리도 고려하였다. RDF에 대한 키워드 검색 질의는 문서 전체나 특정 태그가 아닌 키워드를 포함하고 있는 리소스를 검색 결과의 단위로 하는 특징을 가지고 있다. 이러한 특징에 따라 본 논문에서는 RDF에 대한 키워드 검색 질의를 지원하기 위하여 검색의 결과로 반환된 다양한 리소스들의 중요도를 평가할 수 있도록 랭크 평가 전략과 기존의 페이지 랭크 합수를 응용한 랭크 평가 합수를 정의하였다.

### 참고 문헌

- [1] Mounia Lalmas and Ian Ruthven, "A Model for Structured Document Retrieval: Empirical Investigations", In Proceedings of the Hypertext, Information Retrieval, and Multimedia, 1997.
- [2] Peter F. Patel-Schneider, Jérôme Siméon, "The Yin/Yang web: XML syntax and RDF semantics", WWW 2002, pages 443-453.
- [3] Sofia Alexaki, Vassilis Christophides, Gregory Karvounarakis, Dimitris Plexousakis, and Karsten Tolle, "The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases", SemWeb 2001.
- [4] W3C, RDF Primer, <http://www.w3c.org/TR/2003/WD-rdf-primer-20030123>.
- [5] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema", International Semantic Web Conference 2002.
- [6] Bob DuCharme and John Cowan, "Make Your XML RDF-Friendly",
- [7] Larry Page, Sergey Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", Stanford Digital Library Technologies Project, 1998.
- [8] Lin Guo, Feng Shao, Chavdar Botev, and Jayavel Shanmugasundaram, "XRANK: Ranked Keyword Search over XML Documents", SIGMOD Conference 2003, pages 16-27.