가변 K진 완전트리와 RDF메타정보에 기반한 XML문서 저장 및 검색 프레임워크의 설계 및 구현

김규태* · 정회경**· 이수연*

A Design and Implementation of XML Document storing and retrieval Framework based on a variant k-ary complete tree and RDF Metadata

Kyu-Tae Kim* · Hoe-Kyung Jung** · Soo-youn Lee*

이 논문은 2003년도 광운대학교 교내연구비를 지원받았음

요 약

XML문서가 표준 인터넷 문서로 정착되어 감에 따라 XML문서의 효율적인 저장과 검색의 필요성이 증대하고 있다. 이에 본 논문에서는 XML문서의 효과적인 저장 모듈과 검색 모듈, 그리고 이를 웹 상에서 연동해서 사용할 수 있는 연동 API로 구성된 XML문서의 저장 및 검색 프레임워크에 대한 연구를 하였다. 저장 모듈에서는 가변 K진 완전트리를 기반으로 한 DTD 독립적인 분할-통합형 저장모델을 구현하였고, 검색 모듈에서는 RDF 메타정보를 통해 구축된 색인에 대해 XPath 질의를 수행하는 XPath처리기를 구현함으로써 좀더 의미 있는 구조 검색 기능을 구현하였으며, XML-RPC, HTTP의 GET, POST, PUT 방식 API와 SOAP 방식의 API로 구성된 웹 연동 모듈을 구현하였다.

ABSTRACT

This paper studied and proposed a XML document storing-and-retrieval framework based on a variant k-ary complete tree and a RDF metadata, which is composed of an effective storing module to store xml documents, a retrieving module to retrieve xml documents, and a connecting module to make this system interoperate in the web environment. In this storing module, DTD independent DOM based decomposition model using a method of addressing unique ID using a variant k-ary complete tree is adopted and is implemented. Query Processing Module includes a XPath query process and a content based retrieval function using word index for content information. To retrieve more exactly data, a structural retrieval using RDF metadata is adopted and implemented. In order to implement effectively XML document storing and retrieval system in the web environment, API using XML-RPC, API using HTTP's GET, PUT, POST and API using SOAP have been adopted and implemented.

키워드

XML, Database, RDF Metadata, XPath, DOM Model

*광운대학교 컴퓨터공학부

**배재대학교 정보통신공학부 교수

접수일자 : 2003. 6. 13

Ⅰ. 서 론

온라인상에서 대량의 정보 문서의 표현 및 구조화를 위해 1986년 SGML(Standard Generalized Markup Language:ISO 8879)[1,2]이 국제 표준으로 제정되어 사용되어 왔으며, 이 SGML의 한 응용인 HTML(Hyper Text Markup Language)[3]를 사용한 웹의 빠른 보급으로 인해 인터넷은 급속히 확장되어, 일상의 많은 정보가 HTML로 표현되기에 이르렀다. 그러나 HTML은 고정된 태그집합만을 사용해야 하는 태생적 한계 때문에 매우 단순한 구조로 고정되어 있어 보다 복잡 다양하고 정교한 구조화 문서를 만들어 내기에는 많은 한계점들을 가지고 있었다. 이러한 HTML의 한계를 극복하고 웹의 효용성 및 수용도를 높이기 위해 1988년에 W3C(World Wide Web Consortium)의 추천안으로 XML(extensible Markup Language)[4]이 소개됨에 따라 XML에 기반한 구조화 문서 정보 인프라 구축의 저변이 급속히 확대되어 가고 있으며 XML은 차세대 웹 문서로서 인정받고 있다.

XML은 HTML과 같은 마크업 언어를 정의할 수 있는 메타언어로서 정보의 생성 관리 전달을 정보 구조화라는 새로운 관점에서 바라보게 됨에 따라 기존의 정보를 단순히 생성 측면 보다는 정보의 효율화를 위해 정보의 재사용과 유통의 장점을 가지고 있다. 정보의 구조화를 강조하는 XML 문서들이 폭발적으로 생성됨에 따라 이 문서들을 보다 효과적으로 다룰 필요성이 증대하고 있다.

대용량의 XML문서를 효율적으로 다루기 위해 기존의 데이터베이스를 이용한 많은 연구들이 진행되고 있다[6-16]. 이러한 하부 데이터베이스의 모델에 따라 관계형 데이터베이스를 이용하는 방식과 객체형 데이터베이스를 이용한 방식으로 나눌 수 있다. 관계 데이터베이스를 이용하는 방법은 RDBMS의 우수한 성능을 이용할 수 있고 기존 응용시스템의 데이터를 함께 사용할 수 있는 장점을 가진다. 그러나 검색 시 다수의 테이블에 대한 고비용의 조인 연산을 수행해야 하며 검색 결과를 추출하는데 많은 노력이 필요하다. 그리고, 객체지향 데이터베이스를 이용한 방법은 데이터베이스에서 지원하는 객체 지향적 개념을 이용할 수 있기 때문에 상속과 같은 객체지향 특성을 이용할 수 있으며, 엘리먼트 간의 전후 종속 관계를 클래스에 기반한 객체들 간의 링크로 나타낼 수 있기 때문에 구조적인 문서를 모델링 하는데 적합하지만 현재의 객체지향 데이터베이스는 대용량의 데이터에 대한 복잡한 형태의 질의를 처리하는 능력이 성숙되지 않았다는 문제점이 있다[10]. 또한 XML문서는 그 구조적 성격에 따라 자료중심(Data-Centric) XML 문서와 문서중심(Document-Centric) XML문서로 나뉘는데[7,10], 최근까지 개발된 XML문서의 저장에 대한 연구는 둘 중 어느 하나의 유형에 적합한 구조로 설계 및 연구되어 왔으나 두 가지 유형을 효과적으로 지원하기에는 한계가 있으며, 특히 관계형 데이터베이스의 장점을 살리면서 문서 중심의 XML문서를 효율적으로 저장 관리하는 연구는 많은 한계가 있었다[7,8,10].

이에 본 논문에서는 관계형 데이터베이스의 장점을 살리면서 자료중심의 XML문서 뿐만 아니라 문서중심의 XML문서의 처리에도 적합한 저장 방법에 대한 연구를 하였다. 본 논문에서는 DTD 독립적인 DOM기반 분할기법모델을 적용하고 각 분할요소 즉 노드의 주소기법을 트리깊이에 따라 가변하는 K진 완전트리의 모델을 사용해서 K진 완전트리의 장점인 DB엑세스를 최소화하면서, 필요한 부모/자식 노드들의 위치를 연산하여 검색할 수 있었으며, 더불어 K진 완전트리의 많은 NULL노드 생성 및 ID 낭비로 발생하는 비효율을 줄여 문서 중심의 XML문서에도 적합한 저장, 검색 방법을 제안하였다.

대량의 XML문서를 검색하는데 있어서는 기존의 HTML문서의 저장 검색과는 많은 차별화가 필요하게 필요하다. 기존의 HTML은 여러 검색 엔진들을 통해서 색인 단어에 의한 내용 검색을 문서단위로 수행하는 정도였다. 이러한 이유로 검색 수행 결과 정확한 검색이 어려웠고 검색된 결과를 이용 다시 검색해야 하는 추가작업이 필수적이었다. 하지만 XML 문서의 경우 사용자가 의미 있는 엘리먼트 집합이나 구조를 정의할 수 있기 때문에, 하나의 문서에서 내용정보와 함께 문서의 구조 정보 즉, 제목, 서론, 본문, 장, 절, 결론과 같은 문서의 논리적인 구조 정보를 지니고 있다. 따라서, 기존의 문서에서 제공하던 내용정보에 대한 검색뿐만 아니라 XML문서의 검색에서는 이러한 논리적인 구조 정보에 대한 검색 기능을 통해 좀더 정확한 검색을 할 수 있어 이에 대한 연구가 활발하

다[17,18,19].

이러한 구조적인 검색을 위해 본 논문에서는 XML 문서의 구조질의에 적합한 XPath를 기반으로 문서의 콜렉션까지 검색할 수 있게 확장하고, 내용 정보에 대하여 단어색인을 제시함으로써 내용정보 구조까지 검색할 수 있는 구조기반 검색 기능을 구현하였고 저장 모델에 저장된 모든 구조정보에 대해 사용자가 정의한 RDF(Resource Description Framework)메타정보를 이용해서 색인함으로써 의미 있는 메타정보에 기반한 구조 검색을 제시하였다.

또한 인터넷의 발달로 정보의 활용도가 높아지면서 애플리케이션 개발과 운용에 있어서 사용자 서비스에 대한 수준도 양적으로나 질적으로 향상되게 되었다. 처음에 XML은 데이터 포맷과 애플리케이션간 메시징과 같은 애플리케이션의 외적인 요소들을 주로 정의하는데서 시작되었다. 하지만 XML이 가지는 특성들로 인해 최근에는 애플리케이션 개발과 운용 그리고 애플리케이션간 상호 운용성에 널리 사용되고 있는 중이다[20]. 따라서 본 논문에서는 인터넷에서의 타 응용과의 연동을 위해서, 즉 XML저장,검색 시스템의 웹상에서의 연동성을 높이기 위해, HTTP의 GET,POST,PUT 방식의 API와 XML프로토콜로서 중요한 흐름인 XML-RPC (XML-Remote Procedure Call)를 이용한 API 및 SOAP(Simple Object Access Protocol)를 이용한 API를 설계 구현함으로써, 웹상에서 XML 저장 및 검색 시스템을 활용할 수 있는 방식을 구현 예시하였다.

본 논문의 구성은 제2장에서는 ID 부여 방식인 가변 K진 완전트리 모델에 대하여 기술하며, 제3장에서는 가변 K진 완전트리 및 RDF 메타정보에 기반한 XML 문서 저장 검색 프레임워크의 설계에 대하여 기술한다. 그리고 제4장에서는 제시한 XML문서 저장 및 검색 프레임워크의 구현 결과와 이에 대한 고찰을 기술하고 제5장에서는 결론과 함께 향후 연구과제에 대하여 논의한다.

II. 가변 K진 완전트리 구조정보 표현 모델

2.1 기존의 구조 정보 표현 모델

현재 구조정보를 표현하기 위한 모델로는 SCL(Simple Concordance List)모델, K진 완전 트리 모델, 그리고 ETID(Element Type ID)모델 등이 있다.

```
 100.1        100.2      101 102 103 104    104.1
<doc n=1975><dtitle> Tarzan of the Apes </dtitle>
 104.2          105            105.1
<div type=toc>Contents ... </div>
 105.2                    106   107 108 109
<div type=chapter id=C7> The Light of Knowledge ...
 109.1 110 111 112    113    114 115 116   117   118   119
 <p> ... Let all respect Tarzan of the Apes and Kala, his
         120        120.1        120.2
         mother     </p>     </div>
```

그림 1 SCL모델
Fig 1 SCL Mode

먼저 SCL모델을 살펴보면 SCL모델은 그림1과 같이 구조 문서의 계층적 관계보다는 포함 관계를 이용한 표현방법으로서 SC-list 라는 데이터 타입을 통해 중첩된 정보를 허용하므로 리스트의 리스트와 같은 순환구조를 다룰 수 있다는 장점이 있다. 이 모델은 텍스트와 마크업에 대해 색인 넘버를 부여한 후, 불용어를 제외한 텍스트 어휘들을 텍스트 인덱스에 색인 넘버로 저장하고, 마크업은 시작 태그와 종료 태그의 쌍으로 마크업 인덱스에 저장한다. 그러나 SCL구조는 트리의 깊이를 표현할 수 없으므로 조상이나 형제 엘리먼트를 검색할 수 없다는 큰 단점이 있다.

ETID 모델은 그림2와 같이 엘리먼트들 간의 계층 정보와 동일 부모 엘리먼트를 갖는 자식 엘리먼트들의 순서 정보(SORD: Sibling ORDer), 그리고 동일한 엘리먼트를 갖는 자식들 중 동일 한 타입의 엘리먼트들에 대한 순서 정보(Same Sibling ORDer)를 통해 구조 문서를 표현한다. 이 모델에서는 XML문서내의 엘리먼트에 고유한 TypeID를 먼저 부여한다. 이는 DTD내에 정의된 순서대로 부여하며. DTD내의 각 엘리먼트를 유일하게 구분할 수 있도록 해준다. 그리고 루트 노드의 ETID는 TypeID로 결정되며 자식노드의 ETID는 부모의 ETID를 상속한다음 뒤에 자신의 TypeID를 붙임으로서 부여된다. 그리고 그림2에서 맨 마지막 노드의 SORD,SSORD값은 각각 /1/3, /1/1로서 앞의것은 루트의 첫번째 자식노드의 세번째 자식노드임을 뜻하며, 뒤의 것은 루트의 첫번째 자식노드중에 처음 나오는 다른 종류의 노드임을 나타낸다.

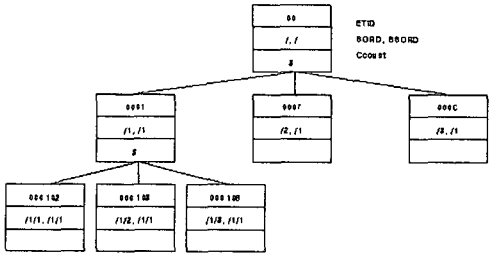
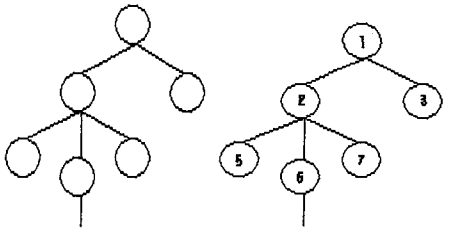


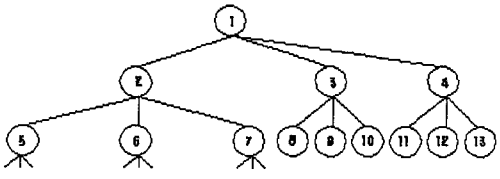
그림 2 ETID모델
Fig 2 ETID Model

이 방법은 기준 엘리먼트로부터 특정 엘리먼트에 대한 계층 정보와 순서 정보를 간단한 문자열 조작만으로 쉽게 구할 수 있다는 장점이 있는데 반해 트리의 깊이가 깊어질수록 각 노드를 표현하기 위한 공간이 무한대로 늘어난다는 단점이 있다.

K진 완전 트리 모델은 문서에 대한 트리로부터 이들 노드중 가장 큰 차수 K를 구하여 K진 완전 트리로 재구성한 후, 여기에 문서 트리를 매핑하여 각 노드에 모든 번호를 부여한다. 이 모델은 문서 구조 사이의 계층 관계를 간단한 공식을 통해 쉽게 구할 수 있다는 장점이 있는 반면, 매핑 과정에서 NULL 노드가 많아질 수 있고 노드의 깊이가 깊어질수록 노드 변화가 커진다는 단점이 있다. 그림3은 K진 완전 트리 모델의 예이다



(a)파싱트리 (b) 노드값을 부여한 파싱트리
(a) Parsing Tree (b) Parsing tree with node value



(c)K진 완전트리(k=3)

(c)k-ary complete tree with k=3
그림 3 K진 완전트리(k=3)
Fig.3 k-ary complete tree with k

K진 완전 트리 모델은 문서 구조 사이의 계층 관계를 간단한 공식을 통해 쉽게 구할 수 있다는 장점이 있는 반면, 매핑 과정에서 NULL 노드가 많아질 수 있고 노드의 깊이가 깊어질수록 노드 변화가 커진다는 단점이 있다. 일반적으로 K값은 노드들의 차수 중 제일 큰 값을 가정함으로 문서중심의 XML문서에선 매우 많은 NULL값을 갖게 된다. 따라서 본 논문에서는 매핑시 트리 깊이 레벨에 따른 가변 K진 완전 트리 모델의 매핑 방식을 사용함으로써 기본적으로 완전 k진 트리 모델이 갖는 엘리먼트간의 계층정보뿐만 아니라, 형제노드의 순서를 표현함으로 임의의 엘리먼트로부터 간단한 검색 과정을 통해 특정 엘리먼트에 쉽게 접근할 수 있으며 ETID 방식보다 적은 ID 공간을 사용하고, 순수 K진 완전트리모델 보다 적은 ID수를 할당함으로써 낭비되는 ID수를 줄일 수 있는 가변 K진 트리 모델을 저장 모델의 ID부여 기법으로 사용한다.

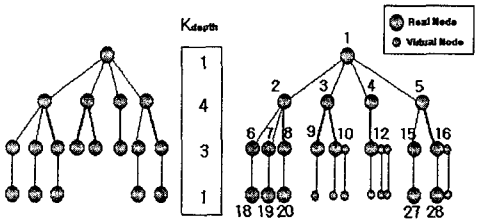


그림 4. 트리 깊이에 따른 다른 K값을 갖는 가변 K진 완전트리모델
Figure 4 A variant K-ary Complete tree

그림4의 예에서는 왼쪽의 트리를 보면 깊이 1레벨은 루트노드 하나이므로 값이 K1=1이고, 깊이 2레벨에는 최대 차수가 4이므로 K2=4가 되고, 마찬가지로 K3=3, K4=1로 현재 문서의 깊이마다 가변 K값을 갖게 된다. 이렇게 구해진 KL은 다음 깊이의 노드의 고정 차수가 되어 실노드는 순차적으로 ID가 부여되고 없는 자리에는 가상노드가 그림4의 작은 원처럼 할당되어 배치된다. 이와 같이 그림 4에서처럼 KL에 의해 L+1레벨의 차수가 그 레벨에서의 완전 K진 트리처럼 고정으로 정해지는 트리를 가변 K진 완전트리라 정의

한다. 이렇게 정의된 가변 K진 완전트리에서 부모 및 자식 노드의 ID를 구하는 연산식은 다음과 같이 구해진다.

(1) N번째 자식노드

Cid=⌊ (GidTreeSP(L)) * KL+1) ⌋ + (N-1)

(2) 부모노드

Pid=⌊ (Gid-TreeSP(L))/KL) ⌋ + TreeSP(L-1)

- 여기서 L=현재노드(Gid)의 트리깊이
- TreeSP(L)은 깊이L에서 첫 번째 Gid 값
- KL은 L레벨에서의 K값

III. 가변K진 완전트리와 RDF메타정보에 기반한 XML문서 저장 및 검색 프레임워크의 설계

3.1 시스템의 설계 및 구성

본 논문의 저장 및 검색기의 전체 구조는 다음 그림에서 보는 것과 같이 응용 계층, 웹으로 연동을 위한 웹 인터페이스 계층, 저장관리기와 검색관리기로 구성된 관리 계층, JDBC로 이루어진 DB연결 계층 그리고, DBMS 계층 등 크게 5개의 계층으로 구성되어 있다.

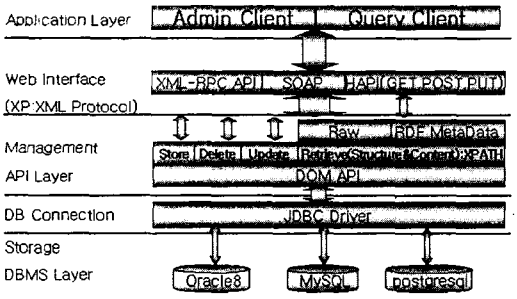


그림5. XML문서저장 및 검색 프레임워크 구조
Fig 5 Framework of XML document storing and retrieval

3.2 문서저장 데이터 모델의 설계

본 논문에서는 자료 중심 및 문서 중심의 XML문서에도 모두 적합한 문서 저장 모델의 설계를 위하여 가변 K진 완전트리의 ID 주소 부여 방식을 적용한 DTD 독립적인 DOM 기반의 범용 저장모델을 설계하였다. XML DOM(Document Object Model)은 XML 문서의 모든 정보를 효과적으로 다루기 위해 트리 구조 형태의 내부 구조로 표현한 규약으로서 각 객체를 분할하여 처리하는데 적합하다. 본 시스템에서 구현한 문서 저장 모델의 테이블 관계도는 다음 그림과 같다.

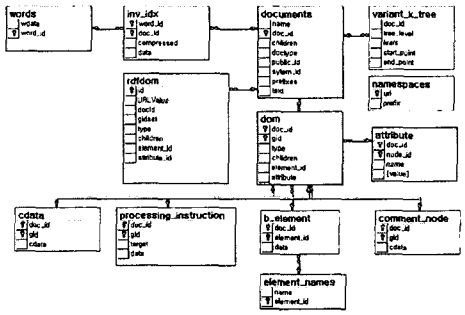


그림 6. DOM 기반의 범용 스키마의 정의
Fig 6. A generic schema based on DOM

우선 XML문서의 DOM의 형태의 구조정보 는 documents, dom, b_element, element_names, attribute, cdata, namespace 테이블로 분할되어 저장되며, 각 XML문서의 가변 k값은variant_k_tree 테이블이 저장이 되고, 내용정보 검색을 위해 words테이블과 RDF를 통한 메타데이타 검색을 위해 rdfdom 테이블이 그림6처럼 설계되었다.

3.3 XPath에 기반한 XML 질의 언어 처리기의 설계

본 논문에서 구현한 질의언어 처리기는 XPath에 기반하여 내용검색을 지원하는 질의 언어처리기이다. 확장 XPath 질의 처리기의 질의 처리 모델은 아래 그림과 같다

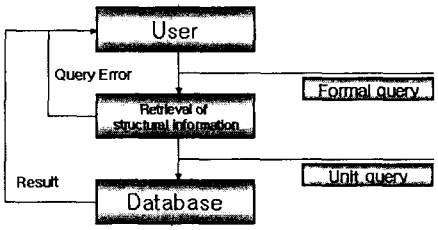


그림 7. XPATH 질의 처리 모델
Fig 7. Process Model of XPath Query

XPath의 기본적인 형식은 URI directory navigation syntax를 따르며 이때 항해는 XML 트리의 각 엘리먼

트 노드를 따라 이동하게 된다. 즉, 결과로 패턴과 일치하는 노드들의 집합을 반환한다. 또한 SQL은 이미 널리 알려진 질의언어로서 관계형 데이터베이스의 데이터를 벤더 독립적으로 구현할 수 있고 그 효용성은 널리 알려져 있다. 하지만 XML과 같은 중첩구조로 구조화된 자료를 검색하기에는 부적합하다. 따라서 XPath 질의 구문처리기에서 XML문서의 구조적인 정보를 처리하고 이를 SQL로 매핑하여 관계형 데이터베이스의 질의에 사용하여 검색 결과를 검색해온다. 구조 검색 기능이 있는 XPath을 접목하여 구조검색 및 내용검색을 할 수 있도록 XPath 언어에 기능을 추가하였다.

확장 XPath 언어는 기본적으로 XPath의 구문을 따르며. 문서집합 단위의 검색 기능과 내용 기반 검색 기능을 추가로 갖는다.

확장 XPath := [document(FileID)] | XPath
XPath := AbbreviatedAbsoluteLocationPath
FileID := FileName | *

여기서 XPath는 W3C XPath 1.0에서 정의하고 있는 XPath 문법의
AbbreviatedAbsoluteLocationPath 노드이다.

기본 처리 모듈은 질의가 정형 질의 형태로 입력되고 이 질의를 분석하여 오류를 체크하게 되고 오류가 없을 경우 검색기에서 단위 질의 형태로 변환/생성하여 인덱스테이블을 통하여 질의하고 결과로 DOM의 노드 ID(Gid)와 DID를 반환하게 된다.

아래 그림은 확장 XPath 문법에 따라 표현한 다양한 질의 표현이다.

1. 구조 기반 검색 질의
   document(*)//Book/title
2. 속성 기반 검색 질의
   documenet(*)//논문[@year="1998"]
3 내용기반 질의
   document(*)//요약[단락 &= "구조"]
4. 복합 질의
   document(*)//요약[@언어="한글" and
   단락 &= "구조"]

그림 8 XML문서 검색 질의 예
Fig 8. An Example of XML Document Retrieval Query Expression

3.4 RDF 스키마문서 생성기의 설계

본 논문에서는 기존의 구조화 문서 기반의 검색기가 가지는 단점을 보완하여 구조화 정보 검색기를 위한 2차 색인 대상을 다음과 같이 정의한다.

(1) 검색 시 의미있는 엘리먼트를 검색 어휘로 사용
(2) 검색 어휘의 기술을 자원 기술 표준인 RDF 스키마 사용

다음은 본 논문에서 구현한 RDF 스키마 문서 생성기의 구조이다

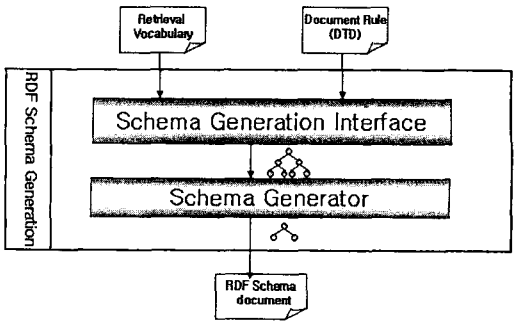


그림 9. RDF 스키마 생성기의 구조
Fig 9. A Structure of the RDF Schema Generator

DTD 트리를 보고 사용자가 선택한 검색 어휘들을 RDF 스키마 문서 생성기 인터페이스가 입력을 받아 RDF 스키마 생성기 모듈에서 문서 생성 규칙에 따라 RDF 스키마 문서를 만든다.

본 논문에서 구현한 RDF 스키마 문서 생성기는 구조화 문서의 DTD로부터 사용자가 지정하는 검색 어휘를 추출하여 검색 어휘를 RDF 문서로 만들어 준다. RDF 스키마 문서 생성 규칙은 다음과 같다.

(1) RDF스키마 문서의 최상위 엘리먼트는 RDF, RDF 스키마, DTD의 네임스페이스 선언을 가진다.
(2) 다음 수준의 엘리먼트는 RDF의 description 엘리먼트를 사용하여 현재 기술하고 있는 DTD 정보를 설명한다.
(3) 검색 어휘로 사용될 엘리먼트, 어트리뷰트를 기

술하고 상위 노드를 RDF의subClassOf를 사용하여 기술한다.

(4) 반복적으로 나타나는 검색어휘는 RDF의 bag, li 엘리먼트를 사용하여 DTD 정보를 잃지 않도록 한다.

결과로 만들어진 RDF 스키마 문서는 자동 색인 생성을 위한 기초 정보로써 활용이 된다. 이것은 자동 색인 생성기의 입력으로 사용이 되지만 이 문서를 웹에서 서비스를 하면 문서 자체가 색인 정보를 담고 있으므로 색인 정보를 바로 서비스 할 수 있는 장점이 있다.

3.5 Web 연동을 위한 외부 API의 구현

외부시스템과의 웹상에서의 효율적인 연결을 위해서는 웹의 기본 프로토콜인 HTTP를 이용해야 추가적인 제약 없이 연동 API의 구현이 가능하다. 그리고 이러한 목적을 위해 XML기반의 표준 프로토콜로서 XML-RPC와 SOAP이 연구 되고 있으므로 이에 기반한 API를 구현하였다. 본절에서는 먼저 XML-RPC를 이용한 API를 설명하고 다음에 HTTP를 이용한 API의 구현, 그리고 마지막으로 SOAP를 이용하여 연동 API를 구현하였다.

XML-RPC는 javax.rpc 패키지를 기반으로 구현되었으며 다음의 API를 가진다.

* 문서 저장 : store()
* 문서리스트검색 : . getDocumentList()
* 문서 삭제 : remove()
* 문서 로드 : getDocument()
* 문서 구조 검색 : query()

HTTP의 3종류의 메서드의 특성을 살려 다음과 같이 구현한다..

(1) GET 메서드

형식 : http://ServerURL:8088/
문서이름.xml?XPathQuery

콜렉션 및 RDF로 매핑된 주소로 해당 자원을 서버로부터 불러오기를 가능하게 하는 getDocument와 query메서드로 사용한다

(2) PUT 메서드

form의 enctype 속성을 multipart /form-data으로 지정하고, 문서 이름을 나타내는 filename를 이용해서 문서 저장한다.

(3) POST 메서드

POST메서드는 메시지 Body 에 텍스트를 입력할 수 있으므로 다음과 같이 Well-form의 형태의 명령 메시지 XML문서를 사용한다.

```
<?xml version="1.0" encoding="euc-kr">
<tubby:request>
    <명령어엘리먼트/>
</tubby:request>
```

형식을 가지며 아래와 같은 명령 엘리먼트가 온다.

* 문서검색
  :<getDocument docName="문서이름"/>
* 문서리스트검색
  :<getDocumentList/>
* 문서삭제
  : <remove docName="문서이름" />
* 구조검색
  : <query xpath="질의어" />

XML-SOAP API는 Admins.wsdl 웹서비스 문서를 통해 문서저장, 문서 리스트검색, 문서삭제, 문서 로드 XML-RPC의 해당 메서드를 각각 호출하며, Query.wsdl 웹서비스 문서를 통해 문서 구조 검색 메서드를 수행한다.

IV. 실험 및 고찰

본 논문에서 제안된 프레임워크의 유효성을 실험하기 위하여, XML문서 저장 및 검색 관리 인터페이스는 아래의 그림에서처럼 클라이언트 지원을 위하여 웹 응용서버의 역할을 하도록 IIS 5.0웹서버상에서 ASP로 구현하였고, XML DB서버는 Aparch Cocoon 2.0 XML Web Server를 사용해 서버단 웹인터페이스를 구성하였으며, 두 웹서버상에서 HTTP API를 가지고 연동되도록 구성하였다.

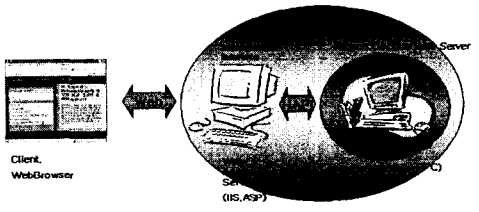


그림 10. 구현 환경
Fig 10. Implementation Environment

4.1 실험

4.1.1 엘리먼트 구조검색

그림 11은 대표적인 엘리먼트 검색의 예이다. "모든 문서에 대하여 요약 엘리먼트를 찾아라" 란 질의에 대하여 논문의 요약 엘리먼트 각각에 대하여 영문 요약과 한글 요약을 찾아주었으며, 질의 처리에 걸린 시간은 0.02초가 걸렸음을 보여주고 있다.

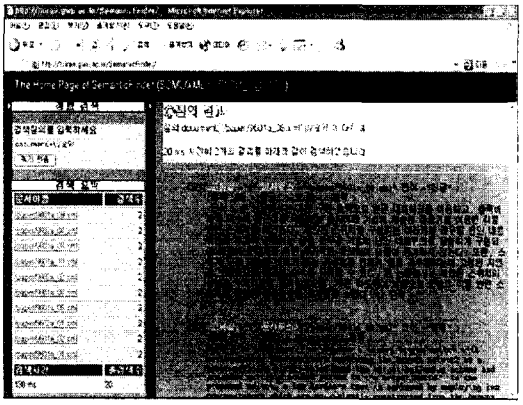


그림 11. 엘리먼트 검색의 예
Fig11. An example of element retrieval

4.1.2 속성 검색

다음 그림 12은 속성 검색의 예이다.

질의 document(*)//요약[@언어="한글"] 정형질의에 대하여 모두 10건의 엘리먼트가 검출되고 그 중 "/paper/9601a_08.xml" 문서를 선택했을 때 해당 엘리먼트의 내용을 윈도우의 오른쪽에 HTML로 변환해서 보여주고 있다.

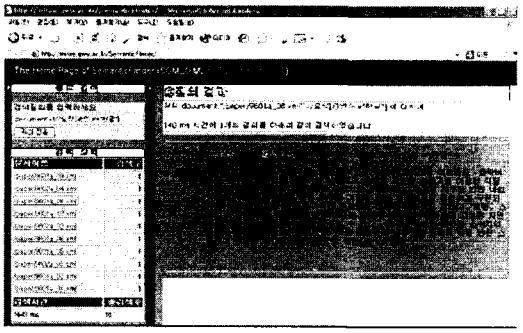


그림 12. 속성 검색의 예

Fig 12. An example of attribute retrieval

4.1.3 구조 및 내용이 포함된 복합 검색

그림13은 복합질의 document(*)//요약[@언어="한글" and 단락 &= '구조'] 로서 "모든 문서에 대하여 요약에 속성이 한글이고 하부 단락에 구조란 단어가 있는 요약 엘리먼트를 찾아라"란 질의에 대하여 수행한 결과를 보여주고 있다.

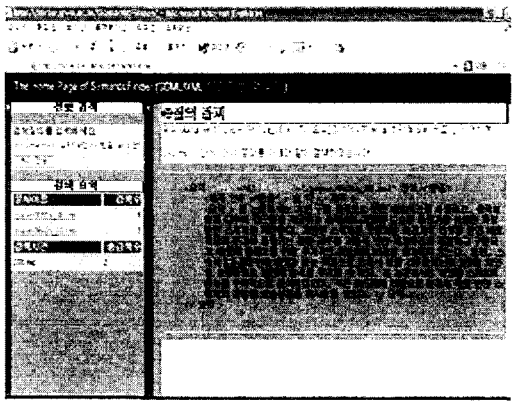


그림 13. 복합 검색의 예
Fig 13. An example of complex retrieval

4.2 고찰

본 논문에서 실험대상으로 사용한 파일은 정보과학회 논문 11편과 세익스피어의 영문 희곡 37권을 대상으로 하였으며, 용량은 정보 과학회 논문이 약 600Kbyte의 텍스트, 세익스피어의 영문 희곡 7.5M 바이트이다.

표1. 논문과 세익스피어 XML문서에 대한 가변 K진트리와 완전 K진트리와의 비교
Table 1. An comparison of variant k-ary complete tree and k-ary complete tree on paper and shakespeare xml collection

문서 종류	문서 의수	평균 문서의 깊이	ID할당비율 (가변K/완전K)
세익스피어	37	6.98	0.070%
논문	11	11.45	0.061%

표2 에서 제시한 것처럼 RDF 메타정보를 이용한 구조검색은 의미가 있는 용어집합으로 이루어진 RDF문서

를 정의함에 따라 일반 XML구조검색기능에 XML엘리먼트에 의해 생성되는 색인과 관리자에 의해 메타데이타를 추가한 색인을 추가로 사용함으로써 XML문서 정보 검색에 좀더 정제된 검색을 할 수 있다.

표2. 검색기의 비교

Table 2. A comparison of Retrieval System

	단순 검색기	XML기반 검색기	RDF메타정보를 사용한검색기
검색 단위	문서	모든 엘리먼트	모든 엘리먼트와 의미 vocabulary 엘리먼트
저장 단위	문서	엘리먼트와 구조정보	엘리먼트와 구조정보
의미 검색	N/A	N/A	의미 vocabulary element
색인 방법	추가 인덱스	모든엘리먼트 인덱스	모든 엘리먼트 인덱스와 의미 vocubulary element
재 사용성	N/A	XML문서	XML문서
표준 지원	N/A	XML	XML,RDF
메타 정보 지원	문서	XML 엘리먼트	XML element, RDF

V. 결 론 및 향후 연구 과제

본 논문에서는 XML문서의 효과적인 저장 모듈, 그리고 검색 모듈과 이를 웹상에서 연동해서 사용할 수 있는 연동 API로 구성된 XML문서 저장 및 검색 프레임워크에 대한 연구를 하였다.

저장 모델에서는 DTD 독립적인 DOM기반 분할기법모델을 적용하고 각 분할요소 즉 노드의 주소기법을 트리 깊이에 따라 가변하는 K진 완전트리의 모델을 사용해서 K진 완전트리의 장점인 DB엑세스를 최소화하면서, 필요한 부모/자식 노드들의 위치를 연산하여 검색해올 수 있었으며, 더불어 K진 완전트리의 많은 NULL노드 생성 및 ID 낭비로 발생하는 비효율을 줄일 수 있었다.

질의모듈로는 XML문서의 구조질의에 적합한 XPath를 문서 집합 단위까지 검색할 수 있게 확장하고,내용 정보에 대하여 단어색인을 제시함으로써 내용정보 구조까지 검색할 수 있는 구조기반 검색 기능을 구현하였다. 그리고 저장시 사용자가 입력한 메타정보를 기반으로 의미 있는 블록 단위로 인덱싱을 하고 이를 이용해 좀더 정확한 메타정보에 의한 검색을 구현하였다.

그리고, XML저장,검색시스템의 웹의 연동을 위하여, XML-RPC 및 HTTP의 GET, POST, PUT 방식의 API와 웹서비스 방식의 SOAP API를 제공함으로써,웹 상에서 XML 저장/검색 시스템을 활용할 수 있는 방식을 구현 예시하였다. 실험데이타로 정보과학회 논문 11편과 세익스피어 희곡 37편 모두 약 8.1M 용량의 전문으로서 제안한 DOM모델에는 약 34만개의 노드로 분할 저장되었으며 완전K진트리와 비교하여 shakespeare의 경우 약0.07%, paper의 경우 0.06%의 ID를 사용함으로써 저장모델이 유효함을 보였고, 구조화 검색에 대하여 빠른 검색을 보였다.

향후 계속 진행하여야 하는 연구로는, 엘리먼트 단위의 수정 알고리즘, 부족한 엔티티/링크정보의 추가하여 멀티미디어 문서를 처리할 수 있도록 구조의 확대에 대한 연구, 차세대 질의언어인 XQuery의 지원에 대한 연구가 필요하다.

감사의 글

본 연구는 2003년도 교내연구비의 지원에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다.

참고문헌

[1] ISO 8879, Information Processing Text and Office System Standard Generalized Markup System(SGML), 1986.

[2] ISO 9069, "SGML Document Interchange Format", 1988.

[3] Dave Raggett, Arnaud le Hors,Ian Jacobs, "HTML 4.01 Specification",World Wide Web Consortium Recommendation, 1999, Available at

http://www.w3.org/TR/html401
[4] W3C, eXtensible Markup Language(XML)1.0 ,http://www.w3.org/1998/REC-xml/19980210.html, Feb,1998.
[5] 이강찬,손홍,박기식,"XML 표준화 동향", 한국정보과학회지,제19권제1호,pp6-14 ,2001
[6] R.G.G. Gattell, Douglas K. Barry, "The object database Standard: ODMG2.0, Morgan Kaufmann Publishers, Inc., 1997
[7] P. Florescu and D. Kossman, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulletin 22(3), pp.27-34, 1999.
[8] S. Malaika, "Using XMLin Relational Database Applications," 15th Int'l Conf. On Data Engineering, Sydney, Australia, p167, 1999.
[9] T. Shimura, M. Yoshikawa, and S. Uemura, "Storage and Retrieval of XML Documents Using Object-Relational Database," DEXA99, pp.206-217, 1999.
[10] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. DeWitt, and J.F. Naughton, "Relational Databases for Querying XML Documents: Limitations and Opportunities, " Proc, of 25th Int'l Conf. on VLDB, Edinburgh, Scotland, UK, pp.302-314, 1999.
[11] 연제원, 조정수, 이강찬, 이규철, "XML 문서 구조 검색을 위한 저장시스템 설계," 한국 정보과학회 봄학술발표 논문집(I), 제 26권 1호 1999.
[12] 한상웅, 홍의경, "ORDBMS를 이용한 XML 문서 저장 시스템 설계", 한국 정보과학회 가을 학술발표 논문집(I), 제 27권 2호, 2000.
[13] 김규태,현득창,이수연,정광철,"관계형 데이터베이스를 이용한 SGML문서 처리",한국정보과학회,제3권 제3호 p238-p247
[14] 김훈, 홍의경, "객체관계형 데이터베이스를 이용한 XML 문서 저장 모델 설계," 한국 정보과학회 가을 학술발표 논문집(I), 제 27권 2호, 2000.
[15] 이용석,손기락,"XML문서 저장 시스템 설계 및 구현", 한국정보과학회 학술 논문집(I),25권 2호, 1998
[16] M. Graves, Designing XML Databases, Prentice-Hall,2001
[17] T.Arnold-Moore, M, Fuller, B. Lowe, J. thom, R. Wilkinson, "The ELF data model and SGQL query language for structured document databases," proceeding 6th australasian Database Conference, 1995
[18] GOMZALO NAVARRO, Richard Baeza -Yates, "Proximal Nodes: A Model to Query Document Databases by Content and Structure," ACM transactions on Information Systems, vol 15. No. 4,Oct,1997,pp.400-435
[19] Toung Dao,"An Indexing Model for Structured documents to support Queries on Content, Structure and Attributes," Proceeding of ADL'98, pp88-97,19
[20] Simon St.Laurent,Joe Johnston,Edd Dumbill, "Programming Web Services with XML-RPC", O'Reilly,20

저자소개

김규태(Gyu-Tae Kim)

1994.2 광운대컴퓨터공학석사
2002.8 광운대컴퓨터공학박사

※관심분야:XML데이터베이스,전자상거래

정회경(Hoe-Kyung Jung)

컴퓨터공학 박사
배재대 정보통신공학부 교수

※관심분야 : SGML, XML, 데이터베이스

이수연(Soo-Youn Lee)

1977 연세대 전자공학 석사
1983 일본교토대 정보공학박사
1973년 - 현재
광운대 컴퓨터공학과 교수

※관심분야 : SGML/XML,XML,전자상거래