

# ORDBMS를 이용한 XML 저장 시스템 설계\*

한상웅<sup>0</sup> 홍의경  
서울시립대학교 전산통계학과  
{swhan92, ekhong}@venus.uos.ac.kr

## Design of XML Document Storage System Using ORDBMS

Sang-Woong Han<sup>0</sup> Eui Kyeong Hong  
Dept. of Computer Science & Statistics, University of Seoul

### 요 약

XML은 최근 인터넷상에서 정보 교환의 표준으로 자리잡고 있다. 따라서, 그동안 semistructured data 와 XML 데이터를 데이터베이스에 저장하고 검색하기 위한 많은 연구들이 진행되어 왔다. 본 논문에서는 기존에 개발된 RDBMS를 이용한 시스템과 OODBMS를 이용한 시스템의 장단점을 수용하여 ORDBMS 상에서 XML 문서를 저장하고 검색할 수 있는 시스템을 설계하였다. 이 시스템은 DTD 독립적인 XML 문서를 저장하기 위해 스키마와 분할저장 방법을 이용하여 임의의 XML 문서를 저장하고 저장된 문서의 갱신을 용이하게 처리할 수 있도록 하였다. 전체적인 시스템은 웹상에서 문서를 저장, 검색할 수 있는 웹 기반 시스템과 XML을 이용하는 응용프로그램의 데이터베이스 연동을 지원하는 응용기반 시스템으로 구분하였다.

### 1. 서론

최근 인터넷이 발전함에 따라 현재 수 많은 양의 정보들이 전자문서의 형태로 존재하고 있고, 앞으로도 전자문서의 사용은 급속도로 증가할 전망이다.

이러한 전자 문서들을 저장하기 위해서는 플랫폼에 독립적이고 문서의 다양한 내용의 표현이 가능하며 구조화된 문서의 전송 및 교환이 용이하도록 해주는 표준이 요구되었다. 이에 따라 W3C(World Wide Web Consortium)는 HTML의 단점을 보완하고 SGML의 복잡성을 제거한 XML(eXtensible Markup Language)을 웹 문서의 표준으로 지정하였다[7].

XML은 이기종간의 정보교환이 용이하며 EC/EDI, 전자도서관, 전자 상거래 등의 다양한 응용분야에서 이용되고 있다. 따라서 XML 문서를 데이터베이스에 효율적으로 저장, 검색하는 방법에 대한 연구가 활발히 진행되고 있다.

본 논문에서는 ORDBMS를 이용하여 XML 문서를 저장하는 방법을 제시하고, 검색 인터페이스에 의해 얻어진 사용자의 질의를 SQL 형태로 변환하는 질의 생성기와 질의에 의하여 얻어진 결과를 XML 형태로 재구성하는 질의 결과 생성기로 구성된 XML 문서 저장 시스템을 설계하였다.

### 2. 관련연구

#### 2.1 저장 시스템에 의한 분류

XML 데이터를 저장하기 위해 기존에 연구된 하부 저장시스템으로는 관계 데이터베이스 시스템, 객체지향 데이터베이스 시스템과 semistructured 데이터베이스를 지원하기 위한 독자적인 저장관리 시스템 등이 있다[1].

관계 데이터베이스 시스템을 이용한 방법은 XML 문서의 저장을 위한 구조정보를 관계 데이터베이스에 테이블 형태로 구성하여 저장한다. 이를 이용한 방식으로는 문서에 대한 DFS(Depth First Search)를 이용하여 구조정보를 표시하는 방법과 계승 엘리먼트 ID를 이용하여 저장하는 방법 등이 있다. 관계 데이터베이스를 이용하는 방법은 RDBMS의 우수한 성능을 이용할 수 있고, 기존의 응용시스템의 데이터를 함께 사용할 수 있는 장점을 가진다. 그러나 검색 시 다수의 테이블에 대한 고비용의 조인(join) 연산을 수행하여야 하며 set-value를 지원하지 않는 등의 단점을 가진다[6].

객체지향 데이터베이스 시스템을 이용한 방법은 데이터베이스에서 지원하는 객체지향 개념을 이용할 수 있기 때문에 상속과 같은 객체지향 특성을 이용할 수 있으며, 엘리먼트 간의 전후종속 관계를 클래스에 기반한 객체들간의 링크로 나타낼 수 있기 때문에 구조적인 문서를 모델링하는데 적합하지만 지금의 객체지향 데이터베이스는 대용량의 데이터에 대한 복잡한 형태의 질의를 처리하는 능력이 성숙되지 않았다는 문제점을 가지고 있다[2].

XML이 semistructured 데이터의 범주에 속하므로 semistructured 데이터를 위한 독자적인 시스템을 확장하여 XML 데이터를 저장하고자 하는 접근 방식이 있다[5]. 그러나 이러한 접근 방법은 20여년간 연구되어 온 데이터베이스 기술을 활용하는데 어려움이 있다[6].

관계 데이터베이스 시스템과 객체지향 데이터베이스 시스템의 장단점을 수용한 객체관계 데이터베이스 시스템을 이용한 경우 계층적 질의를 조인을 이용한 것보다 빠르게 처리할 수 있고 대용량의 데이터에 대한 질의를 안정성 있게 처리할 수 있는 장점을 가진다.

\* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음

2.2 저장 방식에 따른 분류

XML 문서의 구조정보를 데이터베이스에 저장하는 방법으로는 크게 가상분할(virtual fragmentation) 방법과 분할저장(decomposition) 방법이 있다[3].

가상분할 방법은 XML 문서 전체 내용을 하나의 Blob 형태로 데이터베이스에 저장하고 문서 내에 있는 각 단말노드에 대하여 실제 문서상에서 가지고 있는 시작 오프셋과 종료 오프셋을 저장한다. 이에 반하여 분할저장 방법은 XML 문서의 내용을 엘리먼트별로 나누어서 저장하고, 검색 시 구조 정보를 참조하여 해당 엘리먼트나 하위 엘리먼트들의 조합을 생성하여 처리한다.

가상분할 방식은 문서를 한꺼번에 저장하였기 때문에 분할 저장 방법에서 필요한 통합과정이 필요치 않으므로 단말노드에 대한 검색의 경우 위치정보를 이용하여 구조적인 검색이 가능하고 검색 효율이 상대적으로 우수한 반면 XML 문서의 일부분에 대하여 엘리먼트가 추가, 삭제되는 등의 갱신연산이 발생하는 경우에는 문서내의 모든 다른 엘리먼트들의 위치 정보들도 모두 수정해야 하므로 이에 따른 오버헤드와 데이터베이스의 일관성을 유지해야 하는 문제가 있다. 따라서 가상분할 방식은 주로 검색을 위주로 한 시스템에 적합한 방식이라 할 수 있다[4].

분할저장 방식은 문서를 DTD의 엘리먼트 별로 나누어서 저장하므로 엘리먼트의 삽입, 특정 엘리먼트의 수정이나 삭제 등의 갱신연산에 대해서는 관계되는 엘리먼트만을 수정하면 되므로 문서의 편집 및 관리가 쉽고 동일한 내용을 가지는 노드를 공유할 수 있다는 장점을 가진다. 하지만 문서를 검색하여 내용을 추출해야 하는 경우에는 각 엘리먼트의 내용을 조합하여 결과를 구성해야 하므로 검색과정이 복잡하고 검색시간이 많이 걸린다. 또한 DTD에 의존적인 데이터 모델을 사용하는 경우에는 DTD 구조정보가 변경되는 경우에 이와 관련된 데이터베이스 테이블을 모두 재구성해야 하는 문제점이 있다.

3. XML 문서 저장 시스템

전체 시스템은 크게 웹 환경을 지원하는 웹 기반 시스템과 응용프로그램 환경을 지원하는 응용기반 시스템으로 나눌 수 있으며, 여기서는 현재 구현이 많이 진행된 웹 기반 시스템을 중심으로 기술하고자 한다.

3.1 시스템 구조도

본 논문에서 제시하는 XML 저장 시스템의 구조도는 그림 1과 같다. 시스템은 크게 웹 기반 시스템과 응용기반 시스템으로 구성되어 있다. 웹 기반 시스템에서는 웹 브라우저를 이용하여 XML 문서의 저장과 검색을 수행할 수 있도록 하였고, 문서의 수정은 Web 기반 편집기를 이용할 수 있도록 하였다. 응용기반 시스템은 XML을 이용하는 응용프로그램의 데이터베이스 연동을 지원할 수 있도록 하는 시스템으로 XML-QL을 지원하는 응용 프로그램을 이용하여 데이터베이스에 저장된 문서에 대해 검색이 가능하도록 설계 하였다.

- 각 모듈별 기능을 살펴보면 다음과 같다.
- 저장 관리자: XML 문서를 파싱하여 각 객체를 DTD 독립적인 스키마에 적합시켜 데이터베이스에 저장
- 질의 생성기: 질의 인터페이스에 의하여 얻어진 사용자의 질의 요구를 SQL 형태로 변환하는 도구
- 질의결과 생성기: 질의의 결과를 결과 집합 형태로 입력받아 이를 XML 데이터 형태로 재구성하는 도구
- XML-QL to SQL 변환기: 응용프로그램에서 XML-QL 질의어로 질의했을 때 이를 SQL 형태로 변환하는 도구

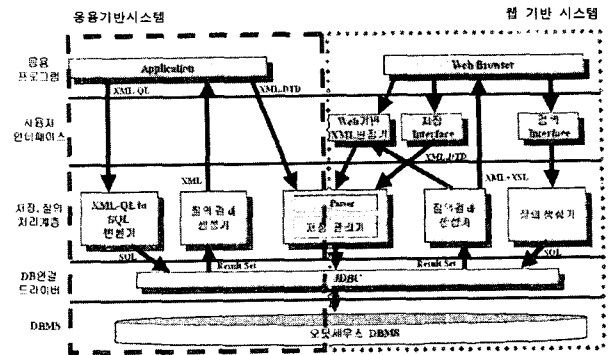


그림 1: XML 저장 시스템 구조도.

3.2 XML 문서 저장 스키마

그림 2는 본 연구에서 이용한 스키마를 OMT 객체도의 형태로 표현한 것이다.

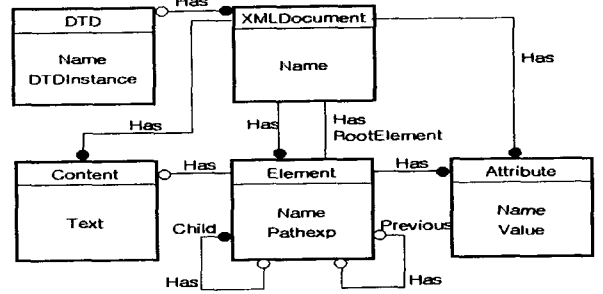


그림 2: XML 문서 저장 스키마.

3.3 XML 문서 저장

XML 문서의 저장은 다음과 같은 과정을 거치게 된다. 저장 인터페이스를 통하여 사용자가 지정한 파일은 서버에 등록되고 등록된 XML 문서는 파서에 의하여 파싱되며, 이는 DOM(Document Object Model) 트리의 형태로 결과가 얻어진다. 파싱의 결과로 얻어진 DOM 트리는 저장관리기에 의하여 테이블에 매핑되어 저장된다.

저장관리기는 문서의 루트노드부터 DFS 방식으로 노드를 순회하며 각 노드의 정보는 해당 테이블에 저장되는데, 노드의 타입이 엘리먼트이면 Element 테이블에 엘리먼트명과 부모 엘리먼트의 OID, 경로등의 정보가 저장되며 애트리뷰트가 있는지를 체크하여 애트리뷰트는 Attribute 테이블에 애트리뷰트나 열린 순서와 함께 애트리뷰트의 이름과 그 값을 저장한다. 이러한 과정을 자식노드를 재귀적으로 순회하면서 모든 엘리먼트를 데이터베이스에 저장한다.

```

<?xml version="1.0"?>
<bookstore>
  <book category="XML">
    <title> Java XML Programming </title>
    <author>
      <firstname> Alexander </firstname>
      <lastname> Nakhimovsky </lastname>
    </author>
    <price> 7.99 </price>
  </book>
  <book category="Database">
    <title> Object-Oriented Database Systems </title>
    <author> Elisa Bertino </author>
    <publisher> Addison-Wesley </publisher>
    <price> 5.99 </price>
  </book>
</bookstore>
    
```

그림 3: 예제 XML 문서.

그림 3과 같은 XML 문서는 파서에 의하여 트리의 형태로 바뀌고, 트리 형태의 XML 데이터는 저장관리를 거쳐 테이블에 그림 4와 같이 저장된다.

DTD 테이블

OID	Name	DTDName	XML Document
o60	bookstore.dtd	book DTD file	{o70..}

XML Document 테이블

OID	Name	DTD	RootElement
o70	bookstore.xml	o60	o1

Element 테이블

OID	Name	Parent	XML Document	CardList	Key	Next	Content	Attribute List	Fulltext
o1	bookstore	Null	o70	{o2,o8}	Null	Null	Null	Null	bookstore
o2	book	o1	o70	{o3,o4,o7}	Null	o8	Null	{o9}	bookstore.book
o3	title	o2	o70	Null	Null	o4	o50	Null	bookstore.book.title
o4	author	o2	o70	{o5,o6}	o3	o7	Null	Null	bookstore.book.author
o5	firstname	o4	o70	Null	Null	o6	o51	Null	bookstore.book.author.firstname
o6	lastname	o4	o70	Null	Null	o5	o52	Null	bookstore.book.author.lastname
o7	price	o2	o70	Null	o4	Null	o53	Null	bookstore.book.price
o8	book	o1	o70	{o9,o10,o11,o12}	o2	Null	Null	{o81}	bookstore.book
o9	title	o8	o70	Null	Null	o10	o54	Null	bookstore.book.title
o10	author	o8	o70	Null	o8	o11	o55	Null	bookstore.book.author
o11	publisher	o8	o70	Null	o10	o12	o56	Null	bookstore.book.publisher
o12	price	o8	o70	Null	o11	Null	o57	Null	bookstore.book.price

Content 테이블

OID	Content	XML Document	Element
o50	Java XML Programming	o70	o3
o51	Alexander	o70	o5
o52	Mathematically	o70	o5
o53	7.99	o70	o7
o54	Object-Oriented Database Systems	o70	o9
o55	Elisa Bertino	o70	o10
o56	Addison-Wesley	o70	o11
o57	5.99	o70	o12

Attribute 테이블

OID	Name	Value	XML Document	Element
o80	XML	o70	o2	o9
o81	Category	Database	o70	o8

그림 4: 저장된 테이블.

3.4 XML문서 검색

전체적인 검색 과정을 살펴보면 검색 인터페이스를 통하여 언어인 사용자의 질의요구는 질의 생성기에 의하여 SQL로 변환되어 질의가 수행되며, 질의의 결과는 결과 집합의 형태로 질의결과 생성기에 넘겨지고 여기서 XML 형태로 결과를 재구성하여 사용자에게 보여지게 된다.

XML 문서에 대한 검색은 크게 내용검색, 구조검색, 애틀리뷰트 검색, 그리고 내용+구조 등의 혼합 검색으로 나눌 수 있다.

다음은 본 논문에서 구현한 시스템이 지원하는 질의 예와 질의 생성기에 의하여 얻어진 SQL-3 형태로 변환된 문장이다.

- 내용 검색
  - 'XML'이라는 단어를 포함하는 문서
    - ⇒ **select deref(XMLDocument).Name from Content where Content like '%XML%'**
- 구조 검색
  - 'book' 엘리먼트의 자식노드로 'publisher'를 포함하는 문서
    - ⇒ **select deref(XMLDocument).Name from Element where Pathexp lik '%book.publisher'**
- 애틀리뷰트 검색
  - 애틀리뷰트 'category'의 값이 'Database'인 엘리먼트
    - ⇒ **select deref(Element).Name from Attribute where Name='category' and Value = 'Database'**
- 내용+ 구조 검색
  - 'book' 엘리먼트의 자식노드로 'publisher'를 포함하면서 그 값이 'Addison-wesley'인 문서
    - ⇒ **select deref(XMLDocument).Name from Element where Pathexp like '%book.publisher' and deref(deref(Element).Content).Content = 'Addison-Wesley'**

4. 테스트

본 시스템은 데이터베이스로 ORDBMS인 오닷세우스를 사용하였고, Java와 JSP를 이용하여 구현하였다. 테스트를 위한 예제 문서로는 세익스피어의 희곡 36편을 이용하였다.

그림 5와 6은 'notorious'란 단어를 포함하는 문서를 검색하는 검색 인터페이스와 질의에 의하여 얻어진 결과를 나타낸 것이다.

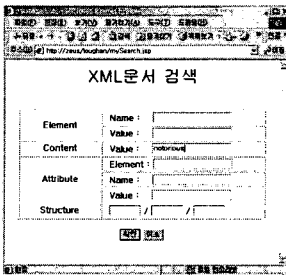


그림 5: 검색 인터페이스.

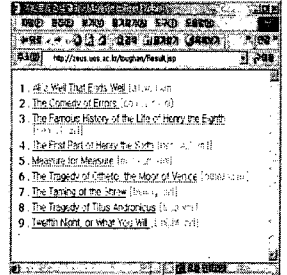


그림 6: 검색 결과.

5. 결론 및 향후 계획

본 연구에서는 XML 문서를 ORDBMS를 이용하여 저장, 검색할 수 있는 시스템을 설계하였다. XML 문서의 저장과 질의를 위하여 저장 관리기 및 질의 처리기, 질의결과 생성기를 주 구성요소로 하는 시스템을 제시하였다.

DTD에 독립적인 XML 문서를 저장하기 위한 데이터베이스 스키마를 이용하여, DOM 트리의 형태로 표현된 데이터를 테이블에 매핑시켜 저장하고, 저장된 데이터를 검색하기 위한 시스템을 시각적인 인터페이스와 함께 구현하고 있다.

Semistructured 데이터에 대한 질의의 특징인 패스 표현의 처리를 위한 효율적인 방법에 대해 현재 연구 중이며, 사용자 정의 타입과 같은 객체 관계적 특성을 활용하여 보다 향상된 데이터 모델 및 스키마 설계를 통한 시스템의 업그레이드를 위해 노력중이다.

향후로는 응용기반 시스템의 구현을 위해 XML-QL 질의를 SQL 형태로 변환하는 XML-QL to SQL 변환기와, 질의의 결과를 XML로 생성해주는 질의결과 생성기를 구현할 계획이다.

6. 참고 문헌

- [1] S. Abiteboul, P. Buneman, and D. Suciu, Data on the Web: From Relations to Semistructured Data and XML, Morgan Kaufmann Publishers, San Francisco, CA, 1999.
- [2] D. Florescu and D. Kossmann, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulletin 22(3), pp.27-34, 1999.
- [3] P. Francois, "Generalized SGML Repositories: Requirements and Modeling," Computer Standard & Interface, 1996.
- [4] S. Malaika, "Using XML in Relational Database Applications," 15th Int'l Conf. on Data Engineering, Sydney, Australia, p.167, 1999.
- [5] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom, "Lore: A Database Management System for Semistructured Data," SIGMOD Record 26(3), pp.54-66, 1997.
- [6] 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.
- [7] W3C, Extensible Markup Language(XML) 1.0, http://www.w3.org/TR/1998/REC-xml19980210.html, Feb.1998