韓國 OA學會 論文誌
第6卷 第1號, 2001. 3.

2001-6-1-3-1

UNISQL/X를 이용한 XML 문서 저장 시스템 설계 및 구현

안 병 태* 김 현 아**

Design and Implementation of XML Documents Storage System using UNISQL/X

Byung-Tae Ahn* Hyun-A Kim**

요 약

최근 인터넷의 발전으로 인하여 정보교환을 위한 표준으로 XML에 대한 연구가 활발히 진행되고 있다. 본 논문에서는 객체관계 데이터베이스인 uniSQL/X를 이용한 XML 문서 저장 시스템 설계하고 구현하였다. 관계 데이터베이스와 객체지향 데이터베이스의 장점을 수용한 객체관계 데이터베이스(Object-Relational Database : ORDB)를 이용하여 XML 문서 저장 모델을 설계하여 XML 문서의 구조적인 정보를 효과적으로 표현할 수 있다. XML 문서의 빈번한 수정이 용이하도록 분할저장 방식을 사용하였고, DTD에 관계없이 XML 문서를 저장할 수 있도록 DTD 독립적인 모델을 제안하였다. 또한 데이터의 중복문제를 해결함으로써 검색 속도가 향상되었다.

Abstract

Nowadays, the study on XML as a standard method of exchange of information is active as Internet thechnology develops. This thesis designs and implements XML documents storage system using uniSQL/X in object-relational database. The system takes advantage of the merits of object-oriented databases, so that the structural information of XML documents can be effectively represented. The system uses split-storage method to allow frequent modifications of XML documents and suggests DTD-independent model so that it can store XML documents without DTD. And retrieval speed is improved by solving the issue of data duplication.

* 마산대학 겸임교수
** 창원전문대학 연구교수
논문 접수 : 2000년 10월 27일 심사 완료 : 2001년 1월 20일

Ⅰ. 서론

최근, 인터넷의 발전은 많은 사용자들로부터 기존 정보의 공유와 호환성, 편의성 등을 필요로 한다. 이러한 요구에 부응하기 위하여 정보 전달의 중요 수단으로 HTML(HyperText Markup Language)이 부각되어왔다. 일반적으로 HTML은 쉽게 웹서비스 문서를 작성할 수 있는 장점을 가지고 있다. 하지만 고정된 하나의 문서 구조만 가질 수 있어 문서 구조 서술의 제약이 많고, 효과적인 문서 검색이 어렵다는 단점을 지니고 있다. 이러한 이유로 1997년 W3C는 HTML의 편리한 사용성과 SGML의 확장성 등의 장점을 취합하여 XML(Extensible Markup Language)을 정의하였다.

이러한 XML 문서를 저장 및 질의할 수 있는 백엔드 데이터베이스 시스템으로는 기존의 RDBMS와 OODBMS 그리고 ORDBMS가 있다.

기존의 상업용 RDBMS를 사용할 경우, RDBMS의 우수한 성능의 이용과 기존의 응용프로그램에서 이용하는 데이터와 XML데이터에 대한 질의가 가능하다는 장점이 있다. 하지만 엘리먼트를 테이블로 표현시 많은 조인을 유발하고 set-valued 속성을 지원하지 않는 제약사항을 갖는다.

한편 OODBMS를 이용할 경우에는 엘리먼트에 대한 클러스터링을 기존의 시스템보다 더 잘할 수 있다. 그러나 OODBMS 자체가 대용량의 데이터에 대한 질의 처리가 성숙되어 있지 않다는 단점이 있다.

RDBMS와 OODBMS의 장점을 수용한 ORDBMS를 백엔드로 이용할 경우, 계층적 질의를 조인을 이용한 것보다 빠르게 처리할 수 있고 대용량의 데이터에 대한 질의를 안정성 있게 처리할 수 있는 장점을 갖는다. 그래서 본 논문에서는 ORDBMS를 기반으로 한 갱신이 자유롭고 탐색이 빠른 XML 문서 저장 시스템을 설계 및 구현하였다.

한편 본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 XML 문서 저장 방식에 대한 연구 배경을 살펴보고, 3장에서는 XML문서 저장을 위한 시스템을 설계하였다. 4장에서는 이러한 시스템 설계를 기반으로 한 XML 문서 저장 시스템을 구현하였다. 끝으로 5장에서는 결론과 향후 연구 과제를 제시한다.

Ⅱ. 관련 연구

1. XML 문서 저장 방법

1.1 관계 데이터베이스 시스템

관계 데이터베이스 시스템을 이용한 방법은 XML 문서의 저장을 위한 구조정보를 관계 데이터베이스에 테이블 형태로 구성하여 저장한다. 이를 이용한 방식으로는 첫째로 LIST 형태를 이용한 방법, 둘째로 경로 엘리먼트 ID를 이용한 방법, 셋째로 DFS Numbering을 이용한 방법, 넷째로 UID를 이용한 방법 등이 있다. 그런데 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 연산을 수행하여야 하며 set-value를 지원하지 않는 등의 단점을 가진다.

1.2 객체지향 데이터베이스 시스템

객체지향형 데이터베이스를 이용한 방법은 관계형 데이터베이스를 사용하는 방법에 비해 XML 문서에 대한 모델링에 적합하며, 엘리먼트들 간의 전후종속 관계를 클래스 구조를 이용하여 쉽게 적용할 수 있다. 그러나 엘리먼트에 대한 접근시 관계형 데이터베이스를 이용할 경우 DFS를 이용한 UID나 계승 엘리먼트 ID값을 키로 하여 접근이 가능하다. 반면에 객체지향형 데이터베이스의 경우 이를 위해 특별한 모델을 제공하지 않을 경우 데이터베이스내의 모든 객체를 탐색하게 되는 경우가 발생하게 된다.

1.3 객체관계형 데이터베이스 시스템

관계 데이터베이스와 객체지향 데이터베이스의 장단점을 수용한 객체관계 데이터베이스를 이용하여 XML 문서 저장 모델을 설계하면 XML 문서의 구조적인 정보를 효과적으로 표현할 수 있도록 모델설계가 가능하다. 또한 계층적 질의를 조인을 이용한 것보다 빠르게 처리할 수 있고 대용량의 데이터에 대한 질의를 안정성 있게 처리할 수 있다.

2. UniSQL/X을 기반으로 한 XML 응용의 기본구조

XML 데이터를 정보 교환의 틀로 사용하는 방식은 두 가지가 있다. 데이터베이스 엔진을 사용해서 XML 데이터를 생성하는 방법과 미들웨어 시스템을 사용해서 XML과 기존의 기반구조 사이의 상호변환을 수행하는 방법이다. 본 논문에서는 미들웨어 시스템을 사용해서 XML과 기존의 기반구조 사이의 상호변환을 수행하는 방법으로 설계하였다.

2.1 데이터베이스 엔진을 사용해서 XML 데이터를 생성하는 방법.

장점 : 속도가 빠르다. 서버에는 미리 만들어진 XML 데이터가 항상 준비된 상태이므로 클라이언트가 요청할 때마다 XML 데이터를 생성할 필요가 없으므로 속도가 빠르다.

단점 : 데이터베이스 엔진으로 XML을 생성하는 접근 방법은 데이터베이스 시스템 안에 저장된 데이터만을 다룰 수 있으며, 표현할 수 있는 데이터의 종류가 어느 정도 한정되어 있다.

2.2 미들웨어 시스템을 사용해서 XML과 기존의 기반구조 사이의 상호변환을 수행하는 방법.

장점 : 여러 시스템들에 대한 접근을 중앙 집중적으로 관리하기 편하다. 데이터에 대한 규칙 기반(rule-based)의 처리를 수행할 수 있다.

단점 : 구축 비용이 높고 특정 소프트웨어 제품에 대한 의존성이 높다. 또한 만일, 데이터가 자주 갱신되지 않는 환경이라면 데이터베이스를 통한 방법보다 비효율적이다.

Ⅲ. 시스템 설계

1. XML 문서 모델 설계

〈그림 1〉은 XML 문서와 DTD 정보를 분리하여 DTD와 독립적인 관계 데이터 모델로 변환하는 사상 기법을 제안한다. 제안된 데이터 모델의 스키마는 모두 6개의 릴레이션으로 구성되어 있으며, DTD와 XML 문서의 모든 정보를 저장하고 관리 가능하도록 설계하였다.

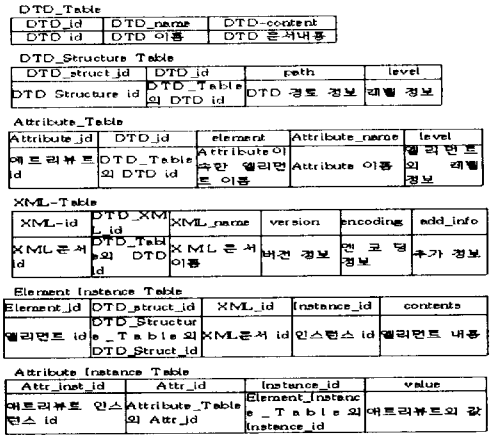

그림 1. XML 저장 스키마 구성

그림 1에서 보듯이 XML 문서를 이루는 모든 DTD는 DTD_Table에 등록되며, 이러한 DTD에서 추출된 구조 정보는 DTD_Structure_Table에 경로 정보와 함께 저장된다. 그리고 이와 관련된 애트리뷰트 정보는 Attribute_Table에 저장된다. XML_Table에는 XML문서의 기본적인 정보가 저장되며, XML문서의 내용정보는 Element_Instance_Table, Attribute_Instance_Table에 저장된다.

2. XML 문서 저장 스키마

DTD와 독립적으로 XML 문서를 저장하고 구조적 검색이 용이하도록 DTD_Table과 DTD_Structure_Table, Attribute_Table을 별도로 구성한다. 이 세 개의 테이블에는 DTD에서 추출할 수 있는 모든 정보가 저장된다. DTD_Table은 저장되는 문서의 모든 DTD에 id를 부여하고, 그 DTD에 따라 작성된 XML 문서를 XML_id를 사용하여 표현한다. DTD_Structure_Table에는 DTD 명세에서 가능한 모든 경로 정보를 엘리먼트별로 구분하여 PATH필드에 저장한다.

경로에 대한 정보는 엘리먼트 단위로 나누어 저장하므로 XML 문서의 갱신 시, 추가적인 정보의 수정없이 해당 엘리먼트와 관련된 부분만 갱신이 이루어지게 되어 문서 내용의 수정이 자유롭다. 또한, 어떠한 DTD 정보를 저장하더라도 스키마에 정의된 6개의 테이블만을 사용하므로 DTD의 갱신이 발생하여도 DTD 관련 테이블의 인스턴스 갱신으로 처리된다. 그밖에도 특정위치에 대한 탐

색 요구나 하위 엘리먼트를 포함하고 있는 엘리먼트에 대한 검색 요구가 발생했을 때, DTD_Structure_Table에 경로 정보를 이용하여 직접 해당 엘리먼트로 접근이 가능해 효율적인 검색이 지원된다. Attribute_Table에는 DTD에 표현된 애트리뷰트를 추출하여 애트리뷰트 이름과 포함 엘리먼트를 저장한다. 그림 2는 본 연구에서 제안한 경로 정보의 표현 방식을 도식화한 것이다.

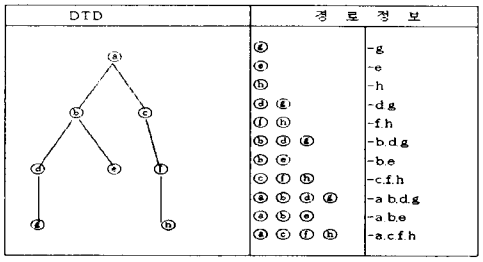

그림 2. DTD에 따른 경로 정보 표현

XML_Table에는 문서 자체에 대한 파일명과 버전정보, XML 문서가 따르는 DTD_XML_id를 저장하고, Element_instance_Table에는 엘리먼트가 속한 XML 문서의 XML_id 필드와 각 엘리먼트의 경로를 나타내기 위해 DTD_struct_id 필드를 사용하여 엘리먼트의 내용을 저장한다. 또한, 반복되는 엘리먼트를 구분하기 위해 동일한 엘리먼트 집합에는 동일한 Instance_id를 부여한다. Attribute_Instance_Table에는 애트리뷰트의 내용정보를 저장한다.

이와 같이 XML 문서의 내용과 DTD의 구조정보를 구분하여 저장함으로써 DTD마다 생성되는 전체 테이블 수를 줄이고, 검색 시 발생하는 조인 연산을 감소시켰다. 또한 자유로운 갱신 연산이 가능하고 경로 정보를 이용한 원문 복원 능력과 검색 성능을 강화하여 데이터 중복을 해결하였다.

Ⅳ. 시스템 구현

본 논문에서 구현한 데이터베이스 시스템은 Sun사에서 개발된 Unix 운영체제인 Solaris 2.5이다. 그리고 DBMS는 객체관계형 데이터베이스인 UniSQL/X Release 3.5.3을 사용하였다. 웹 서버로는 Java Web Server를 사용하였고, 클라이언트 시스템은 윈도우즈 95 이상의 환경에서 웹 브라우저를 통해 접근할 수 있도록 구성되었으며, 사용자에게 편리한 인터페이스를 제공하기 위해 Java Servlet을 이용하여 구성하였다. 그리고 객체관계형 데이터베이스에 손쉬운 접근을 위해서 JDBC를 사용하여 DBMS에 접근하였고, 본 시스템의 전반적인 부분에 이식성이 높은 JDK.1.3을 사용하여 구현하였다.

1. XML 문서 처리기 설계

그림 3은 본 논문에서 사용한 XML 문서 처리기이다. xml4j란 파서를 이용하여 파싱을 하면, valid한 문서인지를 검사한 후 결과물인 전체 트리는 하나의 XML 문서를 표현하고, 각 노드는 하나의 엘리먼트로 구성되는 DOM 트리 형태로 나타나게 된다. 이 결과물에 의해 스키마를 생성하고, UniSQL/X에 저장이 된다.

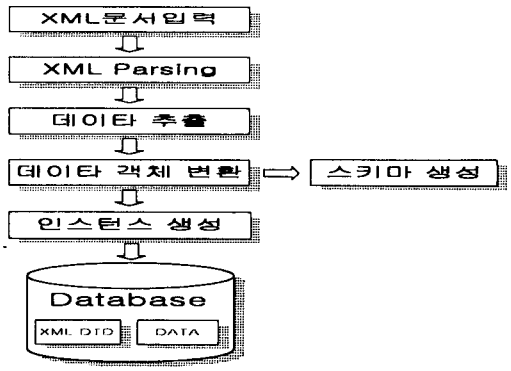

그림 3. XML 문서 처리기 구조

2. XML 문서 관리 시스템 흐름도

그림 4는 본 논문에 대한 전체 XML 문서 관리 시스템의 흐름도이다. 플랫폼 독립적인 Servlet을 사용하여 미들웨어 기능을 제공한다. 효과적인 문서 교환을 지원하기 위하여 클라이언트로부터 보내진 일반적인 질의는 servlet으로 처리되고 XML 문서로 입력되는 데이터는 해당 스키마 및 인스턴스가 생성되어 데이터베이스에 저장된다. XML 문서를 요구하는 질의에 대해서는 질의 결과를 XML 문서로 변환하여 돌려준다.

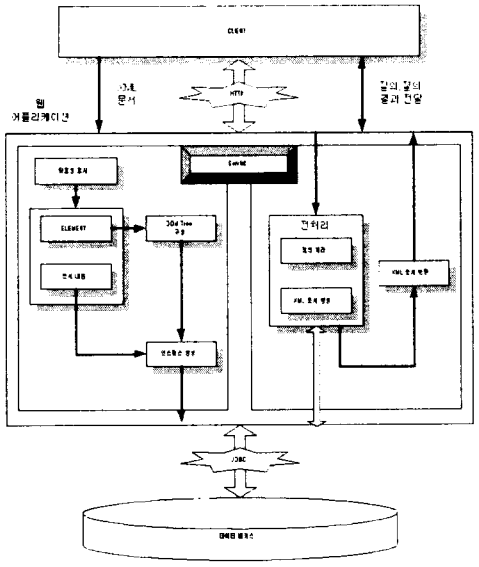

그림 4. XML 문서 관리 시스템 흐름도

3. XML 문서 저장 구현

그림 5의 DTD는 서점의 도서검색을 위한 도서정보로서, book, title, author, price와 관련된 7개의 엘리먼트와 하나의 애트리뷰트로 구성되어 있으며, XML문서에는 이 DTD를 따르는 두 권의 도서정보를 담고 있다.

```
<!ELEMENT bookstore (book)*>
<!ELEMENT book (title,author*,price)>
<!ATTLIST book genre CDATA #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (name | (first-name, last-name))>
<!ELEMENT price (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT frist-name (#PCDATA)>
<!ELEMENT last-name (#PCDATA)>
<?xml version="1.0"?>
<!DOCTYPE bookstore SYSTEM "1.dtd">
<bookstore>
 <book genre="autobiography">
  <title>1234</title>
  <author>
   <first-name>ppp</first-name>
   <last-name>aaa</last-name>
  </author>
  <price>888</price>
 </book>
 <book genre="philosophy">
  <title>5678</title>
  <author>
   <first-name>eee</first-name>
   <last-name>bbb</last-name>
  </author>
  <price>999</price>
 </book>
</bookstore>
```

그림 5. 도서정보 DTD와 XML문서 예

DTD_Table에는 DTD 문서의 id와 그 문서의 이름인 "book.dtd", 그리고 DTD 문서의 내용이 저장된다. 그림 6은 DTD_Table의 저장 정보를 나타낸 것이다.

DTD_id	DTD_name	DTD_content
1	book	-

그림 6. DTD_Table

DTD_Structure_Table에는 "book.dtd"의 리프 엘리먼트인 name, first_name, last_name, title, price로부터 루트 엘리먼트인 bookstore까지의 경로 정보를 상위 엘리먼트에서 하위 엘리먼트 순으로 조합하여 저장한다. 그림 7은 저장된 경로정보를 나타낸다.

DTD_struct_id	DTD_id	path	level
1	1	name	3
2	1	first_name	3
3	1	last_name	3
4	1	title	2
5	1	author.name	2
6	1	author.first_name	2
7	1	author.last_name	2
8	1	price	2
9	1	book.title	1
...	...	...	...
18	1	bookstre.book.price	0

그림 7. DTD_Structure_Table의 저장된 경로 정보

Attribute_Table에는 애트리뷰트를 가지는 엘리먼트 정보가 저장되므로 그림 8의 예를 보면 book과 book의 DTD 구조의 레벨값인 "1"이 Level 필드에 저장되고, Attr_name 필드에는 엘리먼트 book이 가지고 있는 애트리뷰트의 이름 "genre"가 저장된다.

Attr_id	DTD_id	element	Attr_name	level
1	1	book	genre	1

그림 8. Atttibute_Table의 저장 예

XML_Table에는 XML 문서가 따르는 DTD 정보로 DTD_XML_id필드에 DTD_Table의 DTD_id "1"을 저장하고, XML 문서 이름 "book.xml"을 저장한다. 그림 9는 XML_Table을 나타낸 것이다.

XML_id	DTD_XML_id	XML_name	version	encoding	add_info
1	1	book	1.0		

그림 9. XML_Table의 저장 정보

Element_Instance_Table의 DTD_Struct_id 필드는 DTD_Structure_Table에 저장된 경로 정보 리프 엘리먼트에서 해당하는 필드의 id를 저장한다. 그림 10은 Element_Instance_Table의 저장 예로, Element_id가 "4"인 튜플의 경로 정보는 DTD_Structure_Table의 DTD_struct_id "18"번을 참조하여 표현 가능하고, 이 튜플은 XML_Table의 등록된 XML문서 "1"에 포함된 엘리먼트임을 나타내며, 그 내용은 contents 필드의 저장된 "8.99"이다.

Element_id	DTD_struct_id	XML_id	Instance_id	contents
1	4	1	1	The Autobiography of Benjamin Franklin
2	2	1	1	Benjamin
3	3	1	1	Franklin
4	8	1	1	8.99
5	4	1	2	The Gorgias
6	1	1	2	Plato
7	8	1	2	9.99

그림 10. Element_Instance_Table의 저장 정보 예

Attribute_Instance_Table에는 엘리먼트 book의 애트리뷰트 "genre"의 실제 내용인 "autobiography"와 "philosophy"가 저장된다.

그림 11은 Instance_id가 1이고 book 엘리먼트의 "genre" 애트리뷰트 값이 "autobiography"인 튜플과 연관된 엘리먼트의 내용으로 Element_Instance_Table의 Element_id 1,2,3,4를 저장한다.

Attr_inst_id	Attr_id	Instance_id	value
1	1	1	autobiography
2	1	2	philosophy

그림 11. Attribute_Instance_Table의 저장 내용 예

그림 12는 db에 저장된 book.xml을 IE5.0으로 출력 폼을 나타낸 것이다.

```
<?xml version="1.0" ?>
<!DOCTYPE bookstore (View Source for full doctype...)>
- <bookstore>
  - <book genre="autobiography">
      <title>1234</title>
    - <author>
        <first-name>ppp</first-name>
        <last-name>aaa</last-name>
      </author>
      <price>888</price>
    </book>
  - <book genre="philosophy">
      <title>5678</title>
    - <author>
        <first-name>eee</first-name>
        <last-name>bbb</last-name>
      </author>
      <price>999</price>
    </book>
</bookstore>
```

그림 12. XML 문서 출력 폼

V. 결론 및 향후 연구과제

본 논문에서는 XML 문서의 갱신이 비교적 자유로운 분할 저장 방법을 기반으로, DTD 독립적인 객체 관계형 스키마를 정의하여 테이블 생성수와 갱신의 문제점을 해결하고, DTD의 리프 노드에서 상위 엘리먼트까지의 경로 정보를 인스턴스로 갖는 DTD 구조 테이블을 제안하여 탐색 및 원문 복원 능력을 강화하였으며, 데이터 중복을 해결할 수 있는 데이터 모델을 제시하여 분할 저장 방법의 검색 비효율성을 해결하였다.

그리고 데이터 관한 정보를 XML문서 형태로 표현하고 DTD와 데이터 스키마를 이용하여 DB에 저장하는 XML문서 처리기를 개발하였다.

향후 연구 과제로는, DB에 저장된 데이터를 XML 문서로 나타내는 XML 문서 생성기를 구현하고 본 설계를 이용한 정보 검색 시스템을 구현 및 평가하여 객체 관계 데이터베이스를 이용하여 효과적으로 XML문서를 저장하는 것이다.

참고 문헌

[1] Hiroshi Maruyama, Kent Tamura, Naohiko "XML and Java", Addison Wesley, 1999
[2] Didier Martin, Mark Birbeck, Michael Kay, Brian Loesgen, "Professional XML", WROX Press, 2000
[3] Benoit Marchal "XML by Example", Que, 1999
[4] Bradley D. Brown, Brad Brown "Oracle8i Web Development", McGraw-Hill, 1999

저 자 소 개

안 병 태
1999년 : 국민대학교 전산과학과(학사)
2001년 : 경남대학교 컴퓨터공학(석사)
현 재 : 마산대학, 창신대학, 창원전문대학, 진주보건대학 외래교수.
관심분야 : 데이터베이스, XML, 웹 & DB 연동, 멀티미디어.

김 현 아
1992년 : 경남대학교 전산통계학(학사)
1998년 : 경남대학교 전자계산교육(석사)
현 재 : 창원전문대학 연구교수
관심분야 : 전자계산교육 방법론, XML, 멀티미디어, 데이터베이스