

MMDB를 이용한 효율적인 XML 처리 시스템

정문권¹, 이경희², 조완섭³
충북대학교 정보산업공학과¹, 충북대학교 전자계산학과², 충북대학교 경영정보학과³
munkweon@orgio.net¹, khlee@trut.chungbuk.ac.kr², wscho@trut.chungbuk.ac.kr³

An Efficient XML Data Management System Using an MMDB

Mun-Kweon Jeong¹, Kyung-Hee Lee², Wan-Sup Cho³
Dept. of Information Industrial Engineering, Chungbuk National University¹
Dept. of Computer Science, Chungbuk National University²
Dept. of Management Information Systems, Chungbuk National University³

요 약

인터넷을 통한 응용 프로그램의 통합과 다양한 정보의 등장으로 인해 정보를 원활히 교환하고 관리 저장할 수 있는 정보 시스템의 요구가 급증하고 있다. 이에 대한 해결책으로 등장한 XML은 기존의 데이터베이스 관리 시스템과 연동하여 다양한 플랫폼과 이질의 데이터에 관계없이 저장과 관리가 용이하며, 데이터 교환을 최적화 할 수 있다. 본 논문에서는 XML 문서의 논리적인 구조정보를 표현하는 DTD와 Database Schema 간의 변환구조를 MMDB에 저장함으로써 XML DTD를 기반으로 Database Schema를 설계하고 XML DTD의 변환 구조를 효율적으로 관리하는 방안을 제시한다. 이렇게 함으로써 빈번하게 사용되는 XML DTD 변환 정보를 메모리에 상주하게 되고, 그 결과 디스크 출입회수를 줄이게 되어 XML을 사용하는 응용에서 XML Data에 대한 처리 효율성을 높일 수 있게 된다.

1. 서 론

W3C에서 XML 표준이 발표된 이후 인터넷의 발전과 더불어 다양한 정보의 확산으로 인해 XML은 인터넷 기반의 다양한 분야에서 데이터 교환을 위한 표준으로 확산되며 활발한 연구가 진행중이다. 새로운 영역의 e-business가 늘어나면서 전자상거래(EC), 고객관계관리(CRM), 전사적 자원관리(ERP), 전자문서교환(EDI), 지식관리시스템(KMS) 등에 이르기 까지 경쟁력 확보를 위해 새로운 e-business 시스템 구축이 요구되며, 이를 효율적으로 수행하기 위해 XML의 지원이 필수적인 요소로 되고 있다.

XML은 문서에 대한 구조정보를 제공하며, XML 태그는 데이터를 해석하는데 사용할 수 있다. 문서의 구조정보를 내포하고 있는 데이터로써의 XML 문서를 효과적으로 관리하는 구조와 질의어 설계 및 처리에 대한 연구가 활발히 진행되고 있다.

XML 문서를 저장 관리하고 질의하는 방법으로는, Lorel[3]이나 STORED[3,4]와 같이 기존의 XML 스키마에 의존하지 않고 반구조적 데이터에 대한 저장과 질의 처리기법을 개발하는 기법과, XML Schema, DTD 등의 XML 스키마를 기반으로 저장구조를 설계하는 기법이 있다. 이 중 DTD는 이기종간의 문서교환에서 중요한 역할을 하기 때문에 DTD 기반의 XML 문서 관리 시스템에 관한 연구가 중요 시 되고 있다[1].

XML DTD(Document Type Definition)란 특정 형의 문서를 마크업하기 위하여 XML을 적용하는 규칙들이다. 문서 형 정의는 마크업으로 표현될 수 있는 엘리먼트 형, 엘리먼트 상호 관계와 속성, 참조에 관한 규정을 포함하며 문서 형 선언으로 표현된다. 본 논문에서는 XML 데이터 처리의 성능을 향상시키기 위하여 MMDBMS(Main-Memory DBMS)를 도입한다. XML 자료중에서 빈번하게 사용되는 DTD 변환정보를 디스크 데이터베이스에 상주시키는 대신에 메인 메모리에 상주시킴으로써 디스크 출입 회수를 줄여 XML Data 처리 성능을 향상시

킨다. 이를 위하여 본 논문에서는 XML 문서의 DTD 변환 정보를 구분하여 MMDB에 저장하며, 이를 위한 방안들을 제시한다.

본 논문의 구조는 다음과 같다. 제 2 장에서는 관련 연구를 살펴본다. 제 3 장에서는 MMDB를 이용한 XML 처리 시스템의 구조를 설명하고, 제 4 장에서는 제안된 시스템을 이용한 XML 질의 처리에 대하여 설명하고, 제 5 장에서는 결론을 맺는다.

2. 관련연구

XML 문서를 데이터베이스에 저장관리하는 시스템에서는 무엇보다도 데이터 모델링이 중요하다. 데이터베이스를 이용하여 XML문서를 데이터 모델링 하는 방법은 첫째, 기존의 데이터베이스 시스템을 이용하여 XML 데이터를 저장하고 질의처리하는 방식과, 둘째 Tamino와 Excelon[5,6]과 같이 XML을 가장 잘 표현할 수 있는 방식으로 XML을 지원하고 그에 맞는 질의 처리를 가능하게 하는 데이터베이스시스템을 개발하는 방식이다. 이러한 방식에서 가장 중요한 문제는 XML의 문서 구조와 데이터베이스의 구조를 매핑하고 데이터를 변환할 수 있는 최적의 방안을 찾는 것이다. XML을 효율적으로 저장 관리하기 위해서는 XML의 객체관리를 최적화할 수 있는 데이터 모델링이 매우 중요하다.

지금까지 연구된 XML 문서 처리는 DBMS에 따라 관계형 모델과 객체지향 모델로 분류된다. 가장 많이 사용하고 널리 상용화된 관계형 데이터베이스 모델은 질의 처리 성능이 우수하고 대용량 데이터베이스의 안정성이 우수하며, 사용자들이 쉽게 사용할 수 있으나 XML문서의 구조를 표현하기에는 테이블과 튜플의 수가 너무 증가함으로 인하여 검색 시 다수의 테이블에 대한 고비용의 조인 연산이 수행되어 시스템의 성능이

우려되는 단점을 가진다[10]. 그러나 객체지향 데이터베이스 모델은 XML의 EID를 객체지향의 상속과 같은 객체지향 속성을 이용하여 XML의 구조정보를 표현하는데 복잡한 데이터구조로 표현이 가능하다는 장점이 있다[2]. XML문서의 저장시스템은 XML 문서의 특성을 고려하여 복잡하고 다양한 구조를 효율적으로 저장할 수 있어야 한다. 또한 XML문서의 구조정보를 이용하여 효율적으로 저장관리 및 질의처리가 가능하도록 해야 한다.

3. XML 처리 저장 시스템

XML 문서는 XML Document, DTD, Element, Attribute, Entity, Comment 등의 여러 요소로 구성된다. XML Document는 문서 전체를 표현하는 객체이며 XML DTD는 문서의 논리적인 구조를 정의하는 객체로 XML 문서구조에 대한 기술을 공유할 수 있으며, XML 문서의 특정 인스턴스들의 유효성을 검증할 수 있다. 문법에 맞는 문서는 XML 문서에서 사용된 요소들이 모두 시작 태그와 끝 태그를 가지고 있고, XML 문서에서 요구하는 중첩 규칙을 위반하지 않은 문서를 의미하는 것으로, DTD가 없는 XML 문서를 비검증용(non-validating parser) 파서로 파싱했을 때 오류가 없는 문서를 말한다. 이에 비해 유효한 문서는 XML 문서가 DTD의 정의대로 올바르게 작성되었는지 검사하는 유효성 검증(validating)과정을 거치게 되는데, DTD가 있는 XML 문서를 검증용 파서(validating parser)를 사용하여 파싱했을 때 오류가 없는 문서를 의미하고 있다.

본 논문에서는 XML Document와 DTD 객체 사이의 관계를 설정하여 DTD 변환정보를 MMDB에 저장하게 된다. DTD의 논리적 구조를 따르는 XML문서의 구조정보는 엘리먼트를 기반으로 하므로 이러한 XML 문서상의 엘리먼트를 유일하게 구별하고 엘리먼트간의 계층정보를 표현할 수 있는 EID(Element ID)를 부여한다[8]. MMDB를 사용하여 XML DTD 변환정보를 저장하는 시스템에서는 엘리먼트와 EID의 사상을 위한 매핑 테이블을 MMDB에 저장시키게 된다. MMDB를 이용하여 XML DTD 변환정보를 저장하는 시스템은 그림 1과 같다.

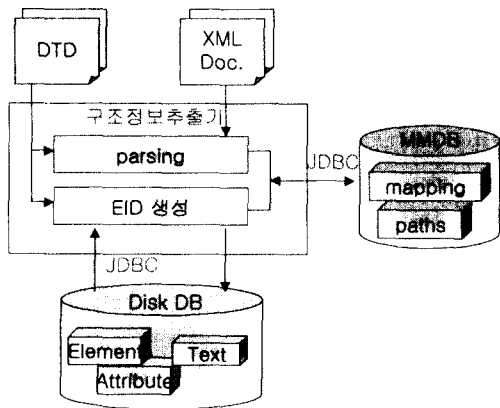


그림 1 MMDB를 이용한 XML 처리 저장 시스템
DTD의 논리적구조를 따르는 XML문서의 구조 정보는 엘리먼트를 기반으로 한다. 그림 1의 MMDB를 이용한 XML 처리

장 시스템에서 구조 정보 추출기는 XML 문서와 DTD 파일과 XML문서상의 엘리먼트를 받아 저장하는 역할을 하며, 동시에 DTD 변환정보를 생성하기 위한 DTD 사상의 객체와 문서 인스턴스를 위한 문서 객체를 생성하고 문서 인스턴스 내에 정의되어 있는 구조 정보들을 추출하여 데이터베이스 스키마와 DTD 구조정보를 비교하여 MMDB에 매핑 변환정보를 생성한다. 그림 1에서 Disk DB에 저장된 정보를 MMDB에 이동시키고 Disk DB를 없애면 MMDBMS를 이용한 XML 저장 시스템이 되며, 이러한 구조도 향후 연구할 예정이다. 그림 1의 시스템 구조에서 Disk DBMS로서는 Oracle 8i를 사용하고, MMDBMS로는 한국전자통신연구원에서 제작한 Tachyon[11]을 사용하고 있다.

```

<?xml version="1.0" encoding="euc-kr"?>
<!DOCTYPE lecture [
  <!ELEMENT lecture(title, time, prof, classroom, book)>
  <!ATTLIST lecture type (computer|management)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT time (#PCDATA)>
  <!ELEMENT prof (name, degree)>
  <!ELEMENT name (first, last)>
  <!ELEMENT first (#PCDATA)>
  <!ELEMENT last (#PCDATA)>
  <!ELEMENT degree (#PCDATA)>
  <!ELEMENT classroom (#PCDATA)>
  <!ELEMENT book (title, author, editor, price)>
  <!ELEMENT title
<lecture type="computer">
  <title>XML</title>
  <time>3hour</time>
  <prof>
    <name>
      <first>ws</first>
      <last>cho</last>
    </name>
  <classroom>H943</classroom>
  <book>
    <title>XML Bible</title>
    <author>HAROLD</author>
    <editor>IDG Books</editor>
    <price>$50</price>
  </book>
</lecture>
    
```

그림 2 XML 문서 예제

MMDB를 이용한 XML 처리 저장 시스템을 바탕으로 그림 2와 같은 XML 문서와 DTD, 그리고 XML 문서의 인스턴스가 구조 정보 추출기를 통해 들어오면 먼저 파싱 정보와 함께 XML 문서를 위한 정보를 생성할 때 부여받은 파싱 ID를 통해 해당 문서를 파싱하면서 내부에 정의된 엘리먼트들을 추출하고 EID를 부여하여, DTD 매핑 정보를 이용하여 각각의 관계를 링크로 연결하게 된다. XML 문서의 DTD를 파싱하여 구조 정보 추출기를 통해 데이터베이스 스키마를 생성한다. 이 때 그

림 3과 같은 DTD와 데이터베이스 스키마간의 매핑 정보를 MMDB에 저장한다.

path		element	
pathinfo	pathID	docID	pathID
/lecture	1	1	1
/lecture/title	2	1	2
/lecture/professor	3	1	3
/lecture/professor/name	4	1	4
/lecture/professor/name/first	5	1	5
/lecture/professor/name/last	6	1	6
/lecture/professor/degree	7	1	7
/lecture/class_room	8	1	8
/lecture/book	9	1	9
/lecture/book/title	10	1	10
/lecture/book/author	11	1	11
/lecture/book/editor	12	1	12
/lecture/book/price	13	1	13

text		
docID	pathID	value
1	2	XML programin
1	8	H-943
1	3	wscho
1	7	professor
1	6	cho
1	10	XML Bible
1	11	Harold
1	12	IDG BOOKS
1	13	\$50

그림 3 DTD 매핑 정보

4. XML 질의 처리기

데이터베이스에 저장된 XML 정보를 추출하기 위한 질의어로서 XQL을 사용한다. XQL은 1998년 w3c에서 제안한 XML 문서를 위해서 특별히 설계된 질의어이다[10]. 응용으로부터 XQL 질의어가 들어오면 그림 4에서와 같이 질의처리 분석기에서 원하는 데이터정보를 추출하여 SQL로의 질의변환을 수행한다. 이렇게 질의 변환된 SQL을 이용하여 DiskDB에 SQL 질의를 수행하고 검색된 결과를 다시 XML Data로 변환하여 응용으로 전송한다.

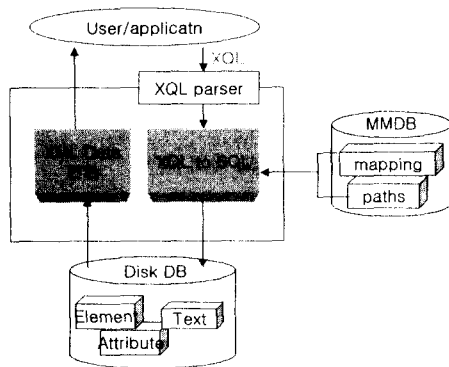


그림 4 MMDB를 이용한 XQL 처리 시스템

5. 결론

본 논문에서는 XML 문서의 논리적인 구조정보를 표현하는 DTD와 Database Schema간의 변환구조를 MMDB에 저장함으로써 XML DTD를 기반으로 Database Schema를 설계하고 XML DTD의 변환 구조를 효율적으로 관리하는 방안을 제시하였다.

XML DTD 변환 구조를 MMDB에 저장함에 따라 XML을 사용하는 응용에서 XML Data에 대한 처리를 더욱 효율적으로 이루어 질 수 있도록 설계하고 이를 이용하여 질의 처리 할 수 있다.

향후 연구과제로는 MMDB를 이용한 XML 문서 저장 시스템의 문서를 효율적으로 관리하고 검색할 수 있는 검색 알고리즘을 연구하고 XML Data의 삽입, 삭제, 갱신의 경우에 관한 연구와 질의 처리 측면에서의 성능 평가를 수행할 예정이다.

참고 문헌

- [1] J. Shanmugasundaram, E. Shekita, R. Barr, M. Carey, B. Lindsay, H. Pirahesh, B. Reinwald, "Efficiently Publishing Relational Data as XML Documents", VLDB Conference, September 2000.
- [2] Takeyuki Shimura, Masatoshi Yoshikawa and Shunsuke Uemura, "Storage and Retrieval of XML Documents using Object-Relational Databases, DEXA 1999, pp.206~217
- [3] Roy Goldman, Jason McHugh, Jenifer Widom, "From Semistructured Data to XML: Mifraing the Lore Data Model and Query language", WebDB, 1999
- [4] Alin Deutsch, M.Fernandex, D.Suciu, "Storing Semi-structured Data with STORED," Proceedings of ACM SIGMOD Conference. Philadelphia, Pennsylvania, May 1999
- [5] <http://www.softwareag.com/developer/>
- [6] <http://www.exceloncorp.com/>
- [7] 이종설, 박종관, 정연수, 손충범, 강형일, 유재수, 김동욱, 최한석, "구조정보 검색을 위한 XML 저장관리 시스템 설계 및 구현", 한국정보과학회 '99 가을 학술발표논문집 (I), 한국정보과학회, 1999. 10, pp. 36~38.
- [8] 민영수, 강승현, 강형일, 유재수, 이하욱, 최한석, "XML문서를 위한 구조정보 추출기의 설계 및 구현", 한국정보과학회 '99 가을 학술발표논문집 (I), 한국정보과학회, 1999. 10, pp. 81~83.
- [9] 박은경, 정채영, 김현주, 배종민, "XML DTD로부터 관계형 테이블로의 사상 구조 설계", 한국정보과학회 '2001 춘계학술발표논문집 2001. 6, pp.133~135
- [10] XML Query Lange(XQL), "<http://www.w3.org/Trands/QL/QL98/pp/xql.html>"
- [11] Tachyon "<http://tachyon.etri.re.kr>"