

객체지향형 데이터베이스를 이용한 XML 문서 저장 시스템 설계

*김영일, *신동욱, *권택근, **김형선

*충남대학교 컴퓨터공학과 정보검색 연구실

**한국전자통신연구원, 컴퓨터 소프트웨어 기술연구소, 전자상거래연구부

Design of an XML Document Storage System using Object Oriented Database

*Youngil Kim, *Dongwook Shin, *Taekgeun Kwon, **Hyungsun Kim

*Department of Computer Engineering, Chungnam National University

**Electronic Commerce Department, ETRI-Computer&Software Technology Laboratory

요 약

최근 인터넷을 통한 정보 교환을 위해 XML(eXtensible Markup Language)에 대한 저장 및 검색에 대한 연구가 활발히 진행되고 있다. 본 연구에서는 객체지향형 데이터베이스를 이용하여 대량의 XML 문서에 대한 저장 및 검색을 지원하는 XML 문서 저장 시스템을 설계하였다. 제안하는 데이터 모델은 XML 문서의 삽입 및 갱신이 용이하도록 분할 방식을 사용하였으며, 객체지향형 데이터베이스에서 구조 정보를 추출하기 위한 새로운 모델을 제시하고 있다. XML 문서의 주된 구조정보를 갖는 엘리먼트와 에트리뷰트를 DTD별로 저장하고, 하나의 DTD를 따르는 문서 인스턴스들에 대한 관계물 리스트를 이용하여 저장해 둬서 객체지향형 데이터베이스 내에서 임의의 위치에 존재하는 객체에 대한 접근을 빠르게 지원할 수 있도록 설계하였다.

1. 서론

최근 인터넷의 발전과 전자 문서의 증가로 인하여 인터넷상의 문서들이 다양한 형태의 정보들을 제공하고 있다. 따라서 정보검색에 관련된 주요한 요구사항으로 기존의 문서들을 다양한 방식으로 관리할 수 있는 기능이 요구되고 있다. 인터넷상의 문서들을 저장하기 위해 플랫폼에 독립적이고 문서에 대한 정보의 전송과 교환이 편리하며 문서의 다양한 내용들을 나타낼 수 있는 방식이 요구된다. 이러한 방식을 위해 표준화된 것으로 SGML(Standard Generalized Markup Language), XML(eXtensible Markup Language)이 있다. SGML/XML 문서들의 특징은 문서를 구조적으로 표현하여 저장, 검색, 관리할 수 있도록 하는 것이다.

XML문서는 기존의 인터넷 문서인 HTML의 단점과 기존의 구조화 문서인 SGML의 단점들을 해결하여 새로운 인터넷 문서의 표준으로 만들어진 문서이다[1]. XML문서가 인터넷에서 사용될 것이며 이를 검색, 관리하기 위한 부분들을 필요로 하게될 것이다.

따라서 XML 문서를 저장 검색하기 위해서 XML 문서의 구조를 잘 표현하고 검색할 수 있는 시스템이 필요하게 된다. 최근 들어 XML 문서의 검색을 위하여 엘리먼트와 에트리뷰

트에 대한 검색을 지원하는 XML-QL, XQL 등의 질의 언어가 제안되고 있다[2][3]. 본 연구에서는 XML 문서의 구조정보 저장을 위한 다양한 저장 시스템들을 살펴보고 효율적인 저장모델을 설계하였다. 본 연구에서 제안된 XML 문서 저장 시스템은 XML 문서에서 가장 일반적인 구조정보를 갖는 엘리먼트와 에트리뷰트를 효과적으로 검색할 수 있도록 객체지향형 데이터베이스를 이용하여 설계하였다.

2. 관련 연구

2.1 일반적인 구조 문서 저장 모델

문서를 저장 관리하기 위한 방법으로 관계형 데이터베이스나 객체지향형 데이터베이스를 사용하는 방법이 있다. 이러한 모델을 분석해 보면 다음과 같다.

관계형 데이터베이스를 이용한 방법은 XML 문서의 저장을 위한 구조정보를 관계형 데이터베이스에 테이블 형태로 구성하여 사용을 한다. 이를 이용한 방식으로는 문서에 대한 DFS(Depth First Search)를 이용하여 구조정보를 표시하는 방법과 계층 엘리먼트 ID를 이용하여 저장하는 방법등이 있다[7][8]. 그러나 이러한 관계형 모델을 이용한 방법은 SQL을 이용한 검색시 다수의 테이블에 대한 고비용의 조인(join) 연

산이 필요한 단점이 있다[5].

객체지향형 데이터베이스를 이용한 방법은 관계형 데이터베이스를 사용하는 방법에 비해 XML문서에 대한 모델링에 적합하며, 엘리먼트들 간의 전후종속 관계를 클래스 구조를 이용하여 쉽게 적용할 수 있다는 장점이 있다[6].

XML 문서의 구조정보를 저장하기 위한 방식으로는 문서를 엘리먼트 단위로 쪼개어 저장하는 분할(Decomposition) 방법과 문서를 실제로 쪼개어 저장하지 않고 가상적으로 분할하여 문서 내에서의 위치정보만을 저장하는 방법인 가상분할(Virtual Fragmentation) 방법이 있다. 가상분할을 사용할 경우 문서에 엘리먼트를 추가, 또는 삭제할 경우 다른 모든 엘리먼트들의 위치정보도 수정해 주어야 하는 단점이 있다. 이에 반해 분할 방법을 사용할 경우 엘리먼트의 전, 후, 부모, 자식 엘리먼트에 대한 정보만 수정해 주면 되나 검색 시간이 오래 걸리는 단점이 있다[9].

2.2 구조정보 검색을 위한 저장 모델의 요구사항

XML 문서의 경우 엘리먼트와 에트리뷰트를 통해 대부분의 구조 정보를 얻을 수 있다. XML 질의어는 엘리먼트를 이용한 문서 계층구조에 대한 질의, 엘리먼트가 갖는 속성에 대한 질의 등을 주된 질의어로 갖는다[2][3]. 이러한 질의어를 만족시키기 위해 저장 모델은 문서내의 임의의 위치에 존재하는 엘리먼트에 대한 접근 능력을 제공해야 한다.

관계형 데이터베이스를 이용할 경우 DFS를 이용한 UID나 계층 엘리먼트 ID값을 키로 하여 접근이 가능하다. 그러나 객체지향형 데이터베이스의 경우 이를 위해 특별한 모델을 제공하지 않을 경우 데이터베이스내의 모든 객체를 탐색하게 되는 경우가 발생하게 된다. 이러한 문제점을 극복하기 위해 김현기의 DocsMaster/RM의 경우 XML 문서의 저장 부분을 O2를 이용하여 처리하고, 데이터베이스에 트리형태로 저장된 객체들에 대한 내용 및 필드 검색을 지원하기 위한 부분으로 STEER-SGML/XML을 연동하여 사용하였다[10]. 그러나 데이터베이스와 파일시스템을 연동하여 사용하는 경우 검색효율은 증가하나 두 개의 시스템을 동시에 개발하여야 되며, 두 시스템내의 일관성을 유지하기가 어렵게 된다. 또한 O2 DBMS의 경우에는 External ID로서 OID를 지원하나, Object Store DBMS의 경우 외부에서 참조 가능한 OID를 지원하지 않으므로 OID를 이용한 파일시스템과의 연동은 특정 객체지향형 데이터베이스에 한하여 적용할 수 있는 모델이 된다.

본 연구에서는 객체지향형 데이터베이스에서 문서내의 임의의 위치에 존재하는 엘리먼트 및 에트리뷰트를 접근할 수 있는 저장모델을 설계하였다. 데이터베이스만을 이용하여 내용 및 구조 검색이 가능하도록 문서의 구조정보를 저장하기 위한 스키마를 설계하였으며 문서의 삽입, 삭제가 용이하도록 하기 위해 문서를 엘리먼트 단위로 쪼개어 저장하는 분할 방법을 사용하였다.

3. XML 문서 저장 시스템의 설계

3.1 XML 문서 스키마

XML 문서는 크게 문서의 전체적인 구조를 나타내는 DTD(Document Type Definition)와 이러한 DTD를 따르는 문서 인스턴트로 구별할 수 있다. 이러한 특성을 이용하여 XML 문서를 저장하기 위한 스키마를 두 부분으로 분류하였다. 하나의 DTD를 따르는 문서들의 구조정보를 표현하기 위하여 DTD 객체와 DTDElement 객체를 이용하여 문서들의 구조정보를 트리형태로 저장하고, 각각의 문서 인스턴스를 저장하기 위하여 Document 객체와 Element, Attribute 객체를 이용하여 트리형태로 저장하게 된다. 설계된 XML 문서 저장 스키마는 그림 1과 같다.

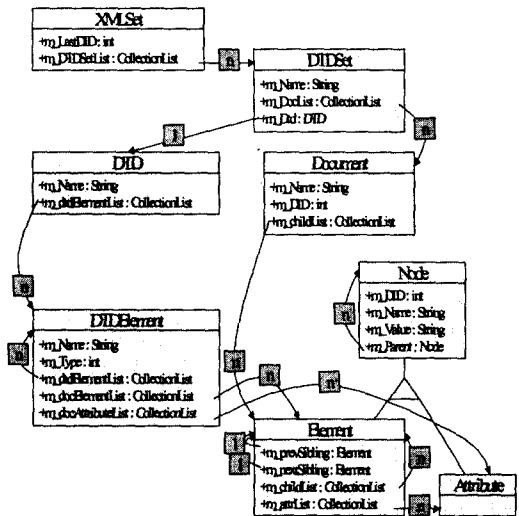


그림 1. XML 문서 저장 스키마

저장 스키마에 나타나는 객체들은 다음과 같은 특성을 갖는다.

- XMLSet : 전체 XML 문서 집합을 관리한다.
- DTDSet : 하나의 DTD를 따르는 문서들을 관리한다.
- DTD : DTD를 따르는 모든 문서 인스턴스에서 나타나는 엘리먼트에 대한 구조 정보를 저장한다. DTD내에서 최상위에 존재하는 엘리먼트들에 대한 리스트를 갖고 있다.
- DTDElement : 문서 인스턴트 내에서 해당 위치에 존재하는 엘리먼트들에 대한 리스트와 해당 위치에 존재하는 에트리뷰트 리스트, 자손으로 존재하는 DTDElement 리스트를 갖는다. DTDElement로 이루어진 트리구조를 탐색함으로써 특정 구조를 갖는 객체를 검색할 수 있도록 지원한다.
- Document : 하나의 문서 인스턴스를 나타낸다. 문서 인스턴트의 최상위 엘리먼트에 대한 리스트와 문서번호를 갖는다.

- Node : 엘리먼트와 에트리뷰트의 상위 클래스. 엘리먼트 이름 또는 에트리뷰트 이름, 엘리먼트 내용 또는 에트리뷰트 값을 갖는다.
- Element : 왼쪽/오른쪽 형제, 자손 엘리먼트 리스트와 엘리먼트에 속한 에트리뷰트 리스트를 갖는다.

3.2 XML 문서 스키마

XML 문서를 저장하기 위하여 다음과 같은 과정을 거친다. 우선 저장하고자 하는 XML 문서를 파싱하여 DOM(Document Object Model) 형태로 결과를 얻어 온다[4]. 그리고 파싱 결과로 얻어진 DOM을 이용하여 데이터베이스에 Document 객체를 생성하고 Document.m_childList를 이용하여 문서에서 나타나는 Element와 Attribute 객체를 트리형태로 저장하게 된다. 또한 하나의 DTD를 따르는 문서들에 대한 구조 정보를 저장하기 위해 저장하고자 하는 문서에 해당하는 DTD 객체를 XMLSet.m_DTDSetsList에서 추출하여 DTD.m_dtdElementList를 이용하여 DTDElement들을 탐색하고 저장하고자 하는 Element와 Attribute의 구조정보를 나타낼 DTDElement를 추출하여 DTDElement.docElementList와 DTDElement.docAttributeList에 Element와 Attribute 객체를 추가하게 된다. 그림 2는 XML 문서의 저장 방식을 나타낸다.

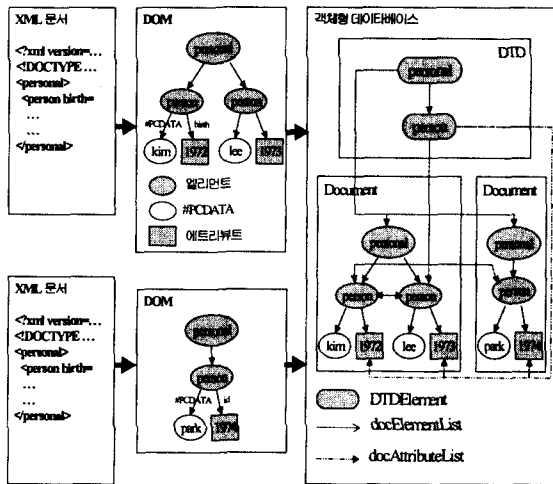


그림 2. XML 문서 저장 방식

그림 2와 같은 방식으로 저장된 XML 문서는 특정 위치의 엘리먼트를 쉽게 탐색할 수 있다. 예를 들어 "person이라는 엘리먼트의 자손인 모든 person 엘리먼트중 이름이 kim인 엘리먼트를 찾아라"라는 질의를 생각해 보자. DTD 객체에서 DTDElement.m_Name이 person인 DTDElement 객체를 찾고 DTDElement.m_dtdElementList에 존재하는 DTDElement 객체중에서 DTDElement.m_Name이 person인 DTDElement를 찾아서 DTDElement.docElementList에 대하여 콜렉션(collection)에 대한 질의를 통해 Element 객체 중에서 Element.m_Name이 kim인 Element 객체를 검색하여 얻어 오게 된다.

4. 결론 및 향후 계획

본 연구에서는 XML 문서에서 임의의 위치에 있는 엘리먼트와 에트리뷰트를 효과적으로 검색하기 위하여 객체지향형 데이터베이스에서 XML 저장 스키마를 설계하였다. 파일시스템과의 연동 없이 내용 및 구조 검색이 가능하므로 개발이 간편하며, XML 문서를 분할 방식을 이용하여 저장하므로 삽입 및 갱신이 용이하도록 하였다. 따라서 빈번한 삽입 및 갱신이 발생하거나 문서내의 특정 위치의 엘리먼트를 주로 검색하는 시스템에 효과적인 설계이다. 향후 연구 방향으로 본 설계를 이용한 저장 및 검색 시스템을 구현하고 평가하여 XML 문서를 객체지향형 데이터베이스를 이용하여 효과적으로 저장하고 검색하는 것에 대한 연구가 필요하다.

5. 참고 문헌

- [1] Extensible Markup Language(XML) 1.0, <http://www.w3.org/TR/PR-xml-971208>
- [2] Jonathan Robie, Joe Lapp, David Schach, XML Query Language(XQL), <http://www.w3.org/TandS/QL/QL98/pp/xql.html>
- [3] Alien Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, Dan Suciu, A Query Language for XML(XML-QL), <http://www.w3.org/TR/1998/NOTE-xml-ql-19980819>
- [4] Document Object Model(DOM) Level 1 Specification, <http://www.w3.org/TR/1998/PR-DOM-Level-1-19980818>
- [5] R. Sacks-Davis, T. Arnold-Moore, and J. Zobel, "Database Systems for Structured Documents" In International Symposium on Advanced Database Technologies and Their Integration, pp. 272-283, Naran Japan, 1994
- [6] Patricia Francois, "Generalized SGML Repositories: Requirements and modelling" Computer Standards and Interfaces 18, 1996
- [7] 장현철, "관계 데이터베이스를 이용한 구조화 문서의 색인 및 검색" 충남대학교 석사학위논문, 1998
- [8] 연제원, 조정수, 이강찬, 이규철, "XML 문서 구조검색을 위한 저장 시스템 설계" 정보과학회 학술발표 논문집(B) 26 권 1호, 1999
- [9] 연제원, 장동준, 김용훈, 이강찬, 이규철, "효율적인 검색지원 SGML 저장관리의 설계 및 구현" 데이터베이스 학술대회 논문집, 1999
- [10] 김현기, 노대식, 강현규, "DocMaster/RM:SGML/XML 문서 관리 시스템" 한국정보처리학회 추계학술발표논문집 제 6권 제1호, pp.76-79, 1998