

ODBMS기반의 XML 문서저장관리시스템

왕지현⁰ 김현기 정의석 임수중 임명은 윤보현
한국전자통신연구원, 언어공학연구부
(jhwang, hkkim, eschung, isj, melim, ybh)@etri.re.kr

A ODBMS-based XML Document Repository System

Wang, JiHyun⁰ Hyun-Ki Kim Euisok Chung Soojong Lim Myeong-Eun Lim
Bo-Hyun Yun
Dept. of Linguistic Engineering, Electronics and Telecommunications Research Institute

요 약

컴퓨터 산업이 발전하고 정보량이 늘어남에 따라 XML이 다양한 분야에 사용되고 있다. 본 논문은 여러 분야에 활용되고 있는 XML 문서들을 효율적으로 저장하고 관리할 수 있을 뿐만 아니라 다 사용자 환경에서 문서의 전체나 일부를 공유할 수 있는 기능을 제공하는 XML 문서저장관리시스템을 소개한다. 다양한 DTD를 수용하기 위해 객체 지향적인 일반화 데이터 모델을 제안하며, 시스템을 구성하고 있는 각 구성요소 모듈을 설명한다. 일반화 데이터 모델은 DTD 의존적인 모델에 비해 연산량이 적을 뿐더러 여러 DTD를 하나의 스키마로 수용할 수 있다.

1. 서 론

대량의 정보들을 구조화된 방식으로 저장, 관리하고 여러 사용자들이 서로 공유하기 위한 방법으로 XML에 대한 관심이 날로 증가하고 있다. 이에 따라 XML의 기능을 충분히 활용하여 이용하는 것은 가치있는 일이다. XML 문서의 생명주기 전 과정에서 정보의 생산성, 재사용성, 지속성, 이식성 등과 같은 XML사용의 장점들을 얻기 위해서는 XML 엔티티들을 저장하고 관리할 수 있는 문서 관리 시스템이 필수적으로 요구된다. XML 문서 관리 시스템 기능들로서는 전자 문서에 대한 라이브러리 서비스 기능(check-in/check-out, versioning, 사용자 액세스 관리 등), 동적 색인 및 구조 정보 검색 기능, 스타일 시트를 이용한 출판 기능 등이 있다.

XML 문서 관리 시스템의 구현 시 고려해야 될 요구 사항들은 다음과 같다.

- 전자 문서 모델링 기능: XML문서를 엘리먼트별로 나누어 데이터베이스에 저장하고 원래 문서로 정보의 손실 없이 복원하기 위한 모델링이 필요하다.
- 라이브러리 서비스 기능: 문서의 생명 주기(life cycle) 제어 기능, 사용자 권한에 의한 액세스 제어 기능, 다수 저장의 동일 문서에 대한 공동 저작시 발생하는 문서의 불일치를 방지하는 check-in/check-out 기능, 문서의 생명 주기 동안의 변경 사항을 관리하는 버전 기능 등을 제공해야 한다.
- 공동 저작 기능 : 그룹 사용자들이 동일한 XML

문서의 전체 또는 일부분을 동시에 편집, 생성, 갱신 등을 할 수 있는 공동저작 기능을 제공해야 한다.

- 동적 문서 생성/교환 기능: 사용자의 질의 또는 문서 생성 템플릿에 의해 저장소에 저장된 컴포넌트들을 재구성하여 동적으로 문서를 생성하는 기능 및 문서의 컨텍스트를 유지할 수 있어야 한다.
- 동적 색인 기능: 문서 관리 시스템에 임포트되는 문서 및 기존에 저장된 문서의 수정/삭제시 구축된 색인 정보를 동적으로 갱신하는 기능을 제공해야 한다.
- 구조 정보 검색 기능: 저장소에 저장된 XML 문서 중 사용자에게 접근 권한이 허용된 문서들에 대한 구조, 속성, 키워드에 대한 검색 기능이 제공되어야 한다.
- 문서 변환/출판 기능: 문서 관리 시스템에 저장된 문서의 다른 포맷으로의 문서 변환 및 출판 기능을 제공해야 한다.

2. XML 데이터 모델

XML 문서저장관리 시스템은 DB저장 시스템으로 ODMG C++에 기반한 O2 ODBMS를 사용하였다. O2는 ODMG Release 1.2에 호환성이 크고 OQL(Object Query Language)를 지원한다.

관계형(Relational) DBMS에 기반한 XML 모델링은 객체 지향적(Object based) DBMS에 기반한 모델링에 비해 다음과 같은 단점을 갖는다.

- 일 대 일, 일 대 다, 다 대 일, 또는 다 대 다와 같은 엘리먼트들 간의 관계를 유지하기 위해 많은 테이블과 추가적인 매핑 코드들이 필요하다.
- 일반적인 관계형 데이터베이스 시스템들의 가변길이 데이터 타입에 대한 지원이 제한적이기 때문에 가변길이의 XML 데이터를 저장할 때 정보의 손실이 있을 수 있다.
- 공동 저장 기능 지원을 위해 문서 전체나 일부분을 보호하는 락(lock) 기능이 지원되어야 한다. 관계형 모델은 비록 특정 엘리먼트에만 락을 수행할 지라도 데이터가 여러 테이블에 걸쳐 흩어져 있기 때문에 더 많은 비용을 요구한다.

관계형 XML 모델링에 비교해 볼 때, 객체형 모델링은 XML의 계층적인 트리구조가 데이터 모델에 그대로 반영될 수 있기 때문에 조인연산을 필요로 하지 않는다.

2.1 DTD 의존 모델(DTD-dependent Model)

XML은 여러 형태의 문서들을 모델링하기 위한 언어인 반면, 특정 문서 모델이나 기정의 태그 셋에 의존적이지 않다. 초기에 시스템을 개발할 때는 DTD의 논리적 엘리먼트 구조를 그대로 반영한 DTD 의존적인 방법으로 개발하였다. 이 모델은 새로운 DTD를 처리할 때마다 DTD에 기술된 엘리먼트 컨텐트 모델에 따라 스키마를 자동으로 생성하였다. 그러나 이 모델에서는 다음과 같은 몇 가지 단점을 가지고 있다.

- 대부분의 데이터베이스 시스템들은 발생 지시자(occurrence indicators) 및 연결자(connectors)와 같은 데이터 타입을 지원하지 않기 때문에 문서를 삽입하거나 문서 일부를 수정할 때 무결성(integrity) 및 내부 유효성(validation)을 보장할 수 없다.
- DTD에 따른 논리적인 클래스를 생성하기 위해 O2시스템이 제공하는 O2Engine API를 사용하여 스키마 및 오브젝트, 클래스에 대한 직접적인 제어를 수행한다. 이에 따라 예를 들어, 새로운 클래스 이름을 지정해 할 때 O2 시스템 내의 예약어와 이름 충돌(naming conflict)이 일어날 수 있다. 또한 엘리먼트 이름이 한국어일 경우에는 이름 매핑(name-mapping) 테이블이 필요하다.

- 저장소 구조가 수많은 스키마와 데이터베이스로 흩어져 있고 많은 계층적 구조로 구성되어 있기 때문에 저장소 사용자들이 저장소의 특정 객체 및 값으로 표현하여 질의를 하는 것이 어렵다.

2.2 일반화 데이터 모델(Generic Data Model)

위와 같은 DTD 의존 모델의 단점에 따라, 하나의 스키마로 여러 DTD들을 수용할 수 있는 일반화 데이터 모델을 설계했다. 그림 1은 이 모델의 클래스 구조를 나타낸다.

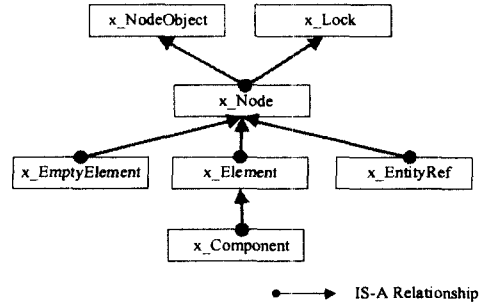


그림 1. XML 트리 모델의 클래스 구조

x_NodeObject 클래스는 XML 트리 모델의 슈퍼 클래스이며, 노드 타입에 대한 정보를 보관하는 클래스이다. XML 파스 트리의 위치정보와 노드 이름/데이터에 대한 정보를 관리하는 클래스이다. XML의 PCDATA, PI(Processing Instruction), Comment 등은 파스 트리의 단말 노드에 위치하며, 별도의 클래스로 정의하지 않고 x_Node 클래스에 값을 설정한다. XML ELEMENT 노드는 x_Element 클래스에 값이 설정된다. x_Element 클래스는 파스 트리의 비단말 노드이므로 자식을 가질 수 있다. 또한 Attribute Instance를 리스트로 가질 수 있다. XML EMPTY ELEMENT는 자식 노드를 갖지 않고 Attribute Instance만을 가지며 트리의 단말 노드에 위치한다. x_EmptyElement 클래스는 EMPTY ELEMENT에 대한 정보를 보관하는 클래스이다. x_EntityRef 클래스는 entity reference에 대한 정보를 보관한다. Entity name은 x_Node 클래스에 설정되며, entity type에 대한 정보를 저장한다. x_Component 클래스는 x_Element 클래스로부터 상속받아 정의되며, 별도의 속성을 갖지 않는다. 이 클래스는 XML 파스 트리에서 자식 노드들을 별도로 저장하지 않고 PCDATA로 합쳐서 저장하는 데 사용된다.

3. 시스템 구조

이 시스템은 Solaris 2.5환경에서 SPARCompiler C++ 4.1과 O2 ODBMS5.0을 사용하여 구현하였다. 그림 2는 XML문서저장시스템의 시스템 구조를 나타낸다. XML 문서저장시스템의 하부 저장시스템으로는 O2 ODBMS를 사용하며, TCP/IP(Transmission Control Protocol/Internet Protocol)를 이용한 연결 지향형 클라이언트-서버구조로 구현되었다. O2는 페이지 서버 아키텍처를 갖는 객체 지향 데이터베이스 시스템으로 C++바인딩과 OQL(Object Query Language)를 지원하며 상속과 다형성등의 객체 지향 개념등을 제공한다. 효율적인 다사용자 처리를 위해 preforked TCP/IP 서버 구조로 개발되었고 사용자들이 쉽게 문서를 저장하고 관리할 수 있도록 하기 위한 클라이언트 애플리케이션을 개발하였다.

시스템은 다음과 같은 구성요소를 가지고 있다. (1) 통신 관리자(Communication Manager), (2) XML 툴킷(XML Toolkit), (3) 라이브러리 서비스 관리자, (4) XML 객체 관리자

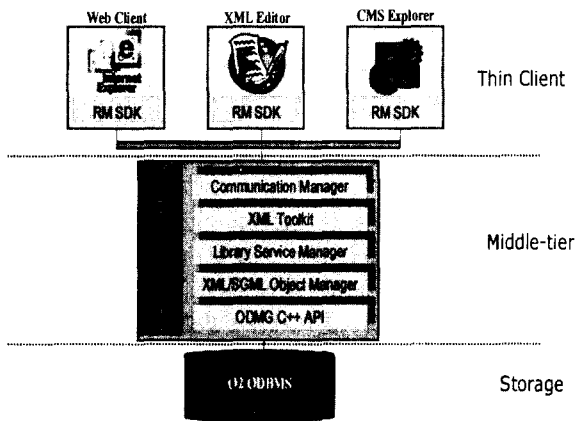


그림 2. 시스템 구조

3.1 시스템 구성요소

통신관리자(Communication Manager)

사용자와 시스템간의 연결을 제공하고 시스템 내부의 메시지 및 작업을 제어한다. 시스템이 시작될 때 자식 프로세스들이 미리 생성되어 클라이언트의 처리 요구를 대기한다. 통신 관리자로 들어오는 클라이언트 처리 요구마다 상태정보를 유지한다.

XML 툴킷(XML Toolkit)

XML 파서의 출력 포맷은 ESIS(Element Structure Information Set) 표현으로 된 간단한 텍스트이다. 하지만 어떤 XML 표현은 완전하게 제공하고 있지 못하기 때문에 James Clark의 SP에 기반한 인-메모리(in-memory) 트리 구조를 제공하는 인텔리전트 XML 툴킷을 개발했다. DTD의 파싱 결과인 룰 트리(Rule Tree)뿐만 아니라, XML 인스턴스의 파싱 결과인 인스턴스 트리(Instance Tree)도 만든다.

라이브러리 서비스 관리자(Library Service Manager)

구조화된 XML 문서를 XML 툴킷을 사용하여 DB저장소에 저장한다. 라이브러리 서비스 관리자의 XML문서에 대한 DB저장소로의 문서 변환 규칙은 다음과 같다.

- 1) 파서에 의해 생성된 인스턴스 트리아의 모든 노드는 부모 및 형제, 자식 노드들을 가리키는 클래스 "x_Node"의 객체로 변환된다.
- 2) 모든 저장소 객체들은 애플리케이션과 저장소간의 일관된 레퍼런스를 유지하기 위해 논리적 객체 판별자(Logical Object Identifier)를 갖는다.
- 3) XML문서가 저장소에 저장될 때 문서에 포함된 XML Entity 레퍼런스들은 논리적 객체 판별자를 사용하여 객체 레퍼런스로 변환된다.

XML/SGML 객체 관리자(XML/SGML Object Manager)

저장소 객체 관리자는 라이브러리 서비스 관리자에 의해 저장소 객체를 관리한다. 다 사용자 환경에서 문서 일부분이나 전체를 하나로 통합하거나 편집하기 위한 체크인/체크아웃(Check-in/Check-out) 기능을 제공하고 문서의 히스토리 관리를 위한 버전 관리 기능, 사용자의 권한에 따른 액세스 제어 등을 제공한다.

4. 결론

ODMG 객체 모델에 따른 XML의 일반화 데이터 모델과 O2 ODBMS에 기반한 XML 문서저장관리 시스템의 구현 내용을 보여주었다. 사용자는 구조화된 XML 문서 뿐만 아니라, 워드나 HTML등 다양한 형태의 문서 포맷도 저장하고 관리할 수 있으며 사용자로 하여금 읽기, 쓰기, 관리 액세스 권한을 설정할 수 있는 기능을 제공한다. 또한 다양한 DTD를 수용할 수 있으며, 특정한 단위로 저장되어 있는 구조화된 문서를 동적으로 저장소로부터 읽어서 원형의 문서로 복원할 수 있는 기능을 제공한다. 그리고 다 사용자 지원을 위한 체크인/체크아웃, 버전 컨트롤, 문서레벨 보안 기능을 제공한다.

5. 참고문헌

[1] W3C XML Working Group, Extensible Markup Language(XML), Available via <http://www.w3c.org/XML>, 1998

[2] Patrica Francois, "Generalized SGML Repositories: Requirements and modeling," Computer Standards and Interfaces 18, 1996

[3] R.G.G. Cattell, The Object Database Standard: ODMG-93, Morgan Kaufmann Publishers, Inc., San Francisco, California, 1996

[4] W3C, Extensible Stylesheet Language(XSL) Specification, W3C Working Draft, 1999

[5] O2 Technology, ODMG C++ Binding Guid, Release 5.0, December 1996