EJB를 이용한 XML 문서 저장

허율[0], 홍의경
서울시립대학교 전산통계학과
{hysun, ekhong}@venus.uos.ac.kr

Storage of XML Documents Using EJB

Youl Heo[0] Eui-Kyeong Hong

Dept. of Computer Science and Statistics, University of Seoul

요 약

웹의 표준으로 자리잡고 있는 XML(eXtensible Markup language)은 정보교환을 위한 중요한 포맷으로 대두되고 있으며, 현재 EC/EDI, 전자 도서관, 전자 상거래 등 다양한 분야에서 XML이 사용되고 있다. 따라서 XML 문서를 데이터베이스에 효율적으로 저장하고 검색하기 위한 많은 연구들이 진행되고 있다.

XML 문서를 저장하고 검색하고자 하는 시스템은 다양한 시스템의 변화에 효율적으로 적응하고 재사용과 확장성이 용이한 컴포넌트 기반의 소프트웨어로 변환되어야 한다. 본 논문에서는 복잡한 시스템의 개발 기간을 단축하고 소프트웨어 품질과 재사용성을 높이는 객체지향 및 컴포넌트 표준으로 알려진 EJB(Enterprise JavaBeans)를 이용해 XML 문서를 저장하는 방법을 제시하였다.

1. 서 론

인터넷의 급속한 보급과 발전함에 따라 현재 수 많은 양의 정보들이 전자 문서의 형태로 존재하고, 앞으로도 전자 문서의 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 XML을 웹 문서의 표준으로 지정하였다. XML 문서는 논리적인 구조 정보를 가지고 있고 플랫폼 및 소프트웨어 중립적이기 때문에 효과적인 검색, 관리 및 공유가 가능한 시스템 환경을 제공한다. 현재 인터넷 웹문서뿐만 아니라 전자 도서관, 전자 상거래, EC/EDI를 포함한 다양한 분야에서 XML 문서를 효과적으로 저장하고 검색할 수 있는 방법에 대한 연구가 진행되고 있다[4].

소프트웨어 산업이 급속하게 발전함에 따라 정보 기술 업체간 경쟁이 더욱 심화되어 소프트웨어 재사용성, 적시성(time to market), 유지 보수성 등이 업체의 생명력으로 대두되면서 컴포넌트 소프트웨 기술이 각광을 받고 있다. 현재 가장 널리 사용되고 있는 컴포넌트 수행은 EJB, OMG의 CORBA Component Model(CCM), 마이크로소프트의 DCOM 등이 있다. 이중 웹 서버 기반의 객체지향 및 컴포넌트 표준으로 알려진 EJB는 복잡한 시스템의 개발 기간을 단축하고 소프트웨어 품질을 높이며 재활용성이 높은 컴포넌트 기반의 소프트웨어 개발을 가능하게 한다[1,2].

* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

본 논문에서는 XML 문서를 저장하고 XML문서의 구조정보와 내용정보를 효과적으로 검색할 수 있도록 설계한 저장 모델을 가지고 엔터프라이즈급 개발을 위한 서버 측 분산객체 컴포넌트인 EJB에 저장 방법을 제시하고자 한다. 본 논문의 구성은 다음과 같다. 2절에서는 XML문서를 저장과 검색하기 위한 저장 모델을 간략히 기술하고, 3절에서는 EJB의 아키텍처를 살펴본다. 4절에서는 XML 문서 저장 모델을 가지고 EJB에 저장하는 방법을 제시한다. 5절에서는 결론 및 향후 연구 방향을 기술한다.

2. XML 문서 저장 모델

2.1 XML 문서 검색을 위한 저장 모델의 특징

XML 문서를 저장할 수 있도록 DTD 독립적인 모델을 제시하였고 XML 문서의 빈번한 수정이 있을 것으로 가정하여 문서를 엘리먼트 단위로 쪼개어 저장하는 분할 저장 기법을 사용하였다. OID를 저장함으로써 조인이 아닌 경로 표현으로 효과적인 질의 수행 및 빠른 검색을 지원하도록 하였고 XML 문서의 구조적인 정보에 대한 효율적인 유지 및 처리를 위하여 Pathexp필드를 사용하였다[3,5].

2.2 XML 문서 스키마

XML 문서를 저장하기 위한 스키마는 그림 2.1과 같다. 스키마를 구성하는 테이블 간의 관계를 도식화한 것으로 n을 가지는 굵은 화살표는 set-value를 의미한다. 저장테이블의 구조는 그림 2.1에서 보는 것과 같이 DTD 테이블, XML 문서 테이블, 엘리먼트 테이블, 컨텐츠 테이블, 애트리뷰트 테이블로 구성된다.

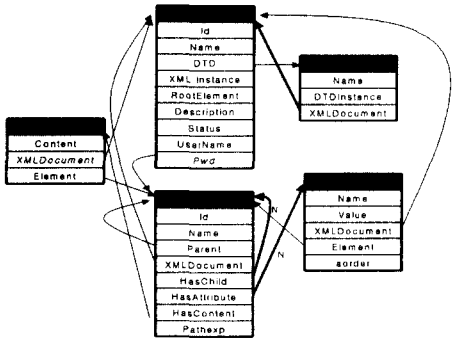

그림 2.1: XML 문서 저장 스키마.

2.3 XML 문서 정보

그림 2.2는 증권정보를 표현한 XML 문서이다. 이 XML 문서는 기업명, 증권 코드, 일자, 종가, 시가, 고가 등의 엘리먼트들로 구성된다.

```
<?xml version="1.0" encoding="euc-kr" ?>
<STOCK>
 <기업명>
    <Stockdata>samyang</Stockdata>
  </기업명>
  <코드>
    <Stockdata>080</Stockdata>
  </코드>
  <일자>
    <Stockdata>2001.03.30</Stockdata>
  </일자>
  <종가>
    <Stockdata>10800</Stockdata>
  </종가>
  <시가>
    <Stockdata>10800</Stockdata>
  </시가>
  <고가>
    <Stockdata>11000</Stockdata>
  </고가>
</STOCK>
```

그림 2.2: 예제 XML 문서(stock.xml).

그림 2.2의 stock.xml 문서를 DOM 파서에 의하여 파싱을 하면 그림 2.3과 같은 트리의 형태로 바뀐다. DOM은 W3C에서 XML 문서를 액세스하기 위해 제공하는 인터페이스인데 XML 문서를 트리 형태로 나타낸 것으로서 XML의 큰 특징 중의 하나인 계층적 정보를 가장 잘 표현하는 인터페이스이다. 파싱의 결과로 얻어진 DOM 트리의 각 노드는 테이블로 매핑되어 저장된다.

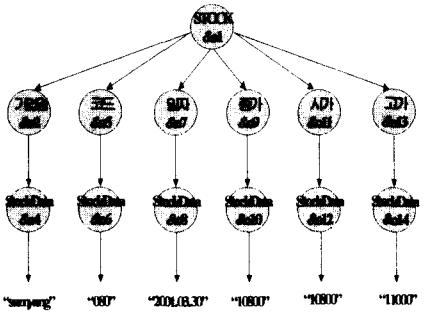

그림 2.3: XML 문서의 DOM 트리 구조.

3. EJB의 아키텍처

EJB 아키텍처는 썬(Sun)사에서 제안한 서버 측 분산 객체 컴포넌트로서 비즈니스 어플리케이션의 개발 및 배치를 위한 아키텍처이다. EJB 아키텍처는 홈/원격 인터페이스와 엔터프라이즈 빈(Bean), 배치 설명 파일(Deployment Description File)로 구성되어 있다. 엔터프라이즈 빈은 세션 빈(Session Bean)과 엔티티 빈(Entity Bean)으로 나뉘는데 엔티티 빈일 경우는 EJB 구성 요소에 기본키 클래스가 포함되고 데이터베이스에 연결되어 영구적인 자료 관리를 할 수 있는 빈을 말한다. 세션 빈일 경우는 기본키 클래스가 EJB 구성 요소에 포함되지 않고 일반 비즈니스 로직을 가진다. EJB 컨테이너는 트랜잭션의 무결성, 명명(Naming), 보안, 지속성 서비스를 제공하는 EJB 서버 상에 놓여 있다. EJB 서버가 제공하는 서비스의 선택은 배치자(Deployer)가 빈을 EJB 컨테이너에 배치할 때 DD(Deployment Description)에 기술하게 하여 빈의 이식성을 향상시킨다. 외부의 클라이언트가 빈을 이용하기 위해서는 JNDI(Java Naming and Directory Interface)를 이용하여 특정 빈의 홈 인터페이스에 접근하여 빈을 생성한 후 원격 인터페이스를 통해 빈이 제공하는 기능을 생성한다[1,2].

4. EJB를 이용한 XML 문서 저장

이 절에서는 객체지향 언어인 자바언어를 기반으로 개발된 XML 문서 저장 시스템을 자바 분산 컴포넌트 기술인 EJB 기반의 응용으로 변환하는 방법을 제시하고 적용해 본다.

1단계에서는 EJB 컴포넌트를 모델링하기 위한 계층별 모델로서 표현 계층과 비즈니스 로직 계층, 데이터 계층으로 나눈다. 표현 계층은 클라이언트 측 인터페이스로서 서블릿과 JSP로 구성되고, 비즈니스 로직 계층은 EJB 빈을 추출하기 위한 계층으로 XML 문서를 파싱하고 저장과 검색을 담당하는 클래스들이며 데이터 계층은

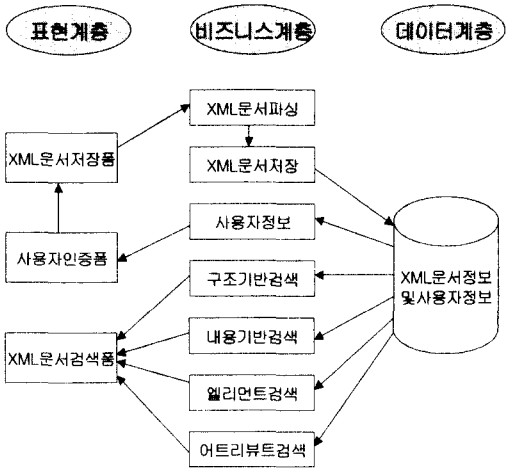

그림 4.1: 계층별 모델.

영속적인 데이터를 저장하는 계층으로서 XML 문서 정보와 사용자 정보를 데이터베이스에 저장한다. 그림 4.1은 1단계 계층별 모델을 나타낸다.

2단계에서는 1단계의 계층별 모델 중 비즈니스 로직 계층을 기반으로 빈 클래스를 분류하는 단계이다. 이 단계는 XML 문서의 저장 부분에 해당하는 클래스명과 클래스 설명를 나타낸다. CreateUserInfo 클래스는 사용자 이름과 패스워드를 검사하여 XML 문서를 저장할 수 있는지 없는지를 식별할 수 있는 클래스 이고 Store 클래스는 XML 문서를 DOM 파서를 통해서 DB에 저장하는 클래스이다. 나머지 클래스들은 XML 문서의 정보를 나타내는 클래스들이다. 표 4.1은 클래스의 분류를 나타낸다.

표 4.1: 클래스의 분류.

클래스분류	클래스명	클래스 설명
Bean Class	CreateUserInfo	User의 정보
	Store Document Element Content Attribute Childment 등	XML문서저장 및 문서정보 표시

3단계에서는 클래스명에 따라 빈 유형을 결정하고 그에 따르는 메소드와 속성 상태를 분류하는 단계이다. 엔티티 빈은 영구적인 데이터 관리가 필요한 빈으로서 컨테이너 관리 지속성(container-managed persistence)과 빈 관리 지속성(bean-managed persistence)으로 나누어진다. 전자는 데이터베이스와 연동하는데 컨테이너가 관리하고 후자는 개발자가 일일이 코딩을 한다. 이 단계에서는 전자의 엔티티 빈으로서 XML 문서를 저장하기 위해 설계된 스키마 형식으로 테이블을 만든다. 이 테이블의 정보를 이용해 어느 사용자가 XML 문서를 저장할 수 있는지 식별할 수 있고 개별 XML 문서의 정보를 알 수 있는 메소드와 속성으로 분류한다. 세션 빈은 DOM 파서를 이용해 XML 문서를 파싱하고 파싱된 구조와 내용을 파악하여 데이터베이스의 테이블에 저장하기 위한 빈들이다. 세션 빈의 메소드와 속성은 파싱된 구조와 내용을 깊이 우선 탐색을 하여 엘리먼트, 애트리뷰트, 컨텐츠, 자식 노드 등이 존재하는지 여부를 확인하여 데이터베이스 테이블에 저장을 담당한다. 표 4.2는 XML 문서 저장의 핵심 부분인 빈의 유형에 따라 메소드와 속성 상태를 분류한 것이다.

5.결론 및 향후 연구 방향

본 논문에서는 XML 문서 검색을 위한 저장모델을 설계하였고 이 저장 모델을 가지고 객체지향 언어인 자바언어로 개발된 응용을 계층별 분류, 클래스의 분류, 빈 유형에 따른 메소드와 속성 상태 분류의 3단계를 거쳐 분산 컴포넌트 기술인 EJB 기반의 응용으로 전환하는 방법을 예로 들었다. EJB 기반을 이용해 XML 문서를 저장하는 시스템으로 변환함으로써 다양한 시스템의 변화에 적

표 4.2: 빈 유형에 따라 메소드와 속성 상태 분류.

컴포넌트 이름	빈의 종류	메소드 시그니처		호출되는메소드	메소드기능설명
XML문서 저장 빈 컴포넌트	Entity Bean	메소드	Check()	setUsername() setpasswd()	DB의 사용자 유무 확인
		속성		String username String pwd	사용자 이름과 비밀번호 확인
		메소드	Document()	setDocid, setName등	개별 XML 문서 정보를 나타냄
		속성		String Docid, String name int dtdid int rootelementid String description	XML 문서 id와 이름, 설명 등을 나타냄
	Stateful Session Bean	메소드	Storeby Filename()	makeQueryVector()등	DOM을 통해 질의 형태로 바꾼 XML문서를 DB에 저장
		속성		String Docname int dtdid String desc	XML 문서 이름, 문서아이디, 문서의설명
	Stateless Session Bean	메소드	makeQueryVector()	ge_RootStore() getChildrenforStore()	XML 문서를 질의 형태로 바꾼 벡터
		속성		String filename int dtdid Element rootElement String desc	XML 문서의 루트 정보를 저장하기 위해서 필요한 요소

응할 수 있고, 거대해지고 복잡해진 소프트웨어의 재사용성과 확장성을 기대할 수 있게 되었다. 향후에는 EJB를 이용해 XML 문서의 검색 부분의 변환 방법과 보안, 웹 상의 모든 문서가 XML 문서로 이루어지지 않은 문서들을 XML 문서로 변환하는 기법을 연구할 것이다.

6. 참고 문헌

[1] Sun Microsystems, Inc., "Enterprise JavaBeans Specifications," Version 2.0, 2000.

[2] Roman, E., "Mastering Enterprise JavaBeans, and the Java™2 Platform, Enterprise Edition," John Wieley & Sons, Inc., 1999.

[3] T.Shimura, M.Yoshikawa, and S.Uemura, "Storage and Retrieval of XML Documents Using Object-Relational Databases," DEXA99, pp.206-217, 1999.

[4] 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.

[5] D.Florescu and D.Kossmann, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulleltin 22(3), pp.27-34, 1999.