

# 객체 모델을 기반으로 한, XML 문서의 관계형 데이터베이스로의 저장을 위한 EJB 컴포넌트 개발

김용수<sup>0</sup>, 주경수  
순천향대학교 공과대학 전산학과  
admin@kimys.pe.kr<sup>0</sup>, gsoojoo@sch.ac.kr

## Developing EJB Component for Storing XML Documents in DataBase based on Object Model

Kim Yong-Soo<sup>0</sup>, Joo Kyung-Soo  
Dept. of Computer Science College of Engineering Soonchunhyang University

### 요 약

최근 소프트웨어 개발비용의 절감과 신뢰성 있는 소프트웨어의 개발을 위하여 컴포넌트 기반의 소프트웨어 개발에 대한 연구가 활발하게 진행되고 있다. 컴포넌트 기반의 소프트웨어 개발 시 중요한 문제 중 하나는, 전체 시스템 아키텍처에서 어떠한 컴포넌트가 필요하며 아울러 어디에 적합할 것인가를 결정하는 것이다.

본 논문에서는 XML 문서를 관계형 데이터베이스에 원활한 저장을 위해서 객체 모델을 기반으로 한, XML 문서의 관계형 데이터베이스에 저장을 위한 컴포넌트를 개발하여, 이를 EJB로 구현하였다. 이에 따라 사용자들은 관계형 데이터베이스 기반의 XML 응용 시스템을 개발하고자 할 때, 컴포넌트들을 단순·조립함으로써, 시간단축 및 비용절감을 기대할 수 있다.

### 1. 서 론

XML은 HTML보다 사용자의 다양한 요구를 충분히 수용할 수 있고 SGML보다 사용하기 쉽다는 장점으로 인해서 웹 문서뿐만 아니라 전자도서관, EDI 등을 포함한 다양한 분야에서 XML을 활용하고 있다. XML은 HTML이 하나의 고정된 DTD를 사용하는 것과는 달리 논리적 구조를 나타내는 여러 DTD를 사용할 수 있다는 점에서 SGML과 같다. DTD의 논리적 구조를 따르는 XML 문서의 구조정보는 데이터베이스에 저장된 XML 문서의 효율적으로 관리할 수 있다.

본 논문의 2절에서는 EJB 아키텍처에 대해 알아보고, 3절에서는 XML 문서의 관계형 데이터베이스에 저장 방법을 설명한다, 4절에서는 EJB 구현을 보도록 한다. 마지막으로 결론 및 향후 연구 방향을 기술한다.

### 2 EJB 아키텍처

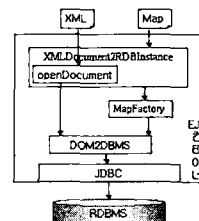
EJB는 컴포넌트 기반 분산 객체 기술이다. EJB는 엔터프라이즈급 개발에서 데이터 추상화와 비즈니스 로직에 대한 부분을 담당하는 매우 핵심적인 요소이다. 엔티티 빈(Entity Bean)에는 기본키 클래스가 포함되고 데이터베이스에 연결되어 영구적인 자료 관리를 담당한다. 비즈니스 로직은 세션 빈(Session Bean)이 담당하는데 이 빈들은 모두 분산 객체 기술을 그 바탕에 두고 있다. EJB1.1 버전부터 RMI-IIOP를 통해 CORBA를 지원하고 있고, EJB2.0에서는 JMS(Java Message Service)를 통해 비동기적인 빈즈를 만들 수 있게 되었다. EJB 컨테이너는 트랜잭션의 무결성, 명명(Naming), 지속성, 보안 서비스를 제공하는 EJB 서버 상에 놓여 있다[3][4].

본 논문의 구현에 사용된 빈은 세션 빈이다. 데이터베이스 테이블과 관련된 빈은 엔티티 빈이지만 다양한 XML 문서에 따라 데이터베이스 스키마가 변형되기 때문에 세션 빈에서 JDBC2.0를 이용한 데이터베이스 접근을 하였다.

### 3. XML 문서의 관계형 데이터베이스로의 저장 방법

그림 3은 XML 문서를 관계형 데이터베이스의 테이블에 저장구조를 보여 주고 있으며 작업 단계는 다음과 같다.

- ① XML 문서 -> DOM Tree로 표현
- ② Map 문서 -> Map 객체로 표현
- ③ DOM2DBMS -> XML 정보를 담고 있는 DOM Tree는 Map의 규칙에 따라 데이터베이스에 저장되는 SQL 문장을 만든다.



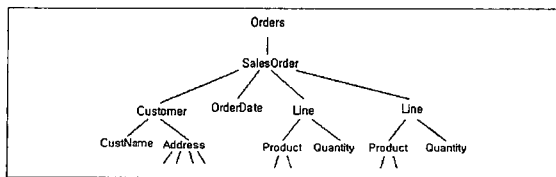
(그림 3) XML 문서 저장 구조

### 3.1 openDocument

openDocument는 컴포넌트 내부 메소드로 입력받은 XML 문서의 엘리먼트들을 DOM Tree로 변환한다. openDocument에서 변환된 DOM Tree의 각 노드를 순회하면서 DOM2DBMS에서 SQL 문을 생성하게 된다. 그림 4의 sales.xml 문서를 openDocument 메소드에 의 해 그림 5와 같은 DOM Tree 형태가 만들어진다.

```
<Orders>
  <SalesOrder SONumber="12345">
    <Customer CustNumber = "543">
      <CustName>ABC Industries</CustName>
      <Address> ..... </Address>
    </Customer>
    <OrderDate>150999</OrderDate>
    <Line LineNumber="1">
      <Product Name="Rainier Cherries">.....</Product>
      <Quantity Unit="ton">2</Quantity>
    </Line>
    <Line LineNumber="2">
      <Product Name="Gala Apples"> .....</Product>
      <Quantity Unit="ton">5</Quantity>
    </Line>
  </SalesOrder>
</Orders>
```

(그림 4) sales order 예제



(그림 5) sales.xml의 DOM Tree 형태

### 3.2 MapFactory

```
<ClassMap>
  <ElementType Name="SalesOrder"/>
  <ToClassTable><Table Name="Sales"/> </ToClassTable>
  <PropertyMap>
    <Attribute Name="SONumber"/>
    <ToColumn> <Column Name="Number"/> </ToColumn>
  </PropertyMap>
  <PropertyMap>
    <ElementType Name="OrderDate"/>
    <ToColumn><Column Name="Date"/> </ToColumn>
  </PropertyMap>
  <RelatedClass KeyInParentTable="Candidate">
    <ElementType Name="Line"/>
    <ForeignKey><ColumnName="LineNumber"/> </ForeignKey>
    <OrderColumn Name="Number" Generate="No"/>
  </RelatedClass>
  <RelatedClass KeyInParentTable="Foreign">
    <ElementType Name="Customer"/>
    <ForeignKey> <Column Name="CustNumber"/> </ForeignKey>
  </RelatedClass>
</ClassMap>
```

(그림 6) sales.map 파일

Map 안에 기술된 레지스트리 정보와 스키마 정보, 매

핑 룰 등의 정보를 관리하면서, 매핑 룰에 대한 매핑 테이블을 생성하고, DOM2DBMS가 SQL 문을 생성하는데 필요한 여러 가지 정보를 제공한다.

그림 5는 그림 3의 sales.xml 문서로 도출된 Map 문서이다. Map 문서는 시스템이 자동으로 생성되기도 하며, 사용자 인터페이스를 통해서 나타나는 XML 문서의 구조를 표현한 DTD를 사용자가 보고 판단하여 관계 있는 것끼리 매핑을 시켜주는 방법으로 생성할 수 있다 [5][6].

### 3.3 DOM2DBMS

openDocument에서 생성된 DOM Tree의 각 노드를 순회하면서 Map 객체 생성기로부터 각 노드와 테이블간의 매핑 정보를 제공받아 SQL 문을 생성한다. 이렇게 생성된 SQL 문을 통해서 XML 문서로 되어 있는 데이터를 데이터베이스에 저장하게 된다.

SQL 문장 생성의 과정은 다음과 같다.

- ① DOM Tree를 순회하며 Map 객체에서 정보를 가져온다.
- ② SalesOrder는 엘리먼트이고 이것은 sales 테이블임을 알고 컬럼을 찾는다.
- ③ SalesOrder는 어트리뷰트 SONumber와 엘리먼트 OrderDate를 포함하고, 각각 Number, Data라는 컬럼이 된다.
- ④ Line을 보면, Map에서 <RelatedClass KeyInParentTable="Candidate">와 같은 선언이 되어 있다. 이것은 자식 테이블을 가지며 sales 테이블에 값을 저장하지 않고 Line 테이블에서 sales 테이블의 값을 참조하는 형태가 된다.
- ⑤ Customer를 보면, Map에서 <RelatedClass KeyInParentTable="Foreign">와 같은 선언이 되어 있다. 이것은 자식 노드가 있기 때문에 컬럼에 데이터가 저장되지 않고 Customer 테이블을 만들고 sales 테이블에 CustNumber라는 외래키를 컬럼으로 갖는다.
- ⑥ ① ~ ⑤의 과정을 거쳐 아래와 같은 SQL 문장을 만들고 XML 문서의 데이터를 저장하게 된다.  
 『insert into sales(Number, Data, CustNumbe) values ('12345', '150999', '543')』

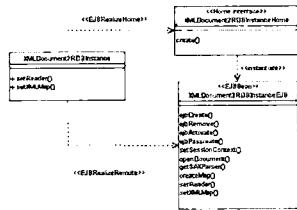
### 4 구현

본 논문은 XML 문서의 관계형 데이터베이스 저장 컴포넌트를 EJB를 이용하여 구현하였다. 개발툴은 JBuilder4.0를 이용하여 컴포넌트를 구현하였고, 실행 환

경으로서 WAS(Web Application Server) 서버인 IAS(Inprise Application Server) 4.1 이용하여 실행하였다.

4.1 컴포넌트 클래스 다이어그램

XML 문서와 관계형 데이터베이스에 저장하기 위한 클래스 다이어그램은 그림 4와 같다. 클라이언트 프로그램에 의해 XML 문서와 Map 문서는 String으로 받게 되고 3점에서 설명한 바와 같이 EJB 컨테이너 안에서 DOM 트리와 Map 객체로 각각 전환되고 이 규칙에 따라 관계형 데이터베이스에 저장된다.



(그림 4) XMLDocument2RDBInstance 클래스 다이어그램

4.2 컴포넌트 인터페이스

4.2.1. Home 인터페이스

EJB 세션 빈에 접근하려는 모든 클라이언트는 홈 인터페이스를 통한다. 즉, JNDI 이름을 이용하여 세션 빈을 찾아낼 때 홈 인터페이스를 이용한다. 무상태 세션 빈은 create() 메소드에 매개 변수를 갖지 않는다.

(표 1) 홈인터페이스

메소드	설명
create()	XMLDocument2RDBInstance 객체 생성

4.2.2 Remote 인터페이스

클라이언트가 세션 빈의 비즈니스 로직을 사용하기 위해서 리모트 인터페이스를 사용한다. 클라이언트가 홈 인터페이스의 create()를 호출하면 EJB 컨테이너는 리모트 인터페이스를 구현한 객체인 EJB 객체를 생성한다. 그후 EJB 객체의 비즈니스 메소드를 호출하면 동일한 이름을 가지는 메소드를 빈 인스턴스로부터 호출하여 서비스를 제공한다. 아래 표 2는 리모트 인터페이스와 각각의 동작을 설명하였다.

(표 2) 리모트 인터페이스

메소드	인자	설명
setXMLMap(String, String)	XML 문서, Map 문서	이 메소드는 컴포넌트 외부에서 읽은 XML 문서와 Map 문서를 String값으로 받아 임시 파일을 생성하는 메소드
setReader(String, String, String)	user, passwd, url	DB 접속 정보들이 들어온다.

4.2.3 EJB Bean 클래스

빈 클래스는 리모트 인터페이스에 선언된 메소드가 실제 구현되어 있는 클래스이다. 또한 내부적으로 쓰이는 자바 빈을 호출한다. 표 3은 빈 클래스의 메소드를 나타내고 있다.

(표 3) 빈 클래스

메소드	인자	설명
setXMLMap(String, String)	XML 문서, Map 문서	이 메소드는 컴포넌트 외부에서 읽은 XML 문서와 Map 문서를 String값으로 받아 임시 파일을 생성하는 메소드.
setReader(String, String, String)	user, passwd, url	DB 접속 정보들이 들어온다 createMap()를 호출, openDocument()를 호출한다.
createMap(InputSource Connection)	conn DB 접속 정보	setReader()메소드에서 호출된다. Map 객체 생성
getSAXParser()	Map 문서	Map 문서 파싱한다.
openDocument(Input Source)	XML 문서	Dom 트리를 생성한다.

5. 결론 및 향후 연구 방향

본 논문에서는 XML 문서 저장에 EJB를 이용하여 컴포넌트화 하였다. 이것은 웹에서 데이터 교환의 표준으로 자리를 잡아가고 있는 XML 문서를 효율적으로 저장하고, 이를 쉽게 재사용 할 수 있게 되었다. 이로 인해 거대하고 복잡해진 소프트웨어의 재사용성과 확장성을 기대할 수 있게 되었다. 현재는 저장 컴포넌트만을 구현하였지만, 향후에는 EJB 컴포넌트를 이용해 저장된 XML 문서를 검색한 값으로 XML 문서를 생성하는 컴포넌트 개발을 연구할 것이다.

참고문헌

- [1] "Extensible Markup Language(XML)", <http://www.w3.org/TR/PR-XML-971208>
- [2] Florescu, D., Kossmann, D.: Storing and Querying XML Data using an RDBMS. Data Engineering 22:3 (1999), 27-34.
- [3] Joo Kyung-Soo, "A Design of Middleware Components for the Connection between XML and RDB", 2001 IEEE International Symposium on Industrial Electronics Proceedings, Pusan, Korea, June 2001
- [4] 유재우, 최재영, 최종명, 박준서, 프로그래머를 위한 EJB, 이한디지탈리
- [5] 이상태; 이정수; 주경수, "객체모형을 기반으로 한 XML DTD의 RDB 스키마로의 변환 방법", 대한전자 공학회, 하계종합논문대회, 제24권 제1호, pp.113-116
- [6] 이상태; 이정수; 주경수, "XML DTD를 기반으로 한, RDB 스키마 설계를 위한 Component 구현", 한국정보처리학회 지식 및 데이터공학연구회 제8회 학술발표대회 논문집, pp.309-316, 2001