

# 데이터베이스 질의 결과로부터 XML 문서 자동 생성 도구 구현

조 송호, 이 원진

강남대학교 컴퓨터미디어공학부

e-mail : shcho@kangnam.ac.kr, jin888798@hotmail.com

## Implementation of an Automatic Tool Generating a XML Document from Database Retrieval

Seung H. Cho, Won J. Lee

School of Computer Eng. and Media Eng., Kangnam University

### 요 약

본 연구에서는 객체를 중간 매개체로 활용하여 관계형 데이터베이스로부터 XML 문서를 자동적으로 추출하는 도구에 대하여 구현하였다. 본 시스템은 XML 문서와 데이터베이스간 맵핑을 위하여 객체-관계 맵핑을 적용하였으며, 데이터베이스 설정, 관계-객체 스키마 맵핑, XML 생성 등의 기능을 제공한다. 본 연구 결과는 데이터베이스 내용을 XML 문서로 생성하여 유무선 컨텐츠를 사용하는 사용자에게 일관된 정보를 제공하거나 기업간 정보 교환시 유용하게 활용될 수 있다.

### I 서론

무선통신 기술의 발달로 무선 인터넷에 접속하는 인구가 증가하고 있지만 무선 인터넷 접속 단말기들은 낮은 데이터 전송속도, 작은 화면 크기의 제약 등으로 인해 기존의 웹 컨텐츠를 표현하기 어렵다. 이러한 한계로 인해 웹 컨텐츠 제공자는 기존 컨텐츠를 모바일 단말기에서 유사한 서비스를 제공하기 위해서는 각 모바일 단말기가 사용하는 WML, HDML, m-HTML 등의 무선용 마크업 언어에 의존하여 컨텐츠를 제작해야 한다. 하지만 사용자에게 제공되는 정보는 끊임없이 변화하기 때문에 이를 정보에 대해 서로 다른 마크업 언어로 일일이 수작업하는 것은 불필요한 비용을 소모하는 것이다[9].

본 연구에서는 데이터베이스 내용을 웹 컨텐츠나

모바일 기기의 마크업 언어에 적합하도록 중간에 XML 문서를 자동 생성하여 컨텐츠 제공자가 웹과 모바일 단말기상에서 실시간으로 정보 서비스를 제공할 수 있게 한다. 본 연구 결과를 이용하면 컨텐츠 제공자는 웹 컨텐츠와 무선 컨텐츠를 구별하지 않고 XML을 이용하여 일관되게 정보 서비스를 제공할 수 있다.

XML 문서와 데이터베이스간 맵핑은 테이블 기반 맵핑과 객체-관계 맵핑으로 분류할 수 있다. 테이블 기반 맵핑 방법은 XML 문서를 테이블들의 집합으로 간주하는 방식으로서, 변환이 쉽고 속도가 빠르며 확장성이 높은 반면에, 간단한 XML 문서에만 적용될 수 있는 단점이 있다. 객체-관계 맵핑은 XML 문서를 객체들의 트리 형태로 간주하여, 이 객체들을 데이터

베이스로 맵핑하는 방식을 사용하여 보다 자유로운 형태의 XML 문서를 생성할 수 있다[5]. 본 연구에서는 이러한 객체-관계 맵핑을 기반으로 데이터베이스로부터 객체를 통해 XML 문서를 자동적으로 생성하는 도구를 구현하였다.

2장에서는 본 연구와 관련된 기존 연구들에 대하여 기술하고, 3장에서는 본 연구에 적용된 맵핑에 대해 기술하고, 4장에서는 본 연구에 의해 구현된 변환 시스템에 대해 기능을 상세히 기술하고, 5장에서 결론을 맺는다.

## 2. 관련 연구

본 연구와 관련된 JDO(Java Data Objects), Castor, Hibernate, DB2XML등에 대해 비교한다.

SUN에서 개발하는 차세대 기술인 JDO[1]는 Java 언어를 이용해 객체에서 데이터베이스와 인터페이스 할 수 있는 기술로서 기존 JDBC와는 다른 방법으로 보다 쉽게 데이터베이스에 접근할 수 있게 한다. JDO는 표준화된 API로서 자바 개발자에게 객체에서 데이터베이스에 대한 접근을 제공한다. 반면, 데이터베이스와 객체를 맵핑하기 위한 표준이 정해지지 않아 객체에서 데이터베이스를 접근하는 방법이 여러 가지가 존재하여 객체와 데이터베이스간 관계 표현이 문제로 야기될 수 있다.

Castor[8]는 오픈 소스 진영에서 개발된 것으로 자바에서 데이터 바인딩 프레임워크를 지원한다. Castor XML과 Castor JDO로 구성되며 XML 스키마에서 객체를 생성하거나 생성된 객체를 XML 문서로 생성할 수 있고, JDO를 이용하여 보다 간편하게 데이터베이스와 객체간 맵핑을 지원한다. 또한, SQL과 OQL을 지원하여 데이터베이스 내용을 객체에 저장할 수 있고 그 반대로 가능하다. 하지만 데이터베이스와 객체를 맵핑하기 위한 기술이 JDO를 기반으로 하기 때문에 JDO처럼 객체와 데이터베이스간의 관계가 개발자에 따라 표현이 다르다는 문제를 내포하고 있다.

Hibernate[7]는 오픈 소스 진영에서 관계형 데이터베이스와 자바 객체와의 맵핑(Object-Relational Mapping)을 지원하는 API들과 맵핑을 설정하는 맵핑 도구를 제공하며, 데이터베이스에 저장된 내용을 객체로 생성하거나 객체에 저장된 내용을 데이터베이스에 저장할 수 있다. OQL(Object Query Language)을 사용한다. Hibernate에서 맵핑은 JDO나 Castor처럼 개발자에 따라 서로 다른 맵핑 규칙을 적용하는 것이 아니라 XML 문서 형식의 MappingFile에서 모든 맵핑 관계를

표현하여 관리하기 때문에 객체간의 관계 표현 및 데이터베이스에서의 관계 표현을 일관되게 처리할 수 있는 장점이 있다. 그리고 객체와 데이터베이스간 맵핑을 효율적으로 지원하는 맵핑 도구를 제공한다.

본 연구와 유사한 DB2XML[3]는 JDBC 드라이버를 이용하여 관계형 데이터베이스로부터 XML 문서와 DTD를 자동 생성하여 웹에서 데이터베이스를 접근할 수 있게 한다. XML 문서 형태를 규정하기 위한 여러 설정 기능을 제공하고 일반적인 SQL문을 이용하여 데이터베이스의 내용을 XML 문서로 생성한다. 하지만 자유로운 형태의 XML 문서를 생성해 내기 어려운 한계로 인하여 데이터베이스간 단순한 자료 전송이나 단순한 형태의 자료를 웹에 표현하기 위해 사용된다.

## 3. 맵핑 관계

본 절에서는 XML 문서와 관계형 데이터베이스간 맵핑시 야기되는 문제들과 사용되는 맵핑 종류에 대하여 기술한다.

### 3.1 맵핑 문제점

XML 문서와 관계형 데이터베이스간 맵핑은 두 가지 모델이 서로 다른 배경을 가지고 있기 때문에 몇 가지 문제가 야기된다. 기본적으로 3계층 모델인 관계형 데이터베이스 모델은 XML 데이터 모델과 상당히 다르다. 데이터베이스는 레코드들로 구성된 테이블의 집합이고, 레코드들은 다시 다수의 필드들로 구성된다. 레코드 필드 값들은 값 영역에 속하고 하나의 값만 가져야 한다.

반면 XML 문서는 무한정의 깊이를 갖는 트리 구조이다. XML 문서가 트리 구조임에도 불구하고 무한정의 이름 공간(name space)을 가지고 있어 같은 이름 공간내 요소 형(Element Type)이 중복되어 선언되지 않으며, 모든 요소는 같은 영역에 선언되는 경우 서로 다른 이름을 가지게 된다. 요소 형은 속성(attribute)을 선언할 수 있고 다른 요소들이 일반적으로 속성명으로 사용된다. 따라서 모든 테이블명은 다르고 서로 다른 테이블인 경우 동일한 필드명을 사용할 수 있는 관계형 모델의 제한(restriction)과 유사하다. 따라서 테이블내 각 행은 요소에 대응시키고 애트리뷰트 값은 필드 값에 대응시키는 것이 가능하다.

XML 문서와 관계형 데이터베이스간 맵핑시 또 다른 문제점은 데이터형(Data Types). 이진 데이터(Binary Data), Null 데이터, 문자 셋(Character Set) 등과 같이

객체와 XML 문서간의 차이로 인해 문제가 야기될 수 있다. 이러한 문제에 대해서는 XML 문서 생성시 사용자가 선택적으로 관련 사항들을 선택할 수 있게 해야 한다[6].

### 3.2 맵핑 종류

일반적으로 XML 문서 스키마를 데이터베이스 스키마로 매핑하기 위해, 테이블-기반 매핑과 객체-관계형 매핑 등 두 가지 매핑이 사용된다[5,6].

테이블-기반 매핑은 XML 문서를 하나의 테이블 또는 테이블 집합으로 모델링한다. 즉, XML 문서 구조는 반드시 <database> 요소와 <table> 요소의 반복으로 구성된다. 테이블-기반 매핑은 두 관계형 데이터베이스 간에 데이터를 전송하는 경우처럼 관계형 데이터를 연속적으로 처리할 때 유용하다. 테이블-기반 매핑의 단점은 이러한 매핑이 아닌 XML 문서에 대해서는 사용될 수 없다는 점이다.

객체-관계형 매핑은 XML 문서내 데이터간 관계를 객체들의 트리로 모델링한다. 이 모델에서 속성, 요소 컨텐트, 또는 혼합된 컨텐트(복잡한 요소형) 등의 요소형은 일반적으로 클래스로 모델링한다. PCDA TIA만의 단순한 요소형, 속성, 그리고 PCDATA 등을 갖는 요소는 스칼라 속성(scalar properties)으로 모델링된다. 이 모델은 전통적인 객체-관계형 매핑 또는 SQL 3 객체 뷰를 사용하는 관계형 데이터베이스로 매핑된다. 즉, 클래스들은 테이블로 매핑되고, 스칼라 속성은 필드로 매핑되고, 객체-값 속성(object-valued properties)은 기본키/외부키(primary key/foreign key)의 쌍으로 매핑된다.

## 4. 시스템 구조

### 4.1 시스템 구성도

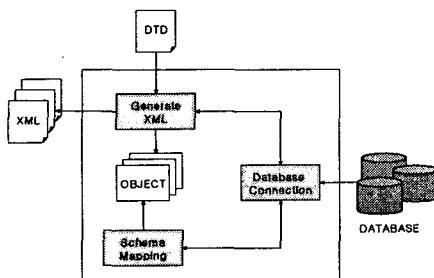


그림 1 시스템 구성도

본 연구에 의해 구현된 시스템의 구성도는 그림 1과 같이 크게 DatabaseConnection부, SchemaMapping부,

GenerateXML부 등 세 부분으로 구성된다. DatabaseConnection부에서 연결할 데이터베이스를 설정하고 SchemaMapping부에서는 DatabaseConnection부에서 설정한 데이터베이스에 접근하여 XML 문서를 생성할 테이블을 자바빈 형태로 생성한다. 그리고 마지막으로 GenerateXML부에서는 자바빈 객체와 데이터베이스 내용을 매칭하여 DTD와 결합하여 요구된 형태의 XML 문서를 생성한다.

### 4.2 클래스 구성도

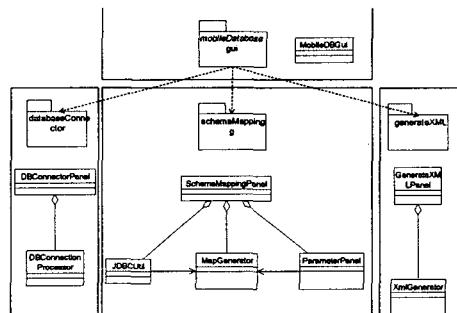


그림 2 시스템의 클래스 구성도

그림 2는 본 시스템을 구현한 클래스들간 구성도를 보여준다.

#### 1)MobileDatabaseGUI

이 클래스에서는 전체 윈도우 화면을 구성하고 DBConnectorPanel, SchemaMappingPanel, GenerateXMLPanel들을 포함한다.

#### 2)DatabaseConnector

DBConnectorPanel 클래스에서는 데이터베이스에 관한 설정들을 입력/수정할 수 있는 화면을 구성하고, DBConnectionProcessor 클래스에서는 DBConnectorPanel에서 설정된 값을 이용하여 데이터베이스 연결에 사용할 property 파일을 생성한다.

#### 3)SchemaMapping

SchemaMappingPanel 클래스에서는 데이터베이스 테이블과 대응되는 빙 객체를 생성하고, JDBCUtil 클래스에서는 databaseConnector 패키지에서 생성된 property 파일을 가지고 데이터베이스에 접근하여 테이블에 해당하는 자바빈 클래스를 생성할 수 있는 환경을 제공한다. SchemaMappingPanel과 ParameterPanel 클래스에서 설정된 값들은 MapGenerator 클래스에서 빙 객체, 데이터베이스와 빙 객체간 맵핑 정보에 관한 파일을 생성한다.

#### 4)GenerateXML

GenerateXMLPanel 클래스에서는 데이터베이스에 대해 질의를 수행한 후 이를 XML 문서로 생성하는 화면을 제공하고 XMLGenerator 클래스에서는 사용자가 등록한 DTD 형태에 따라 최종적으로 XML 문서를 생성한다.

#### 4.3 구현

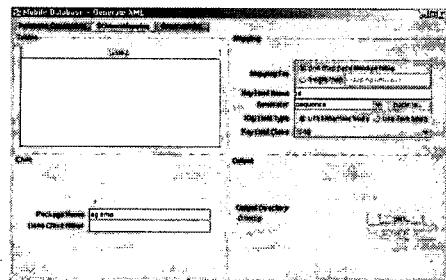


그림 3. 시스템 화면

그림 3에서 Database Connection탭은 데이터베이스에 접속하기 위한 JDBC 드라이버, 데이터베이스 URL, 사용자 이름, 사용자 암호 등을 설정한다. 이 그림은 주로 데이터베이스의 테이블로부터 자바빈 객체의 클래스 파일을 매핑하는 SchemaMapping을 보여준다.  
이 탭에는

1)Tables Panel: 빈 클래스 파일의 생성 및 맵핑하기 위한 데이터베이스 및 테이블 목록을 가져온다.

2)Mapping Panel: 빈 객체와 데이터베이스와의 맵핑 관계를 표현한 맵핑 파일에 대한 생성 타입을 설정하고 생성될 빈 객체의 속성들을 설정한다.

3)Code Panel: 생성될 빈 클래스들의 패키지명, 생성될 빈 클래스들의 기본 클래스명을 입력한다.

4)Output Panel: 빈 클래스 파일의 생성 경로 설정 및 빈 클래스 파일을 생성한다.

그리고, GenerateXML 탭은

1)Database Query Panel: 생성할 XML 문서 내용에 적합한 질의문을 작성한다. 본 시스템에서는 Hibernate 질의어(Hibernate Query Language : HQL)를 사용한다.

2)Output Panel: 생성될 XML 문서 이름 입력, 디렉토리 위치 설정, XML 문서를 생성 등을 명시한다.

본 연구에서는 객체와 데이터베이스간 맵핑은 Hibernate API를 주로 활용하였고 객체로부터 XML 문서를 생성하는 과정에서는 JDOM API를 주로 활용하였다[1, 2, 3, 4, 5].

#### 5. 결론

본 연구는 데이터베이스 내용을 웹 컨텐츠나 모바일 단말기의 마크업 언어에 적합하도록 중간에 XML 문서를 자동 생성하여 컨텐츠 제공자가 웹과 모바일 단말기상에서 실시간으로 정보 서비스 제공시도움을 주고자 연구되었다. XML 문서와 데이터베이스간 맵핑에는 객체-관계 맵핑을 적용하여 XML 문서를 객체들의 트리 형태로 간주함으로써. 보다 자유로운 형태의 XML 문서를 생성할 수 있는 기반을 마련하였다. 본 연구에서는 이러한 객체-관계 맵핑을 기반으로 데이터베이스로부터 객체를 통해 XML 문서를 자동적으로 생성하는 도구를 구현하였다.

본 연구 결과는 기존의 유선 컨텐츠 기업이 무선 컨텐츠 서비스를 하거나 기업간 데이터 교환시 유용하게 사용될 수 있다. 향후에는 데이터베이스 내용을 변경하는 질의문 처리, 기존 웹 애플리케이션 서버에 효율적인 탑재, 데이터베이스간 자료 전송시 DTD를 이용한 전송 등에 대한 연구가 요구된다.

#### 참고문헌

- [1] Nicholas Chase, *XML and Java from Scratch*, QUE, 2001.
- [2] Akmal B. Chaudhri, Roberto Zicari, *Succeeding with Object Databases*, John Wiley & Sons, 2001.
- [3] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto, *XML and Java*, Addison Wesley, 1999.
- [4] Liam Quin, *Open Source XML Database Toolkit*, John Wiley & Sons, 2000.
- [5] 최종명, 유재우, 최재영, 자바 개발자를 위한 XML, 홍릉과학출판사, 2003.
- [6] Ronald Bourret, *XML and Databases*, <http://www.rpbourret.com/xml/XMLAndDatabases.htm>
- [7] Hibernate, <http://www.hibernate.org>
- [8] Castor, <http://www.exolab.org>
- [9] WML, <http://www.wapforum.org/>
- [10] JDO, <http://java.sun.com/products/jdo/>
- [11] DB2XML, <http://www.informatik.fh-wiesbaden.de/~turai/DB2XML/index.html>