Track&Trace 시스템의 서버와 XML 데이터베이스의 구현

유정순*, 하수철*, 박주상**, 황재각**
*대전대학교 컴퓨터공학과
**한국전자통신연구원 우정기술연구센터

Implementation of Server and XML Database for the Track&Trace System

Joung-Soon Yoo*, Soo-Cheol Ha*, Joo-Sang Park**, Jae-Gak Hwang**
*Dept. of Computer Engineering, Daejeon University
**Postal Technology Research Center, ETRI
E-mail : *jsyoo@zeus.dju.ac.kr, *soocha@dju.ac.kr, **kappa@etri.re.kr, **jghwang@etri.re.kr

요 약

본 논문은 물류 Tracking 시스템[1]의 서버와 XML 데이터베이스 구현에 관한 사항이다. 서버는 XML(데이터), XSL(표현방법), JSP(로직)의 3가지 형태로 분리함으로써 관리 및 개발 생산성을 향상시키도록 한다. 공간 낭비와 비효율성을 해결하기 위하여 반구조적인 데이터 모델이며, 트리 구조 형식인 XML을 XML 전용 데이터베이스인 오라클에 사상하여 저장한다. 그 결과 매핑 정보를 알기 때문에 XML 문서 전체를 파싱할 필요가 없어 보다 효율적으로 질의가 가능하다.

1. 서론

다양한 분야에서 XML 문서의 사용이 급격히 증가하고 있고, 이를 영구적으로 데이터베이스에 저장해야 할 필요성이 증가하고 있다. XML 문서는 자기 기술적인 특성(self-describing), 유니코드에 따른 이식 용이성(portability), 트리 혹은 그래프 구조로 데이터를 기술할 수 있다는 이점을 갖는 데이터베이스 포맷으로, XML 데이터 자체를 파일로 저장해서 일종의 데이터로 사용할 수도 있다[2].

그러나, 데이터 크기가 커진다면 문제가 발생하게 된다. 이로 인해 원하는 데이터를 얻기 위해서 모든 XML 데이터를 모두 메모리에 올리거나 매번 파싱해야 하는 문제가 발생한다[3].

XML은 반구조적인 데이타 모델이며, 트리 구조 형식으로 이루어져 있다. 따라서, XML은 순서 정보와 반복 정보가 있다. 반면에, RDBMS는 구조적인 데이타 모델이며, 관계 대수이므로 비순서적이며 비반복적이다[4].

본 논문은 한국전자통신연구원(ETRI) 위탁과제 지원을 받아 수행된 것입니다.

기존의 RDBMS는 OR-Mapping을 이용하여 XML 문서를 저장한다. 이로 인해 공간 낭비와 효율성 문제가 발생하며 속도가 느린 반면에, Native XML 데이터베이스는 이와 같은 문제를 해결한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구로 JSP, XML, XML 데이터베이스에 관한 이론적 배경을 알아보고, 3장에서는 Track&Trace 시스템의 서버를 구현한다. 4장에서는 Track&Trace 시스템의 데이터베이스를 구현하며, 마지막 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

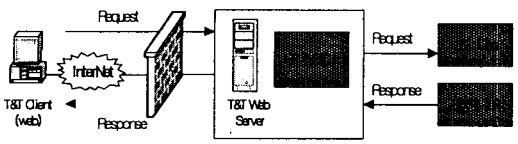
2.1 JSP(Java Server Pages)

JSP는 동적 웹컨텐츠를 간편하고 빠르게 개발할 수 있도록 해주는 Java 기술로 서버와 플랫폼에 독립적인 웹 기반 애플리케이션을 빠르게 개발할 수 있다.

또한, 개발자의 역할과 디자이너의 역할을 완전하게 분리함으로써 서로의 작업을 보다 작게 분할하고, 각

자의 작업에 의해 생성된 결과를 다시 합칠 수 있도록 해준다[5].

JSP의 동작 과정은 그림1과 같다. 클라이언트가 웹서버에 JSP 페이지를 요청하면 웹서버의 JSP 엔진에서 자동으로 서블릿 코드로 변환을 시키고 컴파일을 한다. 그리고, 그 컴파일한 클래스를 실행하고, 그 결과를 클라이언트의 웹 브라우저에 보낸다.



[그림1] JSP의 동작 과정

2.2 XML과 데이터베이스

XML의 사용을 위한 데이터베이스는 크게 XML-Enabled 데이터베이스와 Native XML 데이터베이스로 분류된다. XML-Enabled 데이터베이스는 XML을 RDBMS에 저장하는 방법으로 XML 데이터를 내부적으로 XML이 아닌 다른 형태로 변환해서 저장하는 것이다.

또한, Native XML 데이터베이스는 XML을 XML 전용 DBMS에 저장하는 방법으로 XML의 구조를 그대로 유지하면서 XML 형태 그대로 데이터베이스 내부에 저장하는 것이다[6][7].

2.3 Oracle XML 데이터베이스

Oracle XML 데이터베이스는 RDBMS의 기술과 XML 기술을 통합한 XML 전용 데이터베이스로 고성능의 XML 저장 및 획득 기술을 제공하며 전용 Repository를 통해 인터넷 기반의 다양한 프로토콜을 통해 XML 리소스에 대한 접근 방식을 제공한다. 전용 Repository는 XML 중심의 파일 시스템으로 데이터베이스에 산재하는 모든 XMLType 객체를 인터넷 근간의 저장 방식으로 매핑 한다.

2.3.1 XML Type

XMLType은 XML 컨텐츠가 저장되는 Native Data Type이다. 일반 테이블의 컬럼에 하나의 형식으로 사용된다. XMLType과 XML 스키마 기반 Object일 경우, Oracle9i 데이터베이스의 CLOB로 저장되거나 XML 스키마에 설정된 데이터 형식에 맞는 Object-Relational 형식의 테이블이나 뷰로 저장될 수 있다. 그러나, XML 스키마에 따르지 않는 Object인 경우에는 기본 저장 영역인 CLOB에만 저장이 가능하다[8].

2.3.2 XML 스키마 확인(Validation)

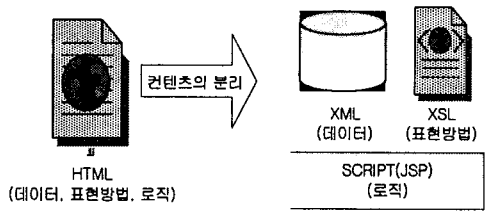
Oracle XML 데이터베이스에 XML 스키마를 저장하는 방법에는 소유자(Owner)만 사용할 수 있는 Local XML 스키마와 모든 사용자가 사용할 수 있는 Global XML 스키마가 있다.

XML 스키마 확인은 네트워크 시스템 상에서 XML 데이터를 교환하거나, 전송 문서에 대한 검증이 필요할 때 사용하며 3가지 방법이 있다. 첫째로, 테이블 생성 시에 스키마와 Element를 지시함으로써 Element의 구조를 검증하는 방법이 있다. 둘째로, 테이블 Constraint를 통해 SQL Function, XMLIsValid()를 사용함으로써 XML 스키마에 대한 Full-Validation을 제공하지만 오류 발생 시 자세한 내용을 볼 수 없다. 셋째로, Trigger를 통해 schemaValidate() Procedure를 사용한 Full-Validation을 제공하며 모든 오류의 확인이 가능하다[9].

3. Track&Trace의 서버 구현

3.1 개요

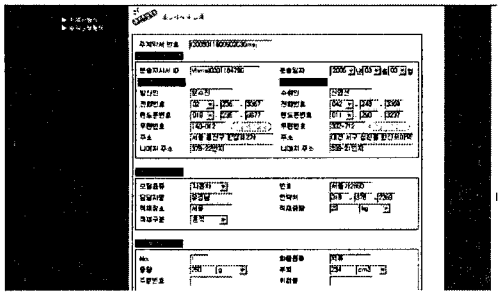
Track&Trace 시스템의 서버 구조는 그림2와 같다. 하나의 HTML은 데이터, 표현방법, 로직을 포함한다. 이것을 XML(데이터), XSL(표현방법), JSP(로직)의 3가지 형태로 분리함으로써 관리 및 개발 생산성을 향상시킬 수 있게 한다.



[그림2] Track&Trace의 서버 구조

3.2 Track&Trace의 화물 종추적 등록

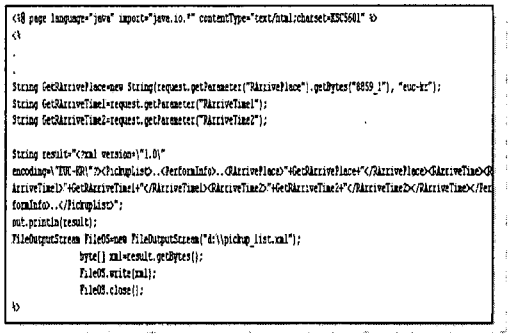
그림3은 Track&Trace 시스템의 화물 종추적 등록 화면으로 운송지시서를 입력하는 화면이다. 입력을 완료한 후 입력 완료를 클릭 하면, 자바 스크립트의 메소드를 호출하여 입력하지 않은 부분이 있는지 확인한 후 사용자 입력 내용을 POST 방식으로 JSP 파일로 넘겨준다.



[그림3] 화물 종추적 등록(운송지시서)

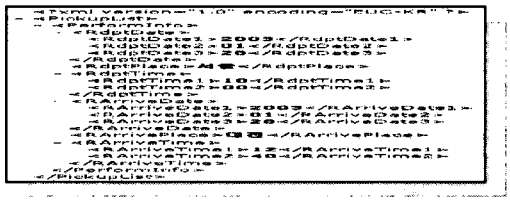
3.3 JSP를 이용한 XML 생성

JSP를 이용한 XML 생성은 그림4와 같으며, Java의 입출력을 사용하기 위해 "java.io.*"를 import 한다. 폼 형태로 넘어온 사용자 입력을 request 객체를 이용하여 저장하고 XML을 생성하며, 한글 입력의 경우 한글 처리를 한다.



[그림4] JSP를 이용한 XML 생성

그 결과 생성된 XML을 웹브라우저에서 출력한 결과는 그림5와 같다.



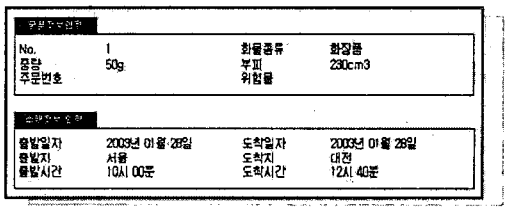
[그림5] 웹브라우저에서 XML 출력 결과

3.4 서버의 XSL 설계

XSL은 XML을 사용하고 있는 웹을 통해 전송되는 데이터가 사용자에게 어떻게 보여질 것인지를 나타내는 StyleSheet이며 그림6과 같다.

[그림6] XSL 설계

생성된 XML에 XSL을 적용하여 웹브라우저에 출력한 결과는 그림7과 같다.

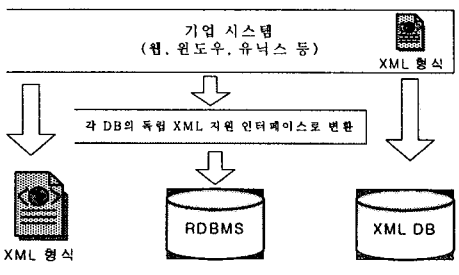


[그림7] 웹브라우저에서 XML 출력 결과

4. Track&Trace의 데이터베이스 구현

4.1 개요

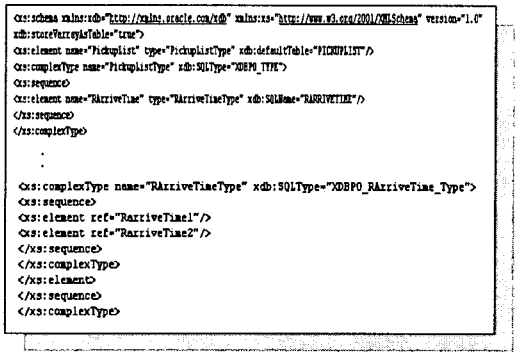
OODB에 XML을 저장할 수도 있으나 XML 문서 형태로 저장을 원한다면 Native XML 데이터베이스를 선택하는 것이 좋으며 XML의 저장 유형은 그림8과 같다. Native XML 데이터베이스는 물리적인 구조로 저장되며, 데이터의 손실 없이 빠른 속도로 저장할 수 있으며, W3C의 표준인 XPath를 이용하여 효율적인 질의가 가능하다.



[그림8] XML 저장 유형

4.2 XML 스키마(XSD) 설계

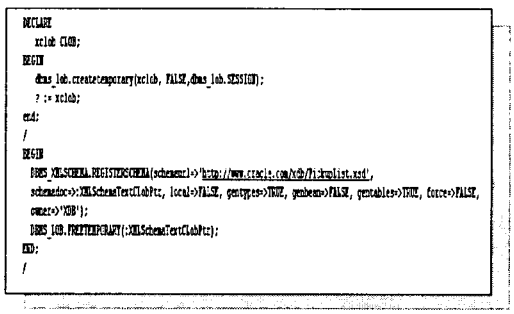
XSD(XML Schema Definition)는 XML 문서의 유효성을 검사할 수 있는 방법이며 그림9와 같다. 또한, MicroSoft사의 XDR(XML Data Reduced)이 있으나 W3C 표준과는 호환되지 않으므로 본 시스템에서는 XSD를 사용하여 구현한다.



[그림9] XML 스키마(XSD) 설계

4.3 Oracle에 XML 스키마 저장

XML 스키마가 저장될 때 URL로 문서로 지정된 경우 등록하는 방법은 그림10과 같으며, DBMS_XML SCHEMA 패키지를 사용하여 등록 및 생성, 삭제가 가능하다. 또한, 등록 시에는 맵핑되는 객체 타입과 디폴트 테이블을 생성하며 선택적으로 JavaBeans도 만들 수 있다.



[그림10] Oracle에 XML 스키마 저장

4.4 XMLType의 테이블 생성

XML 스키마가 등록되면 각각의 Element와 1:1로 대응되는 Object-Realtional Type이 생성되며, XML 문서의 각 Element들이 파싱된 형태로 데이터베이스에 저장된다.

그림 11은 XMLType을 위한 테이블 생성을 나타내며, 그림 12는 StyleSheet를 저장하기 위한 테이블 생성을 나타낸다.

```
CREATE TABLE trace (id NUMBER, col XMLTYPE);
```

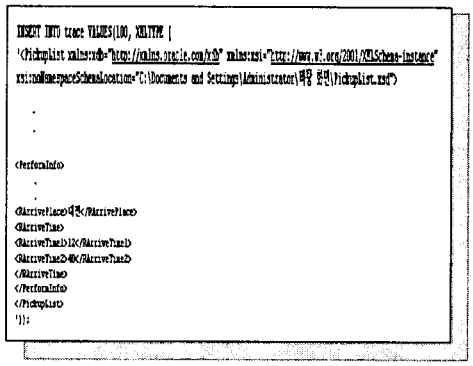
[그림11] XMLType의 테이블 생성

```
CREATE TABLE style_sheet (id NUMBER, sheet XMLTYPE);
```

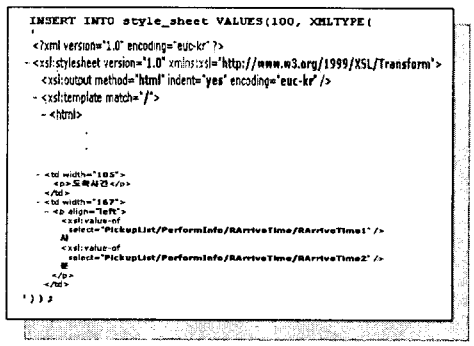
[그림12] StyleSheet를 위한 XMLType 테이블 생성

4.5 XML 문서와 StyleSheet의 저장

그림13은 생성된 XML 문서를 XMLType으로 생성한 테이블에 저장하는 것이며, 그림14는 StyleSheet를 저장하는 것이다.



[그림13] XML 데이터의 저장



[그림14] StyleSheet의 저장

4.6 화물 종추적 검색

XPath는 요소와 속성의 리스트들을 '/'로 구분하여 특정 요소를 찾을 수 있도록 하는 기능을 제공하며 절대 경로와 상대 경로로 지정할 수 있다[9]. 절대 경로는 Root Element부터 정의하며, 상대 경로는 현재

검색한 위치의 Element를 기준으로 하위의 Child Element를 정의한다.

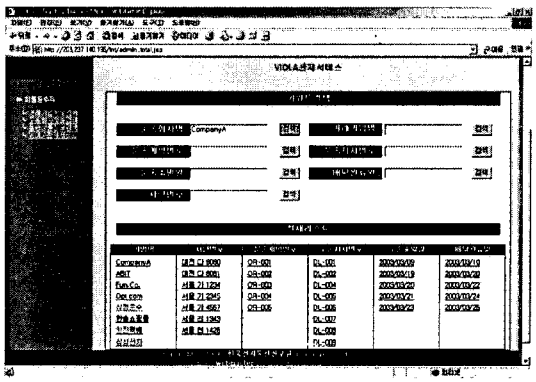
W3C의 XPath 표준을 따르는 Oracle XML 데이터베이스에 의해 제공되는 몇 가지 함수가 있으며 그림15에서는 existsNode() 함수를 사용하여 검색한다. existsNode() 함수는 문서 내에 W3C의 XPath 표현식을 따르는 노드를 포함하는 문서가 있는지 없는지 평가하는 함수이다[9].

```
SELECT count(XMLCOL)
FROM ex
WHERE EXISTSNODE(XMLCOL, '/Pickuplist[User="ANN"]')=1
/
```

[그림15] XPath와 existsNode() 함수를 이용한 검색

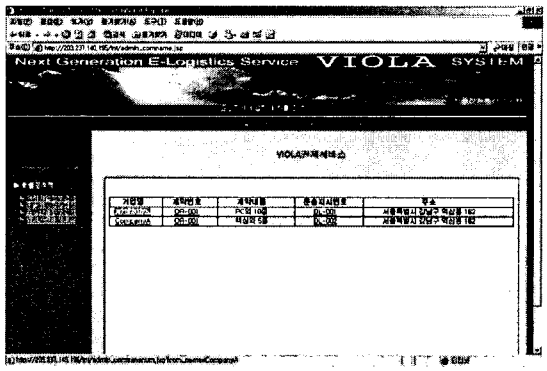
4.7 질의 예

그림16은 Track&Trace의 화물 종추적 검색 화면이며 관리자의 입장에서 전체 리스트를 보여주거나 키워드 검색으로 원하는 정보를 검색할 수도 있다. 운송회사명을 입력하고 검색을 클릭 한다.



[그림16] 화물 종추적 검색

그 결과 운송회사명에 해당하는 내용을 출력하며 그림17과 같다.



[그림17] 화물 종추적 결과

5. 결론

본 논문은 물류 Tracking 시스템의 서버와 데이터베이스 구현에 관한 것으로 HTML로 입력된 것을 XML로 변환하고, XSD와 Oracle의 XML 데이터베이스를 이용하여 저장하며, XSL을 이용하여 웹브라우저를 통해 출력하도록 하였다.

구현된 데이터베이스를 이용하면 검색 시에 매핑 정보를 알기 때문에 XML 문서 전체를 파싱할 필요가 없어 보다 효율적으로 질의가 가능하다.

[참고문헌]

[1] 하수철, "실시간 물류 Tracking 기술에 관한 연구", 최종 보고서, 한국전자통신연구원, 2003

[2] 박성진, "XML 데이터베이스", 한국 인터넷 정보학회, 2권 3호, 2001

[3] 허준회, 이민우, 김종민, 최한석, "Jump TO XML web programming", 정보게이트, 2002

[4] 최현묵, "오라클 iAS와 DB에서의 XML 프로그래밍", Oracle Korea Magazine, 2002

[5] 박용우, "e-Technology의 기반 플랫폼 J2EE", Oracle Korea Magazine, pp.28-45, 2002

[6] 김영숙, 조성호, "XML Bible", 삼양출판사, 2001

[7] 이경하, 정명희, 홍의석, "실전 XML 데이터베이스 구축", 성안당, 2002

[8] 김은희, "웹서비스를 위한 XML과 DB의 랩소디인 블루, 오라클 XML DB", 프로그램세계, pp.4-19, 2002년 8월호(스페셜 리포트)

[9] 이진호, "Oracle XML DB 로드쇼", 한국 오라클, pp.1-96, 2002