

XML 기반의 이기종 DBMS간 데이터 복제 웹 에이전트 설계 및 구현

유 선 영* · 임 재 홍**

*한국해양대학교 대학원, **한국해양대학교 전자통신공학과 교수

Design and Implementation of Data Replication Web Agent between Heterogeneous DBMSs based on XML

Sun-Young Yu* · Jae-Hong Yim**

*Graduate school of Korea Maritime University, Pusan 410-4718, Korea

**Department of Electronics & Communication Engineering, Korea Maritime University, Pusan 410-4318, Korea

요 약 : 최근 인터넷에서 사용되고 있는 하이퍼텍스트 생성 언어(HTML ; Hyper Text Markup Language)는 제한된 태그(Tag)들을 사용하기 때문에 문서를 구조화시키지 못해 정보 축적과 정보추출 방법이 비효율적이고 추출한 정보의 가공이 어렵다. 그러나 확장성 생성 언어(XML ; eXtensible Markup Language)는 사용자가 문서상에 사용될 태그를 자유롭게 정의할 수 있고 다른 사람들도 그 태그를 사용할 수 있기 때문에 HTML에 비해 정보 축적과 정보추출 방법이 용이하다. 서로 다른 데이터베이스를 사용하고 있는 항만 관련 업체들간의 물류에 관한 정보를 공유하거나 교환하기 위해서는 HTML이 아닌 XML이 더 적합하다. 본 논문에서는 항만업체들의 요구를 수용하기 위해 XML을 이용하여 웹에서 이기종 데이터베이스 관리 시스템(DBMS ; Database Management System)간의 데이터 복제 및 정보를 교환할 수 있는 웹 에이전트 시스템을 설계하고 구현하였다.

핵심용어 : 하이퍼텍스트 생성 언어, 확장성 생성 언어, 데이터베이스, 데이터베이스 관리 시스템

Abstract : Since current HTML used on the internet is to use restricted tag, it is not easy to store information and extract data from information of document. XML defined newly tag and is easy to store information and extract data from information. So XML is easier to transact information rather than HTML. XML is suitable for enterprise's requirement needs data exchange between heterogeneous databases. This paper proposes web agent for data replication between heterogeneous DBMSs(Database Management System). Web agent system manages database on the web and exchange data in heterogeneous database using XML. Then we designed and implemented for web agent of data replication between heterogeneous DBMSs.

Key words : HTML(Hyper Text Markup Language), XML(eXtensible Markup Language), Database, Database Management System

1. 서 론

우리 나라는 선진외국에 비해 상대적으로 물류비의 부담이 과중하여 국제경쟁력을 저해하고 있다. 이러한 물류부문의 비효율성을 초래하는 요인에는 여러 가지 문제가 지적되고 있으나 물류를 원활하게 처리할 수 있는 항만, 공항, 도로, 철도 등 사회간접 자본시설의 부족이 가장 심각한 당면 문제로 지적되고 있다(김중철, 1996). 개선 효과는 크지만 투자 비용과 많은 시간이 소요되는 사회 간접자본의 확충보다는 물류 부문의 경영혁신을 통해 물류정보의 질적 혁신과 합리적인 물류정보시스템 개발이 필요하다.

현재 국내에서도 이러한 점을 인식하고 경쟁력을 제고하기 위한 노력의 일환으로 물류망(KL-Net)과 무역망(KT-Net)을

이용한 전자문서교환(EDI) 방식을 바탕으로 한 항만 물류 시스템이 구축되어 있다(장홍훈, 1995).

우리 나라의 항만 물류 시스템에서 이루어지는 수출입과정 EDI는 KL-Net을 통하여 모든 업무가 이루어지고 있다. 즉, 수출입과 관련된 모든 정보가 KL-Net을 통해 전달된다는 것이다. 모든 수출입 관련 정보가 KL-Net을 거치므로 이 시점에서 정보를 획득하여 데이터베이스에 저장하고 가공·처리한다면 선박에 실린 화물의 견적/발주/수급에 대한 일괄처리 및 이동상황과 선박과 육상간의 실시간 정보교환으로 선박의 최신 선박동정과 자재 보급관리 시스템에 대한 자동화 등 양질의 서비스를 제공할 수 있을 것이다. 현재 KL-Net에서도 데이터베이스를 이용한 서비스를 일부 제공하고 있지만 이는 각 기관의 데이터베이스를 연결하는 수준이다(노홍승, 1996).

국내의 항만 물류 관련 정보 시스템들은 KL-Net의 데이터베이스 활용 부재와 각 관련 기관과 업체들이 자체 데이터베이스를 독립적으로 보유함으로써 야기되는 문제점과 국내의

* 정희원, niceysy@kmaritime.ac.kr 051)410-4718

** 정희원, jhyim@kmaritime.ac.kr 051)410-4318

EDI 사용자 시스템은 도스환경을 기반으로 하고 있어서 윈도우 환경과 인터넷 환경에 익숙한 사용자들에게는 친밀감을 주지 못하고 있어 사용상의 불편함을 주고 있다. 따라서 현재의 KL-Net는 양질의 서비스를 제공하기 위해 물류 통합 데이터베이스 시스템을 추가해야 하고 운영환경을 보다 친숙하고 개방적인 환경으로 전환하기 위해서 인터넷을 활용해야 한다.

인터넷에서 정보 저장에 많이 사용되고 있는 HTML(Hyper Text Markup Language)은 표현위주의 언어이므로 정보 축적의 한계가 있고 문서를 분석하여 정보를 추출하는 방법이 비효율적이고 추출한 정보의 가공도 어렵다. 이러한 HTML의 단점을 보완하기 위해 제안된 XML(eXtensible Markup Language)은 태그를 정의하여 만들거나 새로 만들어진 태그를 사용할 수 있어서 문서의 확장이 용이하고 문서의 구조적인 표현이 가능하기 때문에 문서에서 정보를 추출하여 가공하기 쉽다. 따라서 서로 다른 데이터베이스를 보유하고 있는 기관과 업체간의 정보를 공유를 하기 위해서는 HTML보다는 XML이 적합하다(이강찬, 2001).

본 논문에서 제시한 이기종 DBMS(DataBase Management System)간의 데이터 복제 웹 에이전트 시스템은 서로 다른 데이터베이스간의 데이터 교환 및 복제를 위한 매개물로 XML을 사용하여 교환하고자 하는 데이터를 웹 브라우저를 통해 확인과 수정이나 삭제 작업을 한다. 웹 에이전트는 데이터베이스의 정보교환 뿐만 아니라 XML로 구성된 문서의 내용을 데이터베이스에 저장시킬 수도 있다. 또한 웹 상에서 이루어지는 정보교환이므로 데이터베이스 관리를 위한 명령어에 대한 지식이 없는 일반인도 쉽게 다룰 수 있다.

본 논문에서 구현된 웹 에이전트 시스템은 항만 물류정보 데이터베이스를 통합하고, 수출입 화물에 관련된 정보와 항만과 해운에 관련된 부가적인 정보를 제공함으로써 증가하는 항만관련 종사자들의 요구를 수용할 수 있다.

본 논문의 구성은 제 2 장에서 XML의 개요 및 특성에 대하여 정리하고, 제 3 장과 제 4 장에서는 실제로 구현한 웹 에이전트에 대한 설계와 구현 내용을 기술하고, 제 5 장은 결론과 향후 연구방향에 대하여 논한다.

2. 확장성 생성 언어(XML)

2.1 XML의 개요

1996년 4월 팀 브레이(Tim Bray)는 구조화된 문서를 위한 XML의 설계목표를 발표했고, 1996년 11월 보스턴에서 열린 SGML(Standard Generalized Markup Language) '96 컨퍼런스에서 처음 XML 초안이 발표되었다. 1998년 2월 10일에는 W3C 권고에 따르는 XML 스펙 1.0이 발표되고 오늘날까지 꾸준히 연구되어 오고 있다(W3C, 2001).

웹에서 구조화된 문서를 전송 가능하도록 설계된 XML은 표준화된 텍스트 형식의 생성 언어로 SGML의 일부분이며 SGML보다 간결하고 인터넷에서 사용 가능한 문서를 표현하

는 표준이다. XML은 전체를 지원하는 소프트웨어의 개발이 용이하지 않는 SGML과 제한된 태그로만 분류되어 지정되지 않은 태그의 사용이 불가능하다는 HTML의 단점을 극복하였다. 그리고 XML은 구조화된 문서를 정의하고 자유롭게 태그를 정의할 수 있는 SGML과 인터넷에서 손쉽게 하이퍼미디어 문서를 제공할 수 있는 HTML의 장점을 그대로 취합하고 있다.

HTML과 SGML의 필수적인 기능만을 취합하고 복잡하고 어렵거나 비효율적인 부분은 제외함으로써, XML은 두 언어의 핵심적인 장점을 그대로 보유하고 있다.

XML은 HTML과 같은 생성 언어(markup language)이지만, 정확한 뜻은 생성 언어를 정의하기 위한 언어이다.

XML이 HTML과 다른 점은 태그를 정의하고 데이터를 기술할 수 있다는 것이다. XML은 데이터의 관점에서 기술하고 구조화하며, HTML은 처리된 데이터를 표현하는 관점에서 이해할 수 있다. 즉 XML은 HTML의 대체품이 아니라 HTML의 한계를 극복한 것이다(장우혁, 2001)(박철현, 1999).

2.2 XML의 특징

XML은 웹에서 정보를 전달하기 위하여 제안된 메타 언어이고 XML을 지원하는 소프트웨어는 URL(Uniform Resource Locator)을 HTML의 링크를 접근하듯이 접근할 수 있는 기능이 내재되어 있다.

일반적으로 XML 인스턴스 중에서 링크와 DTD(Document Type Definition)는 URL의 사용이 가능해야 한다. 만약, 과거 및 기타 소프트웨어에서 URL을 지원하지 않는다면, XML 스펙에서 언급한 다양한 기능들이 무용지물이 된다. XML을 지원한다는 것은 인터넷 환경을 고려한다는 의미가 되므로 XML은 웹에 대한 활용이 높다.

3. 웹 에이전트 시스템 설계

본 논문에서 구현한 시스템은 사용자의 입장에서 별도의 다른 소프트웨어를 추가하지 않아도 웹에서 데이터베이스의 데이터를 저장, 검색, 관리하고, XML을 이용하여 서로 다른 데이터베이스간의 데이터 복제와 관리가 가능하다.

3.1 시스템 환경

웹 에이전트에서 사용자는 자바를 지원하고, XML 문서를 볼 수 있는 익스플로러 4.0 이상, 넷스케이프 4.0 이상의 브라우저를 사용한다. 웹 서버는 웹 프로그램의 솔루션으로 자바와 연동이 되고 이식성이 좋은 JSP(Java Server Page)를 이용한다.

JSP도 자바와 같이 데이터베이스와 프로그램을 연동시키기 위해 JDBC(Java DataBase Connectivity)를 사용한다. JDBC는 자바나 JSP 등으로 작성된 프로그램을 데이터베이스 안의 데이터들과 연결하기 위한 응용프로그램 인터페이스

(API)이다.

JDBC는 ODBC(Open DataBase Connectivity)와 유사한 형태로 모든 데이터베이스와 자바 프로그램을 연결시켜주는 연결자 역할을 한다. 본 논문의 웹 서버에는 데이터베이스마다 전용 드라이버를 설치하여 속도나 성능을 향상시키고, 환경에 구애받지 않고 폭넓게 사용할 수 있는 순수 자바 드라이버를 이용한다.

3.2 웹 에이전트 구성

본 논문에서 구현한 이기종 DBMS간 데이터 복제를 위한 웹 에이전트 시스템은 기존의 데이터베이스에서 이루어지던 데이터의 저장 및 삭제, 수정 등의 데이터 관리를 인터넷을 이용하여 언제, 어디서나 할 수 있다. 특히, XML을 매개물로 이용하여 서로 다른 DBMS간의 데이터를 웹에서 교환하는 것은 본 논문에서 구현한 시스템의 주요한 기능이다.

웹 에이전트의 전체 시스템 구성은 Fig. 1과 같이 웹에서 데이터베이스를 관리하는 과정과 XML 문서와 데이터베이스간의 정보를 교환하는 과정, 그리고 데이터베이스간의 데이터 복제과정으로 나누어진다.

웹에서 데이터베이스를 관리해야 하는 경우 사용자는 컴퓨터 브라우저를 통해 데이터베이스에 접근할 수 있는 아이디와 패스워드에 관한 정보를 웹 서버로 전송하며 웹 서버는 전송된 정보를 토대로 데이터베이스와 연결을 시도한다. 웹 서버와 데이터베이스의 연결이 이루어지면 사용자로부터 수정하고자 하는 데이터에 대한 추가 입력된 정보를 이용하여 데이터베이스에 저장된 데이터를 추가, 삭제 및 수정한다.

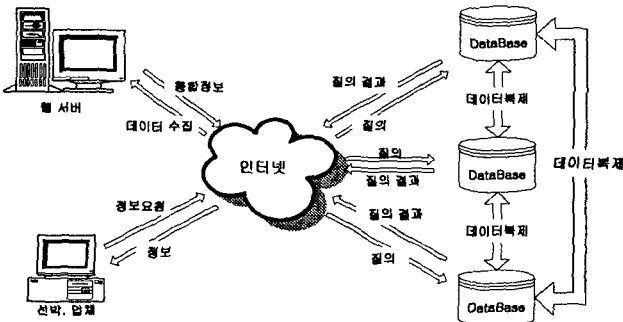


Fig. 1 Configuration of web agent system

사용자가 새로운 정보를 데이터베이스에 저장하고자 할 경우, 사용자는 정보를 XML 문서로 저장한 뒤 XML 문서의 위치에 대한 정보를 입력하여 웹 서버로 전송하면 웹 서버는 입력된 정보를 토대로 XML 문서에 접근하여 XML 문서의 내용을 웹 브라우저를 이용하여 보여준다. 사용자는 웹 브라우저를 통해 보여지는 내용을 확인하거나 수정을 한 다음 XML 문서의 내용이 저장될 데이터베이스에 대한 정보를 입력하면 웹 서버는 XML 문서의 내용을 데이터베이스에 저장한다.

서로 다른 데이터베이스간 데이터를 복제할 경우, 사용자가

해당 데이터베이스에 대한 정보를 입력하면 웹 서버는 입력된 정보를 이용하여 해당 데이터베이스와 연결되면 사용자는 복제하고자 하는 테이블을 선택한다. 웹 서버는 선택된 테이블의 내용을 데이터베이스로부터 전달받아 XML 문서로 사용자가 지정한 위치에 저장한다.

본 논문에서 구현한 시스템은 인터넷에 연결되어 있는 컴퓨터에 사용자가 브라우저를 통해 웹 서버에 접속하게 되면 언제 어디서나 데이터베이스를 관리할 수 있고, 웹 환경에 익숙한 사용자라면 누구라도 관리할 수 있다.

4. 웹 에이전트 구현 및 고찰

3장에서 언급한 내용을 토대로 구현한 웹 에이전트 시스템은 사용자가 웹 서버를 통해 데이터베이스에 접근하여 데이터를 수정하거나 서로 다른 데이터베이스간의 데이터를 복제할 수 있도록 구현하였다.

4.1 데이터베이스 연결

3장에서 언급한 것과 같이, 본 논문에서 구현한 웹 에이전트 시스템은 사용자가 웹 서버에 접속하여 데이터베이스의 종류를 선택하면 해당 드라이버의 이름이 저장된 웹 페이지로 이동하여 데이터베이스의 종류에 상관없이 데이터를 관리할 수 있도록 하였다.

예를 들어, 데이터베이스의 종류를 오라클 데이터베이스로 설정하고, 연결하고자 하는 데이터베이스의 주소와 데이터베이스와 연결할 때 필요로 하는 사용자 아이디와 패스워드를 입력하는 화면은 Fig. 2와 같다.

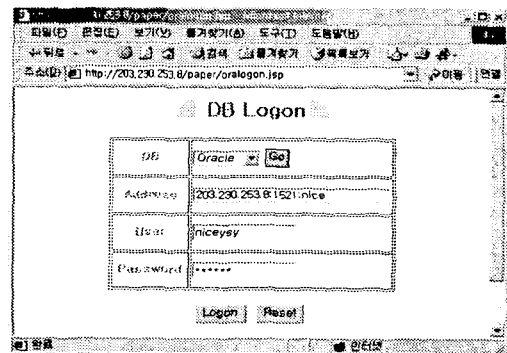


Fig. 2 Web interface of database logon

사용자가 입력하는 아이디와 패스워드는 연결하고자 하는 데이터베이스에 대한 접근 권한이 있어야 한다. Fig. 2와 같이 연결하고자 하는 데이터베이스의 주소는 203.230.253.8이고, 연결 포트번호는 1521, 데이터베이스 이름은 nice이다.

Fig. 2와 같이 데이터베이스에 대한 정보를 사용자가 입력하여 웹 서버로 전송하면, 웹 서버는 입력된 정보를 이용하여 데이터베이스에 연결할 수 있도록 질의 문장을 완성한다. 사용자가 선택한 데이터베이스의 종류에 따라서 바뀌는 JDBC

드라이버의 이름은 Class.forName 부분을 이용하여 원하는 드라이버를 로딩할 수 있다. 데이터베이스가 오라클일 때 드라이버 이름은 "oracle.jdbc.driver.OracleDriver"이고, SQL 서버를 사용할 경우는 "com.inet.tds.TdsDriver"이다. Fig. 2에서 선택한 데이터베이스가 오라클인 경우 소스는 Fig. 3과 같다.

```

Connection conn=null;
Statement stmt=null;
ResultSet rs=null;

try{
    Class.forName("oracle.jdbc.driver.OracleDriver");
    conn
    =java.sql.DriverManager.getConnection(url1+url2, user, pw);
    stmt=conn.createStatement();
    rs=stmt.executeQuery(list);
}catch(java.sql.SQLException e){
    out.println(e);
}
    
```

Fig. 3 Logon program

Fig. 3에서 "url1" 변수는 접속하고자 하는 데이터베이스 종류에 따라 주소영역에 붙은 접두어를 의미하므로 오라클 데이터베이스인 경우 접두어는 "jdbc:oracle:thin:@"가 되고 "list"라는 변수는 데이터베이스에 저장되어 있는 테이블의 목록을 요청하는 "select * from tab"이라는 질의문을 의미한다.

Fig. 2에서 사용자가 로그인 버튼을 선택하면 텍스트 박스에서 입력된 정보를 이용하여 웹 서버는 Fig. 3의 try문을 수행하여 웹 서버와 데이터베이스를 연결하고, 데이터베이스에 저장된 테이블 목록을 가져오는 질의문인 변수 "list"를 수행한 결과를 웹 서버에게 전송한다.

데이터베이스와 연결이 되면 웹 서버는 데이터베이스로부터 테이블 목록에 대한 정보를 얻게 된다. 웹 서버는 데이터베이스에 저장되어 있는 테이블 목록에 대한 정보를 테이블의 형태로 웹 브라우저를 통해 사용자에게 보여준다.

실제 오라클 데이터베이스와 연결하여 테이블 목록을 알아본 Fig. 4의 내용과 웹 서버가 보여주는 화면인 Fig. 5의 내용이 같다는 것을 확인하였다.

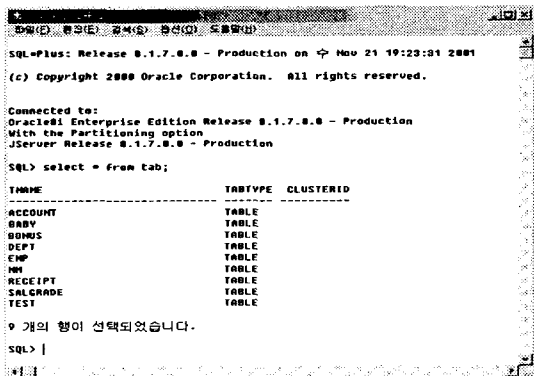


Fig. 4 Table list of oracle database

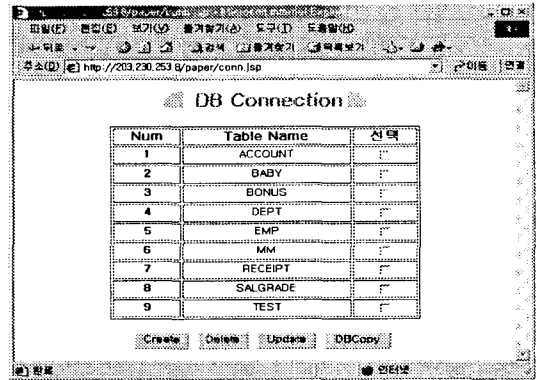


Fig. 5 Result window after DB connection

4.2 데이터베이스 관리

Fig. 5에서 보는 것과 같이 사용자는 테이블의 생성과 삭제 및 수정하는 데이터 관리 기능뿐만 아니라, 사용자가 선택한 특정 테이블을 다른 데이터베이스로 복제할 수 있다.

Fig. 5에서 사용자가 선택한 작업에 따라 웹 서버는 사용자에게 작업을 수행하기 위해 필요한 정보를 요구하고, 사용자는 필요한 추가 정보를 입력한다. 웹 서버는 입력된 정보를 이용하여 새로운 질의문을 완성하여 데이터베이스로 전송하고 데이터베이스로부터 전송된 질의 결과 내용을 브라우저를 통해 사용자에게 보여준다. 따라서 사용자는 데이터베이스의 데이터 관리를 위한 전문적인 지식이 없더라도 웹 서버가 요구한 추가 정보를 입력하면 웹 서버에서 해당 질의문을 생성하여 사용자가 원하는 작업을 할 수 있다.

Fig. 5에서 테이블을 생성하는 Create 버튼을 선택할 경우 사용자는 생성하고자 하는 테이블 이름, 컬럼 이름과 같은 추가 정보를 입력해야 하고 웹 서버는 입력된 정보를 이용하여 테이블을 생성하는 질의문을 데이터베이스로 전송하여 연결된 데이터베이스에 테이블을 추가한다. 테이블이 생성된 다음 웹 서버는 사용자로부터 추가하고자 하는 컬럼 수와 컬럼 내용을 입력받아 생성된 테이블에 데이터를 추가시킨다. Fig. 5에서 테이블을 삭제하는 Delete 버튼을 선택하면 웹 서버는 사용자가 선택한 테이블을 삭제하는 질의문을 완성하여 데이터베이스로 전송하여 데이터베이스에서 선택된 테이블을 삭제시킨다. Fig. 5에서 Update 버튼을 선택할 경우, 웹 서버는 데이터베이스에게 사용자가 선택한 테이블의 내용을 알아내기 위한 질의문을 완성하여 데이터베이스로 전송한다. 데이터베이스가 테이블에 저장되어 있는 컬럼과 레코드의 내용을 웹 서버에게 전달하면, 웹 서버는 이를 테이블의 형태로 바꾸어 사용자에게 보여준다.

예를 들어, SALGRADE라는 테이블을 체크하여 Update 버튼을 선택할 경우, 웹 서버는 Fig. 6과 같이 테이블에 저장되어 있는 데이터의 내용을 보여준다.

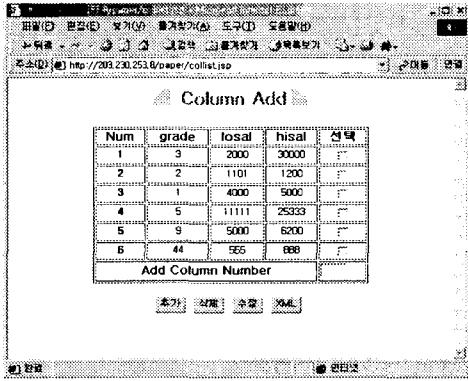


Fig. 6 Web interface for table update

Fig. 6에서 추가, 삭제, 수정 버튼의 기능은 Fig. 5에 있는 버튼들과 같은 비슷한 기능을 수행하지만, 관리하는 대상이 테이블 자체가 아닌 테이블에 저장되어 있는 컬럼과 레코드의 내용이라는 점에서 차이가 있다.

Fig. 5와 달리 Fig. 6에는 테이블의 내용을 XML 문서로 저장하는 XML 버튼이 있다. XML 문서의 루트 태그는 테이블 이름으로 하고, 루트 태그의 자식 엘리먼트로 <record> 태그를 두고 데이터베이스에 저장된 컬럼의 이름은 XML 문서에서 <record>의 자식 엘리먼트인 태그 이름으로 매핑시킨다.

본 논문에서 구현한 시스템은 루트 디렉토리에 “테이블이름.xml”이라는 이름으로 저장된다. 예를 들어, SALGRADE라는 테이블의 내용을 XML 문서로 저장하는 소스는 Fig. 7과 같고, 브라우저로 확인해 보면 Fig. 8과 같다.

```

try{
    PrintWriter pw
    = new PrintWriter(new FileOutputStream(nameOfTextfile));
    pw.println(title);
    pw.println(root);...
    for(int j=1; j<=num; j++){
        String colnmk = "colnm" + j;
        String colvalue = request.getParameter(colnmk);
        colname[j] = colvalue;
    }
    for(int i=1; i<=n; i++){
        String starttag = "<" + record + ">";
        pw.println(starttag);
        for(int k=1; k<=num; k++){
            String itemik = "item" + i + k;
            String item = request.getParameter(itemik);
            String tag = "<" + colname[k] + ">";
            .....
            String endtag = "</" + colname[k] + ">";
            pw.println(endtag);
        }
        String lasttag = "</" + record + ">";
        pw.println(lasttag);
    }
}
    
```

Fig. 7 XML save program

Fig. 7에서 FileOutputStream(nameOfTextfile)은 웹 브라우저가 보여주는 파일 내용을 텍스트문서로 저장하는 기능이고, “pw”는 Fig. 6에서 보여주는 테이블의 내용을 XML 문서로 저장하는 역할을 한다. 변수 “title”은 XML 문서의 선언 부분인 “<?xml version='1.0' encoding='euc-kr'?>”을 표시하고 변수 “root”는 테이블의 이름을 나타내는 것이다.

Fig. 7에서 두 번째 for문의 내용은 Fig. 6에서 보여주는 각각의 컬럼인 grade, losal, hisal의 내용을 XML 문서의 <record> 엘리먼트의 자식 엘리먼트로 저장한다.

Fig. 7과 같은 소스에 의해서 Fig. 6의 내용을 저장한 XML 문서의 내용은 Fig. 8과 같다.

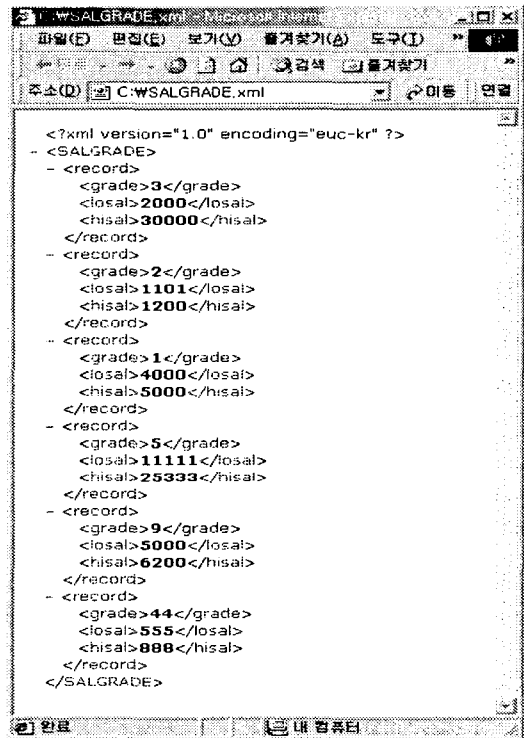


Fig. 8 SALGRADE.xml document

4.3 데이터베이스 복제

서로 다른 데이터베이스간의 데이터 복제를 위해 사용되는 DBCopy 버튼을 선택하면 복제하고자 하는 데이터를 저장할 데이터베이스에 대한 주소와 아이디 및 패스워드를 사용자가 입력하는 화면이 나타난다. 사용자가 추가 정보를 입력하면 웹 서버는 입력된 정보를 토대로 데이터베이스에 연결하여 테이블을 생성하여 복제하고자 하는 데이터 내용을 추가한다.

DBCopy 버튼을 선택하면 데이터베이스에 연결하는 화면이 나타난다. Fig. 2와 같은 화면이 나타나지만 복제할 데이터가 있는 소스 데이터베이스가 아니라 복제해서 저장할 목적지 데이터베이스에 대한 정보를 입력한다는 것이 다르다.

데이터를 복제하고자 할 경우, 복제할 데이터가 있는 데이터베이스의 테이블 목록을 보여주는 웹 브라우저 화면에서 하나의 테이블을 정하여 DBCopy라는 버튼을 선택한다.

웹 서버는 사용자가 체크한 테이블로부터는 얻은 테이블 이름을 이용하여 테이블의 컬럼 이름과 컬럼의 데이터 타입을 알아내는 질의문을 완성하여 데이터베이스로부터 저장되어 있는 테이블의 정보를 알아낸다.

소스 데이터베이스가 오라클인 경우 테이블의 정보를 기억하고 있는 user_tab_column를 이용하는 소스는 다음 Fig. 9와 같다.

```
String colup
= "select column_name, data_type from user_tab_columns
  where table_name = SALGRADE"
try{
  Class.forName("oracle.jdbc.driver.OracleDriver");
  conn6=java.sql.DriverManager.getConnection(url, user, pw);
  stmt6=conn6.createStatement();
  stmt6.executeUpdate(colup);
}
catch(java.sql.SQLException e){
  out.println(e);
}
```

Fig. 9 Table information query program

본 논문에서 구현된 웹 에이전트 시스템의 주된 기능인 이기종 데이터베이스간의 데이터 복제를 위해 XML을 이용한다. 사용자의 입장에서 복제하고자 하는 데이터베이스의 데이터를 직접 목적지 데이터베이스로 데이터를 보내는 것처럼 보이지만 실제로는 복제할 데이터의 내용을 XML 문서로 변환하고 변환된 XML의 내용을 목적지 데이터베이스에 저장하는 것이다.

웹에 존재하고 있는 XML 문서 중에서 사용자는 데이터베이스가 저장하고자 하는 특정 XML 문서를 저장하기 위해서 특정 XML 문서에 대한 위치 정보, 데이터베이스에 저장할 테이블의 이름이 될 루트 태그 이름, 엘리먼트의 이름을 입력한다. 이 같은 수작업은 데이터베이스에 저장하고자 하는 엘리먼트의 수가 많을수록 번거로운 작업이 되지만, 필요한 정보만 입력하여 XML 문서 전체의 저장이 아니라 필요한 부분만 저장할 수 있다는 장점을 가지게 된다.

데이터베이스의 데이터 관리와 유사하게 XML 문서의 내용을 수정 버튼을 이용하여 추가와 삭제, 수정할 수 있다. 그리고, 수정 과정을 거치지 않고 XML 문서의 내용을 원하는 데이터베이스에 저장할 수 있다. 이는 웹에 존재하는 XML 형식의 웹 페이지를 데이터베이스에 저장하여 관리함으로써 정보의 재사용과 정보 공유를 더욱 원활하게 한다.

위에서 언급한 것처럼 본 논문에서 구현한 웹 에이전트 시스템은 별도의 소프트웨어 없이 서로 다른 데이터베이스간 데이터 복제를 할 수 있고 웹에서 정보검색 능력만 가진 초보자라 할지라도 데이터베이스와 정보교환을 쉽게 할 수 있다.

정부기관과 항만과 물류유통에 관련된 업체간의 정보를 교환하기 위해서 사용하고 있는 EDI 문서는 각기 다른 형태의 EDI 문서를 사용하므로 EDI 문서 양식을 변환하는 소프트웨어를 이용한다. 이는 정보 교환의 대상이 바뀔 때마다 새로운 소프트웨어를 사용해야 한다는 점에서 비효율적이다. 본 논문에서 구현한 웹 에이전트 시스템은 EDI 문서 변환 소프트웨어를 사용하지 않고 XML을 이용하여 EDI 문서의 내용을 웹 브라우저를 통해 볼 수 있는 형태로 보여줌으로써 EDI 문서에 대한 지식이 없는 초보자라 할지라도 쉽게 정보교환을 할 수 있다. 뿐만 아니라 웹을 통한 정보 교환이므로 실시간으로 화물에 대한 정보를 수집하여 화물에 관한 정보를 통합하여 관리할 수 있어 물류비용을 절감할 수 있다.

원활한 물류의 흐름을 위해서 국가, 업체, 운송수단에 따라 적용되는 요금과 화물정보, 컨테이너 정보, 위험물 정보 등을 교환해야 한다. 국내뿐만 아니라 해외의 항만과 물류유통에 관련된 업체들은 각기 다른 데이터베이스를 사용하여 물류에 대한 정보를 관리하고 있다. 서로 다른 데이터베이스에 저장된 데이터를 교환하기 위해서는 동일한 데이터베이스로 교체하거나 데이터베이스간의 데이터를 교환하는 별도의 소프트웨어를 사용해야 하므로 이는 비용 면에서 비효율적이다. 본 논문에서 구현한 웹 에이전트 시스템은 데이터베이스를 교체하거나 소프트웨어 없이 XML을 이용하여 데이터베이스로부터 정보를 추출하여 웹 브라우저를 통해 내용을 보여주거나 데이터베이스에 저장된 데이터를 복제하여 다른 데이터베이스에 저장할 수 있어 신속하게 정보교환을 할 수 있다.

5. 결 론

항만과 관련된 기관이나 업체들이 물류 기능을 수행하기 위해 다양한 데이터베이스를 독자적으로 보유, 운영하기 때문에 정보 공유와 교환이 어렵다. 이를 해결하기 위해 본 논문에서 구현한 시스템은 수출입 화물에 관련된 정보뿐만 아니라 실시간으로 선박업무를 지원, 선박과 본사간의 동일한 데이터베이스에 의한 항해분석, 선박 컴퓨터 자원의 효율적 이용, 다수의 사용자가 동일한 인터페이스를 통한 신속한 업무처리를 통해 항만과 해운에 관련된 부가적인 정보 제공과 나아가서는 의사결정에 필요한 정보, 그리고 학술적인 정보 등을 제공한다. 인터넷 환경으로 본사와 선박을 연결하여 기업 내 자원-관리를 구현하여 투자비용 절감과 다수의 사용자가 자원을 공유한다.

서로 다른 데이터베이스에 저장된 정보를 공유하고 관리하기 위해 문서에 대한 구조 정보를 내포하고 있는 XML을 정보 저장소로 이용하여 정보 공유와 정보 검색 및 정보 관리를 쉽게 할 수 있다. 구현된 웹 에이전트 시스템은 사용자에게 인터넷을 통하여 웹 서버와 연결되고 웹 브라우저를 통해 데이터베이스의 데이터와 XML 문서의 내용을 보여준다. 이는 현재 EDI 시스템이 도스환경을 기반으로 하고 있어서 윈도우

환경과 인터넷 환경에 익숙한 사용자들에게 주는 사용상의 불편을 해소시켜 주고, 사용자는 데이터베이스를 관리하기 위한 사전지식 없이 데이터 관리 작업을 언제 어디서든지 할 수 있다.

향후 연구 방향으로는 데이터베이스에 접근하고 데이터를 관리하는 과정이 웹에서 이루어지므로 디지털 서명을 통한 보안 문제와 XML 질의언어인 XQL을 이용하여 정보 저장소에 서 특정 tag의 내용만을 추출 또는 통합하여 정보 검색이 가능하도록 하는 기능 부여에 관한 연구가 요망된다.

참 고 문 헌

- [1] 김종철, (1996) “종합물류정보전산망의 효율적인 구축방안”, 한국해운학회지 제22호, pp. 169-194
- [2] 노홍승, 이철영, (1996) “항만물류 서비스의 개념과 특성 고찰에 관한 연구”, 한국해운학회지 제23호, pp. 101-124
- [3] 박철현외 정재현, 심대익, 이상구, (1999) “구조화된 문서에 대한 DBMS와 IRS의 성능 비교”, '99 한국 데이터베이스 학술대회 논문집 15권 1호, pp. 218-225
- [4] 이강찬, 손홍, 박기식, (2001) “XML 표준화 동향”, 정보과학회지 제19권, 제1호, pp. 6-14
- [5] 장우혁, 김홍식, (2001) “XML기반의 효율적인 데이터 저장관리를 위한 DB2XML 변환 Wrapper의 설계”, 한국정보과학회 학술발표논문집, Vol. 28, No. 1, pp. 106-108
- [6] 장홍훈, (1995) “우리 나라 해운 항만 물류의 EDI 구축과 활성화 방안에 관한 연구”, 한국해운학회지 제21호, pp. 459-479
- [7] W3C, (2001) “Extensible Markup Language(XML) 1.0” <http://www.w3.org/TR/2000/REC-xml-20001006>

원고접수일 : 2002년 8월 28일

원고채택일 : 2002년 10월 15일