

# 대용량 웹 데이터베이스에서의 정보 추출 및 저장을 위한 XML 래퍼 구현

박성진\*

## ◆ 목 차 ◆

- |                  |                  |
|------------------|------------------|
| 1. 서 론           | 4. XML-RDB 래퍼 구현 |
| 2. 관련 연구         | 5. 적용 예          |
| 3. XML-RDB 래퍼 설계 | 6. 결 론           |

## 1. 서 론

최근에 인터넷 사용이 급증하며 웹이 보편화된 정보 전달 및 정보 제공의 수단으로 인식됨에 따라 웹 상에서 제공되는 정보의 양도 폭발적으로 증가하고 있으며, 이로 인하여 웹은 단순한 정보 교환의 기능을 뛰어넘어 정보 저장소의 역할까지 하고 있다. 즉, 지구상에 존재하는 가장 큰 데이터베이스로서 그 가치와 중요성이 새롭게 인식되고 있으며 이러한 웹 상의 정보들을 하나의 대용량 웹 데이터베이스로서 간주하고 이를 효율적으로 활용하기 위한 다양한 연구들이 진행되고 있다.

그러나, 현재 웹 데이터베이스를 구성하는 다양한 웹 페이지들의 대부분이 간편하고 사용하기 쉽다는 장점을 갖는 HTML 문서로 제공되고 있어 정보 축적 및 정보 관리의 측면에서 한계가 있고 이를 보완하기 위한 방법중의 하나가 구조적이고 기능적인 언어로 부상하고 있는 XML(eXtensible Markup Language)을 기반으로 하여 문서를 제작하거나 변환하는 것이다.

XML[1]은 웹 상에서의 데이터 교환을 위해 제안된 표준 언어로 XML 문서의 구조적 유동성은 모든 형식의 데이터가 XML 형태로 기술될 수 있도록 해주며, 웹에서 운용되는 모든 데이터가 통합, 저장, 처리될 수 있는 기반을 제공해 준다. 따라서, 점차 기존의 웹 문서 표준인 HTML을 대체함으로써 향후 웹 데이터

베이스에서의 XML 문서의 양은 급속히 많아질 것이며 이러한 대량의 XML 문서를 효율적으로 저장 관리하고, 원하는 XML 문서를 검색할 수 있는 기능 또한 매우 중요해질 것이다.

본 논문에서는 웹 데이터의 통합적이고 효율적인 관리를 위하여 XML 문서를 직접 저장하거나 혹은 HTML로 작성된 기존의 대용량 웹 문서들을 XML 문서로 변환후 저장하고 검색할 수 있는 XML-RDB 래퍼 시스템을 설계하고 구현하였다. 개발된 XML-RDB 래퍼 시스템을 사용함으로써 HTML 문서로 구성된 대용량 웹 데이터베이스로부터 필요한 정보만을 추출하여 XML 문서로 변환한 뒤 정보저장소에 저장하고 검색 결과를 다양한 인터페이스를 통해 제공하는 것이 가능하다.

## 2. 관련 연구

XML 문서를 저장하고 검색하는 시스템에 관한 연구가 활발하게 이루어지고 있다. 먼저, XML 전용 데이터베이스 혹은 XML 확장 데이터베이스에 기반한 저장 및 검색 기능을 제공하는 시스템의 경우, 기존 데이터베이스를 그대로 활용하기 위해서는 데이터 변환 등의 많은 작업이 행해져야 하는 문제점이 있다.

한편, 기존 데이터베이스 시스템을 이용하여 XML 데이터를 저장하고 질의를 수행하는 방법[3,5,6]은 XML 문서를 저장하기 위한 적절한 테이블이나 클래스를

\* 한신대학교 정보시스템공학과 조교수

생성한 다음 미들웨어나 래퍼(wrapper)를 이용하여 XML에 대한 질의를 데이터베이스에 대한 질의어로 변환하여 원하는 결과를 얻는 방법으로 새로운 시스템을 구현하지 않고 원하는 결과를 생성할 수 있는 장점이 있다. 이 경우, 관계형 데이터베이스 기반[7], 객체-관계형 데이터베이스 기반[8,9], 객체형 데이터베이스 기반[10]의 3가지 방식이 존재한다.

한편, 저장 방식으로는 XML 문서를 파싱 단계를 거치지 않고 BLOB이나 파일에 전체를 하나의 큰 객체로 저장하는 방법과 XML 데이터를 파싱하여 적절한 크기의 형태로 가공한 후 분할 저장하는 방법이 있다[11].

## 2.1 XML-가능 데이터베이스(XML-Enabled Databases)

XML-가능 데이터베이스는 XML 문서와의 데이터 전송을 위한 확장 모듈을 가진 데이터베이스를 말한다. XML-가능 데이터베이스들은 대부분 데이터 중심의 문서(data-centric document)를 저장하도록 설계되어 있다. 이것은 XML 문서를 저장할 수 있도록 특별히 설계된 테이블이 아닌 일반 사용자가 만든 테이블에 저장해야 하기 때문이다. 또한, 문서 중심의 문서(document-centric document)인 경우에는 한 필드에 저장하는 경우도 있다. 이러한 구조를 가진 데이터베이스로는 DB2, Informix, Oracle8i, Microsoft SQL-Server 등이 존재한다.

## 2.2 XML 미들웨어(middleware)

XML 미들웨어는 XML 문서와 데이터베이스간에 데이터를 전송하기 위해 사용하는 프로그램이다. 이것은 매우 여러 종류의 언어로 작성될 수 있으나, 대부분은 ODBC, JDBC 또는 OLE DB를 사용한다. XML 미들웨어 구조를 가지는 XML 저장 시스템에는 다음의 것들이 존재한다.

- XML-DBMS(15)

XML-DBMS는 XML 문서의 관계형 데이터베이스 사이에 데이터를 전송하기 위한 자바 패키지(package)

이다. 이 패키지는 사용자가 작성하는 응용 프로그램에서 사용될 수도 있다. XML 문서 구조와 관계형 데이터베이스와의 매핑(mapping)을 기술하기 위해서 XML-DBMS 매핑 언어를 사용한다. 이 언어는 XML 기반의 언어로서 XML 문서의 태그와 테이블의 필드의 관계를 기술한다. XML-DBMS는 이 매핑 언어를 XML 문서를 데이터베이스에 저장하는데 사용하기도 하고, 데이터베이스의 데이터를 추출하여 XML 문서를 구성하는데 사용하기도 한다.

- DB2XML(16)

DB2XML은 관계형 데이터베이스로부터 XML 문서로 데이터를 전송하기 위한 자바 클래스이다. XML-DBMS는 데이터베이스와 XML 문서 사이에 데이터 전송이 양방향 모두 가능하지만, DB2XML은 데이터베이스에서 XML 문서로의 한 방향만 가능하다. SQL 문장을 입력받아서 XML 문서를 출력하는데, 옵션에서 태그 이름을 조정할 수 있다.

본 논문에서는 DTD 종속적인 스키마 구조와 엘리먼트 분할 방식에 기반한 RDB-XML 래퍼를 설계 구현함으로써 기존 관계형 데이터베이스를 활용하여 웹 상에서 추출한 정보를 XML 문서로 효율적으로 저장하고 검색할 수 있도록 하였다. 특히, XML 문서를 저장하기 위해서 특별히 설계된 스키마 구조를 가지고 있어서 XML 문서의 내용뿐만 아니라 문서의 구조 정보 등도 같이 저장할 수 있으며 이를 바탕으로 XML 문서의 내용 검색이나 구조 검색 등을 효율적으로 할 수 있다.

## 3. XML-RDB 래퍼 설계

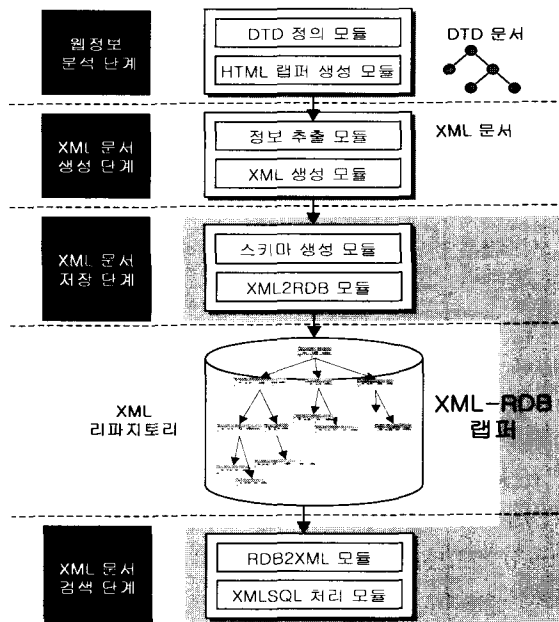
대용량 웹 데이터베이스를 구성하는 웹 페이지에는 많은 유용한 정보들이 포함되어 있지만 다양한 사이트들로부터 필요로 하는 정보들만을 통합하여 통일된 형식으로 제공하기 위해서는 웹 페이지 내에서도 필요한 정보만을 추출하고 XML로 변환, 저장 관리하는 XML 래퍼 시스템이 필요하다.

본 논문에서 설계 및 구현한 XML 래퍼 시스템은

웹 상에서 추출한 정보로 생성된 XML 문서를 어떠한 관계형 데이터베이스에도 분할 저장가능하고 정보 검색 결과를 XML 문서 형태로 변화할 수 있다 즉 저장된 XML 문서 내용을 다시 기존의 XML 파일로 복원할 수 있어 XML 파일과 데이터베이스간에 상호 변환이 가능하다.

### 3.1 XML 랩퍼 시스템의 전체 구성

HTML로 작성된 기존의 웹 문서들은 대용량 웹 데이터의 통합적이고 효율적인 관리를 위하여 XML 문서로의 변화 저장 및 검색이 필요하다 이러한 기능을 제공하는 XML 랩퍼 시스템은 그림 1과 같이 크게 4단계로 구성된다. 먼저, 웹 문서에 대한 표준 DTD를 정의하고 HTML 랩퍼를 생성하는 웹 정보 분석 단계, 필요한 정보만을 웹 문서로부터 추출(extraction)하고 문서로 생성(generation)하는 XML 문서 생성 단계, 추출 변환된 문서를 데이터베이스 안에 분할 저장하는 XML 문서 저장 단계 및 저장된 문서에 대한 자유로운 검색과 검색결과를 XML 문서로 재 생성하는 XML 문서 검색 단계로 구성된다.



(그림 1) XML 랩퍼 시스템 구성

## 3.2 웹 정보 분석 단계

### 3.2.1 DTD 정의 모듈

먼저, 웹 정보 분석의 시작 단계인 DTD 정의 모듈에서는 특정 분야와 관련된 HTML 문서들에 대한 구조적 분석을 통해 꼭 필요한 정보들을 선별한 뒤 효율적으로 구조화함으로써 분야별 표준 DTD를 정의한다.

웹 상에서 검색되는 HTML 문서는 동일한 분야의 경우, 일반적으로 대부분 비슷한 양식에 유사한 내용을 가지고 있다. 따라서, 웹 문서 포맷에 대한 적절한 구조 분석을 통해 분야별로 웹 문서로부터 추출해야 할 필수적인 데이터 내역을 미리 정의하고 적절히 구조화함으로써 통합 관리하고자 하는 웹 정보 분야별로 통일된 DTD를 명세한다. 정의된 DTD의 구조 정보는 정보 추출, 문서 생성 및 저장과정으로 전달되어지고, 일련의 과정을 통해 XML 문서의 저장구조로 변환된다.

### 3.2.2 HTML 랩퍼 생성 모듈

HTML 랩퍼 생성 모듈은 앞 단계에서 정의된 분야별 표준 DTD 안의 엘리먼트에 대응하는 정보를 HTML 문서로부터 추출하는 랩퍼를 각각의 웹 페이지별로 수동으로 구축한다. 파싱을 통해 각 HTML 문서의 구조를 분석하고 위치 정보, 글자 형식 및 포맷 등의 문서 서식에 따라 적절히 XML 문서로 변환하는 랩퍼를 생성할 수 있다.

특정 분야별로 추출하고자 하는 정보를 담고 있는 웹 상의 HTML 문서들은 대부분 구조적 특성을 갖고 있기 때문에 랩퍼 생성은 그다지 어렵지 않으며 문서의 계층 구조를 적절히 분석함으로써 정보 추출 랩퍼를 개발할 수 있다.

## 3.3 XML 문서 생성 단계

### 3.3.1 정보 추출 모듈

정보 추출 모듈은 랩퍼를 통해 HTML 문서로부터 필요로 하는 XML 문서 정보만을 뽑아낸다. 이전 단계에서 생성된 랩퍼는 대응되는 웹 페이지의 구조가 변화하지 않는 한 동적으로 변화하는 HTML 문서의 내용으로부터 끊임없이 새로운 XML 엘리먼트 속성값과 텍스트 값들을 제공한다.

래퍼를 통해 추출된 이러한 각 문서 유형별 필수적인 정보들은 이후에 적절한 XML 문서로 변환된다.

### 3.3.2 XML 생성 모듈

XML 생성 모듈에서는 추출된 데이터를 가공해서 사전에 정의된 DTD를 충족하는 XML 문서로의 매핑(mapping) 과정이 이루어진다.

XML 생성 모듈은 미리 정해진 규칙에 의해 HTML 래퍼가 자동으로 추출한 데이터 항목들이 XML 생성 과정으로 전달하고 이 과정에서 DTD에서 정의한 문서 구조와 전달된 데이터 항목들을 대응시켜 적절한 XML 문서를 생성하게 된다.

## 3.4 XML 문서 저장 단계

### 3.4.1 스키마 생성 모듈

스키마 생성 모듈에서는 DTD의 구조 정보를 사전에 분석하고 그에 따른 XML 문서의 효율적인 저장을 위하여 관계형 데이터베이스 안에 DTD 의존적인 저장 스키마를 생성한다. 생성한 스키마 구조는 XML 문서가 갖는 계층적인 구조적 특성을 관계형 데이터 모델의 외래키(foreign key)를 사용하여 엘리먼트간의 상하(직계) 관계를 일대 다의 참조 관계로 변환하여 표현한다.

특히, XML 문서의 수정이 가능하도록 문서를 엘리먼트 단위로 쪼개어 저장하는 ‘분할 저장’ 방식을 사용하였으며 DTD 정의 시에 엘리먼트는 구조 정보를, 애트리뷰트는 데이터를 표현하기 위한 목적으로 명확히 분리함으로써 엘리먼트를 관계형 데이터베이스의 테이블에 대응시키고, 애트리뷰트는 엘리먼트 테이블의 컬럼으로 매핑시켜 문서 저장 및 검색 과정의 정형화가 가능하도록 하였다.

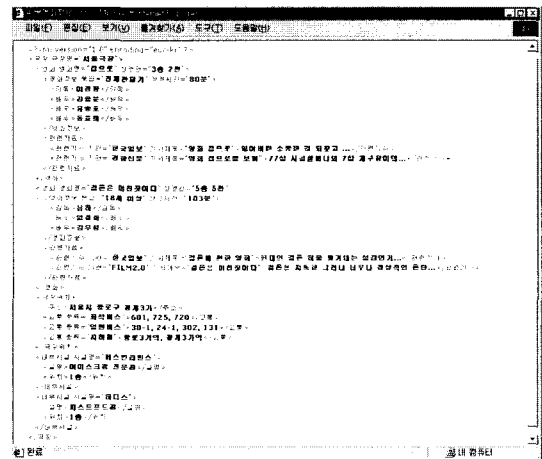
이를 위해서는 DTD 정의시 엘리먼트와 애트리뷰트의 사용에 대한 엄격한 제한이 필요하다. 즉, XML 문서 내에서 오직 하나의 값이 필요한 경우에는 애트리뷰트가 가장 적절하며, 반면에 여러 개의 값이 필요한 경우라면 엘리먼트를 사용하도록 하는 것이다.

### 3.4.2 XML2RDB 모듈

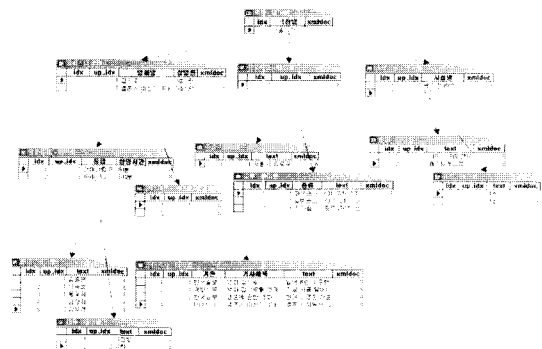
XML2RDB 변환 모듈은 W3C에서 제안된 DOM 파

서를 이용하여 구현된다. XML2RDB 변환기는 DOM 파서를 사용하여 XML 문서 파일을 해석하고 적절한 테이블에 매핑하여 XML 엘리먼트 값들을 저장하게 된다. 이때, XML 문서를 저장하는 리파지토리는 관계형 데이터베이스를 사용하며 이는 가장 범용적으로 널리 사용되는 데이터베이스이기 때문이다.

XML 문서를 받아서 스키마 구조에 맞게 데이터베이스에 저장하는 작업을 한다. 예로 그림 2와 같은 XML 문서의 경우에는 그림 3과 같은 스키마 구조로 각 엘리먼트 테이블에 분할되어 저장된다. 이를 위해서는 저장되어질 스키마 구조가 데이터베이스 안에 미리 생성되어 있어야 한다. XML 문서의 파일명, DTD 정보, 문서내용에 대한 데이터 그리고 입력된 시간에 대한 메타 정보 등이 포함되어 저장된다.



(그림 2) XML 문서 예



(그림 3) 저장 후 테이블 내용 및 구조

```
Show {출력할 엘리먼트명}
On {검색할 해당조건}
InXML '{검색할 XML 문서명}'
InDTD '{검색할 DTD명}'
```

(그림 4) 검색 구문 형식

```
Show 극장.*(영화.영화명(영화정보.*),극장위치.*(주소.*,교
통.text))
On 영화.영화명='집으로'
InDTD 'movie.dtd'
```

(그림 5) 검색 구문 예

```
<?xml version="1.0" encoding="euc-kr" ?>
- <극장 극장명="서울극장">
- <영화 영화명="집으로">
  <영화정보 등급="전체관람가" 상영시간="80
  분" />
</영화>
- <극장위치>
  <주소>서울시 종로구 청계3가</주소>
  <교통>601, 725, 720</교통>
  <교통>30-1, 24-1, 302, 131</교통>
  <교통>종로3가역, 청계3가역</교통>
</극장위치>
</극장>
```

(그림 6) 검색 구문 실행 결과

### 3.5 XML 문서 검색 단계

#### 3.5.1 RDB2XML 모듈

RDB2XML 변환 모듈은 데이터베이스에 저장된 XML 문서를 다시 복원하는 기능을 한다. 이는 XML 문서를 파싱하는 과정에서 생성된 문서에 대한 메타 정보를 사용하여 XML 문서의 엘리먼트 이름과 속성 값들을 다시 복원하는 과정을 거치게 된다. 이때, 저장했던 원래의 XML 문서를 복원할 수 있을 뿐만 아니라 적절한 질의를 실행함으로써 필요로 하는 부분 정보만을 별도의 XML 문서로 생성해 내는 것도 가능하다.

#### 3.5.2 XMLQL 처리 모듈

XMLQL(XML Query Language) 처리 모듈은 분할되어 데이터베이스에 들어 있는 XML 문서 정보들에 대해 본 논문에서 새롭게 제시한 그림 4와 같은 show절로 시작하는 XMLQL 언어를 처리한다. 이 모듈에서는 show절에 의해 명시된 검색 요청을 처리하고 그 결과를 원하는 형태의 XML문서로 만들어 주는 기능을 한다.

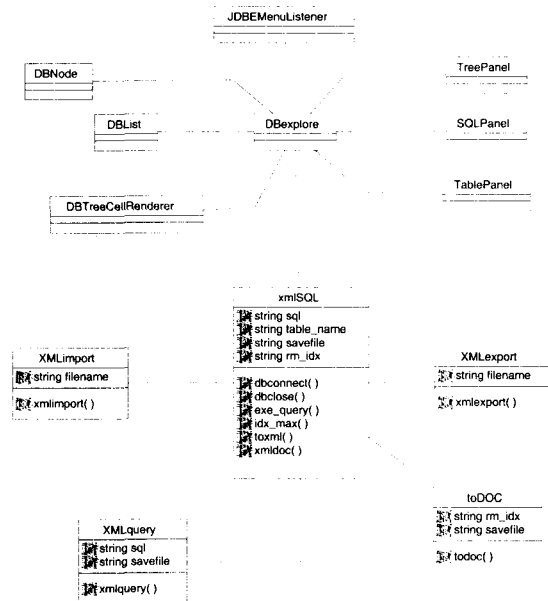
넘겨주는 변수로는 조건과 파일명이 있는데, 그림 5와 같은 질의문을 넘겨주어 실행하면 내부 변환을 통하여 그림 6과 같은 XML 문서가 생성된다. 이때 생성된 문서의 파일명은 넘겨받은 파일명으로 지정된다.

검색 결과의 XML 문서로의 변환을 위해서는 show 검색 구문이 내부적으로는 여러 엘리먼트 테이블들간의 left outer join 연산을 수행하는 SQL 구문들로 변환되어진다. 변환된 내부 SQL 구문은 여러 개의 레코드셋(record set)들을 조합하여 DOM(Document Object

Model) 구조의 루트 노드부터 하위 노드까지 적절한 데이터들을 매핑함으로써 XML 문서를 생성한다.

## 4. XML-RDB 랩퍼 구현

개발된 XML-RDB 랩퍼 시스템은 크게 클라이언트 패키지와 서버 엔진(oraxml) 패키지로 구성되어 있으며 해당 패키지는 다음 그림 7과 같은 클래스들을 중심으로 구성되었다.



(그림 7) 클래스 구성

#### 4.1 클라이언트 패키지

클라이언트 패키지를 구성하는 클래스 모듈들은 DBExplorer, JDBEMenuListener, DBList, DBNode, DBTreeCellRenderer, SQLPanel, TablePanel, TreePanel 들이며 이들 클래스들을 통해 관리자가 데이터베이스내의 XML 문서를 관리하고 각 문서에 대한 메타 정보를 확인하거나 설정할 수 있다. 기본적인 검색 기능과 리파지토리 엔진 모듈의 서비스 확인을 위한 인터페이스를 제공한다.

#### 4.2 서버 패키지

서버 패키지를 구성하는 클래스 모듈은 xmlSQL, XMLImport, XMLQuery, XMLExport, toDOC 등이다. 이들 클래스들은 oraxml이라는 서버 패키지를 구성하여 클라이언트로부터 XML 문서의 저장 및 검색 요청을 처리한 뒤 그 결과를 XML 문서로 생성하여 제공한다.

##### (1) xmlSql 클래스

데이터베이스에 관련된 대부분 작업을 처리하여 준다. 기본적으로 데이터베이스에 연결하고 질의문을 실행하며, 데이터베이스에 연결 해제하는 작업을 한다. 필요에 따라 테이블 인덱스의 최대값을 구하거나 XML 문서의 내용을 반환하는 기능이 있다. xmlSql의 경우 XMLImport나 XMLExport에서 사용된다. 해당 데이터베이스에 대해 기본적인 연결, 질의(query) 실행, 해제와 XML 문서에 대한 변환에 관한 핵심 클래스이다.

##### (2) XMLImport 클래스

XML문서를 받아서 구조에 맞게 데이터베이스에 저장하는 작업을 한다. xmlSql을 호출하여 데이터베이스에 연결, 입력 및 연결해제 등의 작업을 수행하며 저장되어질 스키마 구조는 데이터베이스에 미리 생성되어져 있어야 한다. XML문서의 파일 이름, DTD 정보, 문서내용에 대한 데이터 그리고 입력된 시간에 대한 메타 정보를 포함하여 저장되어진다.

##### (3) XMLExport 클래스

매개변수로 show 구문과 저장할 XML 문서명을 넘겨주면, show절에 의해 생성된 결과를 XML 문서명으로

로 저장하여 준다. 클래스내에서 xmlSql 클래스를 사용하여 데이터베이스 연결 후 XML 문서를 검색 생성하게된다.

##### (4) XMLQuery 클래스

데이터베이스에 들어 있는 정보에 대해 조건에 맞는 정보를 추출하여 XML문서로 만들어 주는 기능을 한다.

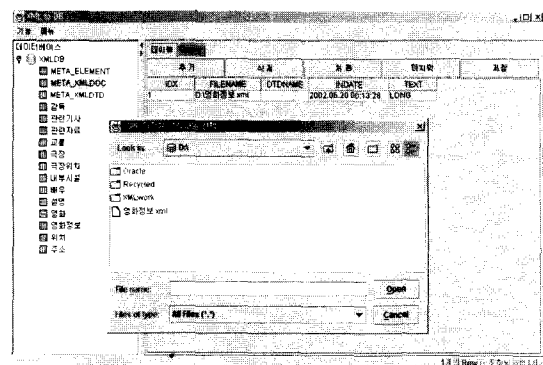
##### (5) 추가 클래스 구조

Java Sun사에서 제공하는 파서인 JAXP의 클래스와 Java에서 데이터베이스와 연결을 위한 Oracle JDBC 클래스 등이 필요하다.

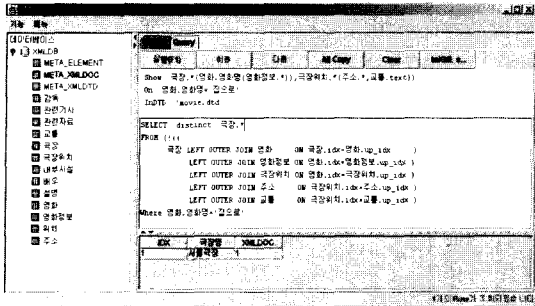
구현 환경은 윈도우즈 환경에서 자바 언어를 사용하여 개발되었으며 JAXP XML 파서와 JDBC 미들웨어 그리고, 오라클 DBMS를 사용하여 개발되었다.

구현한 시스템에서는 이러한 파서의 DOM 인터페이스를 적절히 사용함으로써 주어진 XML 문서를 엘리먼트 단위로 분할하여 메타 정보와 함께 저장소에 저장하는 임포트 기능을 구현하였다. 그림 8은 engine 클래스에 있는 XMLImport를 실행하는 화면으로 XML2RDB 모듈을 구현한 기능을 보여준다.

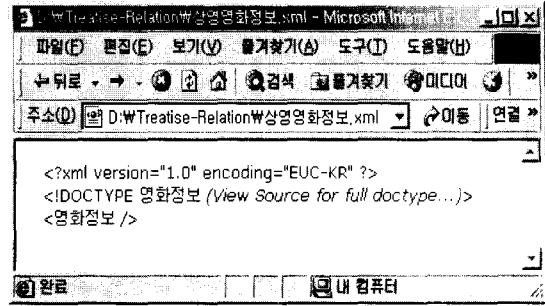
show 검색 구문을 사용하여 해당 결과를 XML 문서로 받아보기 위해서는 그림 9에서 보는 바와같이 입력된 show 검색 구문이 적절한 SQL 형태로 변환되어야하며 export 기능을 선택하여 저장될 XML 문서명을 지정하면, 이를 바탕으로 XML 문서를 생성하게



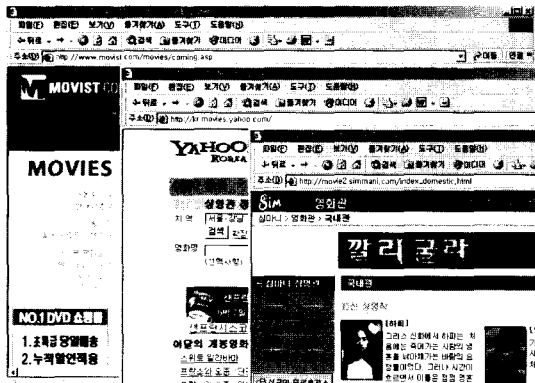
(그림 8) XML2RDB 실행 화면



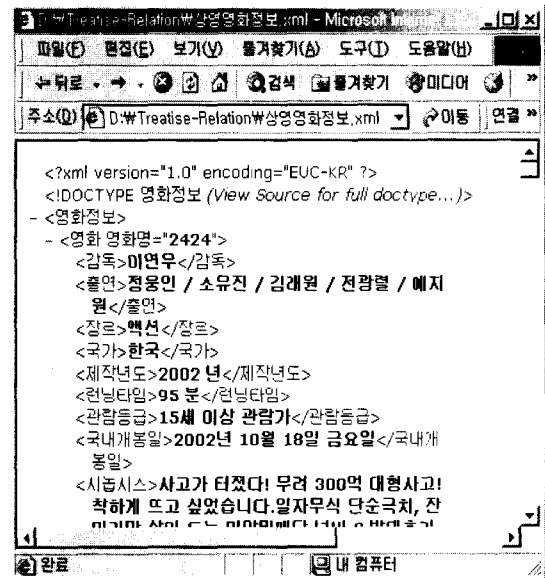
(그림 9) RDB2XML 실행 화면



(그림 11) 추출전 XML 문서



(그림 10) HTML 웹 문서의 예



(그림 12) 추출후 XML 문서

된다. 클라이언트 어플리케이션에서는 외부 조인으로 변환된 SQL이 나타났지만 실제 engine에서는 조인 구문을 나타내지 않고, XMLexport 클래스에서 모두 실행되어 XML 문서로 생성하게 된다.

### 5. 적용 예

개발된 XML-RDB 랩퍼 시스템을 웹 상에서 영화에 대한 포털 사이트를 구축하는 과정에 적용하였다.

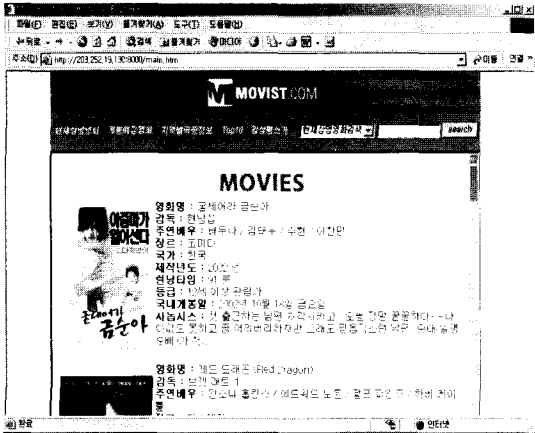
그림 10과 같은 다양한 영화 정보제공 사이트의 웹 문서들로부터 사전에 정의된 DTD에 따른 필요한 정보들을 추출하게 된다.

그림 11은 웹사이트로부터 정보를 추출하기 전의 XML 문서의 내용을, 그림 12는 정보 추출후 XML 문서의 내용을 보여준다.

변환된 XML 문서는 다시 엘리먼트 단위로 분할하여 앞에서 설명한 그림 8과 같은 임포트 기능을 통해

데이터베이스에 저장된다. 저장된 XML 문서는 앞의 그림 9와 같이 show절을 적절히 작성함으로써 원하는 내용만을 검색하여 XML 문서로 생성하는 것이 가능하며 이를 활용하여 다음 그림 13과 같은 영화정보 포털 사이트를 구축할 수 있다.

그림 13의 포털 사이트는 다양한 웹 상의 영화 정보 사이트의 웹 문서들로부터 매시각 동적으로 변화하는 최신의 정보들을 추출, 저장하였다가 통합된 형식으로 정보를 제공할 수 있다. 데이터베이스에 저장된 영화 정보들은 WML로의 변환도 용이하기 때문에 무선 단말기를 통해서도 관련 정보를 실시간으로 제공할 수 있는 이점을 갖는다.



(그림 13) 영화정보 포탈 사이트 예

## 6. 결론

본 논문에서는 웹에서 운용되는 모든 데이터가 통합, 저장, 처리될 수 있는 기반을 제공하기 위해 HTML 문서의 XML 변환 기능을 포함하고 XML 문서에 대한 효율적인 저장 및 검색을 지원하는 XML 래퍼 시스템을 설계하고 구현하였다.

개발된 XML-RDB 래퍼 시스템을 사용함으로써 HTML 문서로 구성된 대용량 웹 데이터베이스로부터 필요한 정보만을 추출하여 XML 문서로 변환한 뒤 데이터베이스에 저장하고 검색 결과를 다양한 인터페이스를 통해 제공하는 것이 가능하다. 추출된 정보를 XML 문서로 변환한 뒤 효율적인 관리를 위해서 DTD 의존적인 엘리먼트 분할저장 형태의 XML 저장 스키마 구조를 적용하였다.

제안한 저장 구조는 내용 검색 및 구조 검색이 용이할 뿐만 아니라 본래의 XML 문서로의 완벽한 복원이 가능하다는 이점을 갖는다. 개발된 XML-RDB 래퍼 시스템은 저장된 XML 문서에 대해 독립된 파일 단위의 접근과 유연한 검색 형식을 지원할 수 있어 적용 예에서 보는 바와 같이 대용량 웹 데이터베이스를 통합하기 위한 기본 토대가 될 수 있다.

추후 연구로는 HTML 문서의 XML 문서로의 변환을 지원하는 자동화된 래퍼 생성 방법과 대용량 XML 문서를 처리할 수 있도록 성능 개선에 대한 연구가 필요하다.

## Acknowledgement

본 논문은 한신대학교 신진교수공모과제 지원에 의한 연구의 일부임

## 참고 문헌

- [1] W3C, Extensible Markup Language(XML) 1.0, <http://www.w3.org/TR/1998/REC-xml19980210.html>, Feb. 1998.
- [2] P.A. Bernstein and U. Dayal, An Overview of Repository Technology, Proceedings of the 20rd VLDB Conference, Santiago Chile, 1994, pp. 705~713.
- [3] 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.
- [4] Jayavel Shanmugasundaram, Kristin Tufte, Chun Zhang, Gang He, David J. DeWitt, Jeffrey F. Naughton, "Relational Databases for Querying XML Documents : Limitations and Opportunities" VLDB 99, pp. 302~314, 1999.
- [5] Takeyuki Shimura, Masatoshi Yoshikawa, Shunsuke Uemura "Storage and Retrieval of XML Documents Using Object-Relational Databases" DEXA99, pp. 206~217, 1999.
- [6] Daniela Florecu and Donald Kossmann, "A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database" INRIA TR, 1999.
- [7] S. Malaika, "Using XML in Relational Database Applications," 15th Int'l Conf. on Data Engineering, Sydney, Australia, p.167, 1999.
- [8] T. Shimura, M. Yoshikawa, and S. Uemura, "Storage and Retrieval of XML Documents Using Object-Relational Databases," DEXA99, pp. 206~217, 1999.
- [9] M. Stonebraker and P. Brown, Object-Relational DBMSs: The Next Great Wave, Morgan Kaufmann Publishers, Inc., 1996.



- [10] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, Object-Oriented Modeling and Design,
- [11] P. Francois, "Generalized SGML Repositories: Requirement and Modeling," Computer Standard & Interface, 1996.
- [12] S. Malaika, "Using XML in Relational Database Applications," 15th Int'l Conf. on Data Engineering, Sydney, Australia, pp.167, 1999
- [13] Daniela Florescu, Donald Kossmann, "Storing and Querying XML Data using an RDBMS" IEEE Data Engineering Bulletin 22(3) : pp.27-34, 1999.
- [14] P. Francois, "Generalized SGML Repositories : Requirements and Modeling," Computer Standard & Interface, 1996.
- [15] Ronald Bourret, XML-DBMS, <http://www.informatik.tudarmstadt.de/DEVS1/staff/bourret/xmldbms/readme.html>, 2000.
- [16] Volker Turau, DB2XML 1.3, <http://www.informatik.fhwiesbaden.de/~turau/DB2XML/index.html>, 2000.

## ● 저 자 소 개 ●



### 박 성 진

1991년 고려대학교 전산학 (학사)

1993년 고려대학교 일반대학원 전산학 (석사)

1998년 고려대학교 일반대학원 전산학 (박사)

1998년~2000년 : 한국전자통신연구원 선임연구원

2000년~현재 : 한신대학교 정보시스템공학과 조교수

주관심분야 : XML Respository, Web OLAP & Mining