XML2RDB와 RDB2XML의 아키텍쳐 구축에 대하여
- Table-Based 자료생성 XML과 RDB의 매핑모델의 설계를 중심으로 -

이건식, 한국정신문화연구원 한국학정보센터, leeks@aks.ac.kr

Keon-Sik Lee, Korean Studies Information Center, Academy of Korean Studies

이글은 인터넷기반 정보시스템에서 XML의 존재 층위를 나누고 각 층위에 각 층위에 적합한 XML과 RDB의 매핑 모델을 제안했다. 그리고 자료생성 층위에 주목하여 XML과 RDB의 매핑 모델을 설계하고 그 절차에 대해 논의했다. 이 연구에서 제안한 XML과 RDB 매핑 모델은 DTD 구조에 유연하며 플랫폼 독립적인 점에서 정보시스템 구축의 비용을 저렴하게 한다.

1. 서언

이 글은 Table-Based 자료생성 XML과 RDB의 매핑 모델을 제안하는 데에 연구의 목적이 있다.

Ronald Bourret(2004)에 따르면 XML과 RDB 간의 매핑 방법으로는 Table-Based Model 매핑과 Object-Relation Model 매핑의 두가지가 있다. 그리고 XML 전용 데이터베이스인 Tamino나 XML-Enabled 데이터베이스인 SQL과 Oracle 등에서 Object-Relation Model 매핑에 기반한 모듈을 제공하고 있다. 한편 한국과학기술정보연구원에서 XML 매퍼 자바클래스를 개발 보급하고 있으나 XML과 RDB의 매핑의 논리구조에 대한 설명문서는 쉽게 접할 수 없으며 국내 XML 전문업체들이 개발하여 보급하는 XML 매퍼 역시 XML과 RDB의 매핑에 관한 논리구조를 명쾌하게 설명하고 있지 않다.

정보시스템의 구축은 정보기술로만 이루어지는 것이 아니다. 오히려 컨텐츠에 대한 깊은 이해를 통해서 좋은 정보시스템이 구축된다고 생각한다. 한국정신문화연구원 역사정보통합시스템은 2001년도에 XML과 RDB의 자료생성 매핑 시스템을 개발하여 정보시스템을 구축한 적이 있었다. 이 매핑 시스템은 새로운 DTD가 정보시스템에 추가될 때마다 XML과 RDB의 자료생성 매핑 모듈을 수정해야 했다. 이것은 매우 불편할 뿐만 아니라 비용적으로도 낭비적인 요소가 많아서 금년도 정보시스템의 업그레이드 개발에서는 DTD에 상관없는 XML과 RDB의 자료생성 매핑 시스템을 사업자에 주문하였다. 그러나 몇몇 XML 전문업체들은 이 문제에 대해 난색을 표명하였다. 그러한 시스템은 불가능하다는 느낌을 주는 회의가 연속되었다. 오늘 이 연구에서는 DTD에 상관없는 XML과 RDB의 자료생성 매핑 시스템의 개발이 가능함을 입증하고자 한다.

오늘 이 연구에서 제안하는 XML과 RDB의 자료생성 매핑 시스템은 XML 전용 데이터베이스나 XML-Enabled 데이터베이스 시스템에서 제공하는 매핑 모듈로 구현할 수 있다. 하지만 이러한 모듈에 의지하지 않고 RDB, 자바, XSL 등의 보편적 기술을 사용하여 매핑의 모델을 설정하고 그 구현방법을 제안하는 것은 몇 가지 점에서 의의를 갖는다고 생각한다.

첫째는 상업용 XML 매퍼는 적용할 수 있는

플랫폼이 제한적이지만 일반적 정보기술을 사용해서 개발된 XML 매퍼는 플랫폼에 독립적인 점에서 의의를 갖는다.
둘째는 XML과 RDB의 매핑이라는 정보기술의 주제를 보다 심도있게 이해할 수 있고 이러한 이해가 바탕이 되어서 컨텐츠에 적합한 기초적인 정보기술을 독자적으로 개발할 수 있는 토대를 마련할 수 있다는 것이다.

2. Table-Based 자료생성 XML과 RDB의 매핑모델

2.1 XML의 존재층위

인터넷 기반 정보시스템에서 XML은 데이터 저장소 층위, 자료생성 층위, 미들웨어 층위, 정보서비스 층위에 존재한다.
데이터 저장소층위에서 XML은 파일시스템으로 존재할 수도 있고 XML의 요소들이 RDB의 Table이나 Field로 분해되어 존재하거나 이 둘을 혼합한 형태로 존재할 수 있다. 정보시스템들은 자료의 무결정성 확보를 위해 파일시스템을 버리고 RDB시스템을 채택하고 있으므로 이 연구에서도 XML을 RDB로 분해하여 저장하는 것을 기준으로 한다.
자료생성층위는 자료를 XML 형태로 구축해서 RDB에 적재하는 층위를 말한다. 이 자료생성에는 클라이언트에서의 자료생성과 서버측에서의 자료생성의 두 가지가 있다. 전자는 소량이며 수시로 그 동작이 수행되지만 후자는 대량이며 일시에 그 동작이 수행된다.
정보서비스층위에는 사용자에게 정보를 제시하는 것과 타 정보시스템과 자료를 교환하는 두 가지가 있다. 이 층위의 정보서비스를 XML 기반으로 구성할 때 정보서비스의 유연성이 확보되는 것은 두 말할 나위 없다.
미들웨어 층위는 데이터저장소층위, 자료생성층위, 정보서비스층위를 통합관할하고 연결하는 층위이다. 이러한 통합관할과 연결을 XML 기반으로 수행할 때 효과도 두 말할 필요가 없다.

2.2 XML과 RDB의 매핑

자료생성층위의 XML, 자료교환 층위의 XML, 정보제시 층위의 XML은 미들웨어층위의 XML을 통하여 RDB와 매핑된다. 그렇다면 Ronald Bourret(2004)이 제안한 Table-Based Model 매핑과 Object-Relation Model 매핑 중 어떤 매핑 방법을 선택해야 할 것인가의 문제가 제기된다.
Table-Based Model 매핑은 XML과 RDB의 TABLE을 평면적으로 연관시키는 방법이다. 따라서 RDB 내에서의 TABLE들 간의 연결관계는 XML과 RDB의 매핑에서 제외하는 방법이다. 이에 반해서 Object-Relation Model 매핑은 RDB 내에서의 TABLE들 간의 연결관계를 XML 요소들의 포합관계로 유지하는 방법이다. TABLE_1이 마스터 테이블이고 TABLE_2가 디테일 테이블 경우에 이 둘의 연결관계는 XML에서 다음과 같이 유지된다.

```
<XMLROOT>
        <TABLE_1>
                <ELEMENT_1>
                </ELEMENT_1>
                <ELEMENT_2>
                        <TABLE_2>
                        </TABLE_2>
                </ELEMENT_2>
        </TABLE_2>
</XMLROOT>
```

위에서 보는 바와 같이 Table-Based Model 매핑에서는 RDB 스키마의 의미가 유지되지 못하나 Object-Relation Model에서는 RDB 스키마의 의미가 유지된다.
자료교환 층위, 정보제시 층위에서는 RDB 스키마의 모든 의미관계가 실현되기도 하고 그렇지 않은 경우도 있다. 자료교환이나 정보를 제시할 때 필요한 의미만 주고 받을 수 있으며 경우에 따라서는 모든 의미를 주고 받거나 제시할 수 있다. 이에 반해 자료생성층위에서는 항상 RDB의 모든 의미관계가 실현되어야 한다. 자료생성층위에서의 XML은 RDB 스키

마의 의미관계를 포함해야 할 이유는 없으므로 단순한 구조의 매핑방법인 Table-Based Model을 선택이 효과적이다. 한편 자료교환 층위, 정보제시 층위에서는 의미관계의 취사선택이 이루어지므로. Object-Relation Model을 선택하는 것이 효과적이다.

2.3 Table-Based 자료생성 XML 매핑 모델의 DTD

Table-Based 자료생성 XML 매핑에서 XML의 DTD는 RDB의 TABLE과 평면적으로 연관된다. RDB의 TABLE를 DTD로 표현하면 다음과 같다.

```
<!ELEMENT dbdata
        (jdbc_info, user-info, table*)>
<!ELEMENT jdbc_info EMPTY>
<!ATTLIST jdbc_info
  driver CDATA "com.sun.jdbc.odbc"
  db_url CDATA "YourFavoriteDB"
>
<!ELEMENT user-info (uname, upass)>
<!ELEMENT uname (#PCDATA)>
<!ELEMENT upass (#PCDATA)>
<!ELEMENT table (headers, row*)>
<!ATTLIST table
  name CDATA #REQUIRED
  reset (true|false) "false" >
<!ELEMENT headers (header+)>
<!ELEMENT header EMPTY>
<!ATTLIST header
  name CDATA #REQUIRED
  fieldType
(CHAR|VARCHAR|LONGVARCHAR|NUMERIC|DECIMALBIT|TINYINT|SMALLINT|INTEGER|BIGINT|REAL|FLOAT|DOUBLE|BINARY|VARBINARY|LONGVARBINARY|DATE|TIME|TIMESTAMP) "VARCHAR" >

<!ELEMENT row (field*)> <!-- number of field <= number of headers -->
<!ELEMENT field (#PCDATA)>
<!ATTLIST field
  name CDATA #REQUIRED
>
```

RDB매핑을 위한 XML은 ODBC 접속을 위한 요소(driver,db_url), DB 계정 요소(uname), DB Table 요소(fieldName, FieldType, row, field, fieldname)를 갖는다.

2.4 Table-Based 자료생성 XML 매핑 자바 클래스

이 모듈은 여러 가지로 작성될 수 있으나 이 연구에서는 우선 Professional Java XML Programming 제 9장에 소개된 자바 클래스를 이용하기로 한다. 이 클래스는 공개된 것이다.

2.5 템플릿 XML의 생성절차

템플릿 XML은 4)항에서 말한 XML이다. RDB에 입력되는 원시 XML들은 이 템플릿 XML로 변환된 다음에 RDB 테이블에 입력된다. 이러한 동작원리는 XML 구조에 상관 없는 자료 생성 XML 모듈을 설계하는 것이다. 우리는 이 템플릿 변환 모듈로 XML 매핑 자바클래스는 한번 코딩되면 그만이다.

상이한 구조의 원시 XML을 RDB의 테이블에 연관시키면서 템플릿 XML을 만들기 위해서는 우선 원시 XML과 RDB TABLE의 필드들과 매핑시키는 XML을 작성해야 한다. 이 매핑 XML은 다음과 같다.

```
<maptable>
        [생략]
<rowdef>
        <간행년 RDBname="year"/>
        <서명 RDBname="title"/>
        <소분류 RDBname="class_3"/>
</rowdef>
</maptable>
```

위 XML에서 요소 '간행년', '서명', '소분류'는 원시 XML에 존재하는 요소들이다. 여기에 정의되지 않은 원시 XML은 템플릿 XML로 전송되지 않는다. 그리고 이들은 각각에 지정된 'RDBname'의 속성 필드로 원시 XML의 데이터가 전송된다.

RDB에 데이터를 전송할 때 마다 위의 XML을 수작업으로 원시 XML에 첨가할 수도 있으나 다음과 같은 XSL의 기능을 이용하여 매핑규칙 XML과 원시XML을 결합하는 것이 매우 효과적이다.

```
<?XML version="1.0"
                encoding="UTF-16"?>
<xsl:stylesheet version="1.0"
XMLns:xsl="http://www.w3.org/1999
        /XSL/Transform">

<xsl:output encoding="UTF-16"
        version="1.0" indent="yes"/>

<xsl:template match="/">
<dbdata>
<xsl:apply-templates select
        ="dbdata/maptable"/>
<xsl:apply-templates select
        ="dbdata/data"/>
</dbdata>
</xsl:template>

<xsl:template match="dbdata/maptable">
<xsl:apply-templates select
="document(@filename)//mapdtd"/>
</xsl:template>
<xsl:template match="//mapdtd">
<xsl:copy-of select="/"/>
</xsl:template>

<xsl:template match="dbdata/data">
<xsl:apply-templates select
="document(@filename)/data"/>
</xsl:template>

<xsl:template match="/data">
<xsl:copy-of select="/"/>
</xsl:template>
</xsl:stylesheet>
```

매핑규칙 XML과 원시 XML이 결합된 XML이 만들어졌으므로 이 XML을 템플릿 XML로 변환하도록 한다. 이를 위한 XSL은 다음과 같은데 지면 관계상 별지를 참조하기 바란다.

2.6 XSL의 기능

XSL은 강력한 기능을 가지고 있으며 동작의 내부를 일반 에디터에서 볼 수 있다는 점에서 정보시스템 구축에 매우 큰 효과를 볼 수 있다. 정보시스템의 자료 생성에 있어서 항상 문제가 되는 것은 데이터의 결함이다. 데이터의 결함은 자료의 무결정성, 크기의 초과, 데이터의 결여 등 수 많은 양상으로 전개된다. 우리는 XSL의 강력한 기능을 이용하면 우리는 많은 효과를 볼 수 있다.

3. 시연

4. 결언

이글은 인터넷기반 정보시스템에서 XML의 존재 층위를 나누고 각 층위에 각 층위에 적합한 XML과 RDB의 매핑 모델을 제안했다. 그리고 자료생성 층위에 주목하여 XML과 RDB의 매핑 모델을 설계하고 그 절차에 대해 논의했다. 이 연구에서 제안한 매핑 모델은 DTD 구조에 무관하며 플랫폼 독립적인 점에서 정보시스템 구축의 비용을 저렴하게 한다. 이 연구에서 제안한 모델은 XSL의 기능을 중점으로 강화하는 방향에서 이루어졌다. 이것은 정확한 데이터 구축에 매우 효과적 기능을 가질 것이라 기대한다.

참고문헌

김정희, 김휴찬, 곽호영(), Legacy 데이터베이스를 위한 XML 게이트웨이 설계 및 구현.

김종민, 노영만 공역(2002), Professional XML Second Edition, 정보문화사.

안성욱 외 공역(2000), Bdggining XML, 정보문화사.

안성욱 외 편역(2001). XSLT Programmer's Refference, 정보문화사.

유진희, 박성준 역(2000), Professional Java XML Programming, 정보문화사.

장은영역(2002), 소스코드로 배우는 XSLT, 한빛미디어.

진장일, 송동혁 역(1999), Oracle XML Application, 한빛미디어

최효진, 안성욱 외 역(2001), Professional XML Databases, 정보문화사.

Ronald Bourret(2001), Mapping W3C Schemas to Object Schemas to Relational Schemas.

Ronald Bourret(2004), XML And Databases.