

XML을 이용한 관계DB의 웹출판에 관한 사례

(A Case Study on the Web Publishing of Relational DB Via XML)

우 원 택*

목 차

I. 서언	V. 결론
II. XML의 기본구조	* 참고문헌
III. XML변환의 과정	* Abstract

I. 서언

관계 데이터베이스 테이블을 XML(Extended Markup Language)파일로 변환하는 주요 목적은 전자상거래나 기업 간(Business-To-Business)의 웹 응용에서 데이터베이스 파일을 효율적으로 이용하기 위한 것으로 관계형 구조의 테이블을 태그를 가지는 계층형 파일구조인 트리구조로 변형하여 각각의 요소¹⁾를 XSLT(Extensible Stylesheet Language Transformations)를 이용하여 원하는 형태로 쉽게 편집 변환하여 이를 다시 HTML형태로 웹상에 뿌려주고 반대로 웹 상에서 XSLT형태로 변환된 XML파일이 전송되면 이를 다시 관계형 데이터 테이블로 변환하여 데이터 베이스에 저장할 수 있다. 이렇게 함으로서 기존의 SQL 서버나 Access 등의 DB를 웹과 함께 더 효율적으로 HTML 형식으로 연동하여 사용할 수가 있다. XML은 그 자체가 데이터를 포함하며 태그가 메타데이터로 표현되기 때문에 데이터베이스 요소를 갖고 있다. 하지만 이 데이터를 처리할 수 있는 여러 요소가 추가돼야 한다. 일반적인 데이터베이스를 이루는 요소로 데이터 스토리지, 스키마, 쿼리언어, 프로그래밍인터페이스 등이 필요하다. XML 문서는 이 가운데 데이터 스토리지에 해당하는 부분이다. 현재 스키마와 관련해서는 DTD나 XML 스키마 언어가 있고, 쿼리 언어로는 XQL, XML-QL 등이 있어 있으며, 프로그래밍 인터페이스로는 DOM, SAX 등이 이용되고 있다. 현재 미국의 스탠포드대학에서는 Lore라는 XML문서전용 데이터베이스를 개발하고 있다. ²⁾따라서 XML은 데이터마이닝 또는 데이터웨어하우스를 웹DB에서 효율적으로 사용할 수 있는데도 응용될 수 있다. 본 연구에서는 XML의 기본 구조를 소개하고 관계데이터의 한 예를 들어 이를 XML 파일로 변환하는 과정과 XML 파일을 다시 XSL(Extensible Stylesheet Language)을 이용하여 HTML로 변환하는 과정을 분석해 보고 VB Script와의 프로그래밍인터페이스등을 살펴봄으로서 향후의 전자상거래응용에서 이를 효율적으로 이용

* 경산대학교 정보과학부 조교수

1) 관계데이터베이스에서는 이를 속성(Attribute)이라 하고 XML문서에서는 이를 요소(Element)라고 한다.

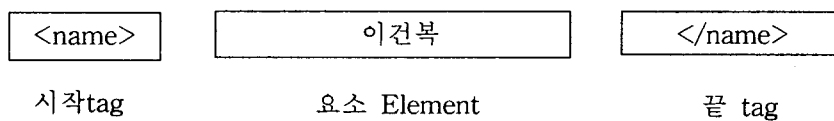
2) Roy Goldman, Jason McHugh and Jennifer widom, "Lore: A Database Management System for XML", Dr. Dobb's Journal, April 2000, p.76

해 보고자 함에 있다.

II. XML의 기본구조

1. XML과 웹

오늘날 문서 데이터의 요구사항은 사람과 시스템이 모두 동시에 읽을 수 있는 형태를 요구한다. 이를 위해서는 자료의 내용과 구조를 동시에 정의할 필요가 있으며 이들간의 관계도 설정되어야 한다. 이러한 목적으로 탄생한 XML문서는 데이터를 설명하기 위해 태그를 사용하며 정보를 나타내는 기본 요소의 표현방식은 다음과 같다.



<그림 1> XML 기본요소의 표현 방식

자료: Jump to.net 전국대학투어세미나, 「XML, Visual Studio.NET C#, SQL Server 2000」, 월간마이크로소프트웨어, 데브피아, KCC교육센터, 마이크로소프트, 2001, p.50

XML문서는 하위 요소를 포함할 수 있으며 기본적인 문서의 구성은 선언부, 주석, 상위 요소, 자식요소, 빈 요소, 속성으로 구성되어 있다. 아래에서 상위요소는 <employee>이며 자식요소는 <name>, <permanent_staff>, <saraly>, <picture>이며, 빈 요소는 <permanent_staff>나 <picture>와 같이 표현할 수 있다. 그리고 속성지정은 <saraly>요소에서 화폐단위인 currency와 같이 이를 지정해 줄 수가 있다.

```
<? xml version="1.0"?>
<!-- 이것은 주석입니다 -->
<employee>
  <name>오세영</name>
  <permanent_staff />
  <saraly currency="원">
    2500000
  </saraly>
  <picture filename="PID1.jpg"/>
</employee>
```

<그림 2> XML문서의 구성

자료: Jump to.net 전국대학투어세미나, 「XML, Visual Studio.NET C#, SQL Server 2000」, 월간마이크로소프트웨어, 데브피아, KCC교육센터, 마이크로소프트, 2001, p.50

그러나 XML문서는 자료의 구조와 출력하여 보여주는 부분을 분리함으로써 데이터 교환의 개방성과 확장성을 높이고 있다. 즉 XML 문서는 문서의 내용과 구조에 대한 정보를 가지고는 있으나 출력에 대한 내용은 정의 하지 않는다. 이렇게 함으로써 XML은 HTTP의 트랜잭션(transaction)에 사용할 수 있게 되었다. 웹서버에서 XML을 사용할 경우의 장점은

하나의 XML파일을 모든 브라우저에서 사용이 가능하고 반면 각각의 브라우저는 자신의 포맷을 변형하지 않고 그대로 유지할 수가 있다. 한편 XML을 사용하는 브라우저의 장점은 데이터를 재 활용할 수가 있고 네트워크의 트래픽과 서버의 로드를 줄일 수가 있다는 것이다. 따라서 비즈니스의 업무를 자동화 할 수 있으며³⁾ 정보의 분산 전달을 가능하게 하며 지식 관리(Knowledge management)와 응용프로그램과 데이터의 통합을 용이하게 할 수가 있다. 그러나 XML은 확장이 용이하며 호환성이 높고, 자체적으로 데이터를 설명한다는 장점을 가지고는 있으나 새로운 언어를 또 만듦으로써 또 다른 혼란을 야기할 수 있다는 문제점과 구조를 정의하기는 하나 의미적 데이터를 표현하는데는 여전히 다소의 제한적인 단점을 가지고 있다. XML을 만들기 위한 도구로는 일반편집기, XML전문편집기, Microsoft Visual InterDev, VS.NET beta1등이 있으며 다른 유용한 도구로는 XML Tree Viewer, XML Validator, XSL Debugger등이 있다. HTML과 XML의 차이점을 보며는 아래와 같다. 즉 HTML은 데이터를 단순히 표시하는 기능만 하는 반면 XML은 표시하지는 않고 표시할 데이터를 정의만 한다는 점이 다른 점이다.

```
<TABLE>
  <TR>
    <TD>이름</TD><TD>최우인</TD>
  </TR>
  <TR>
    <TD>월급</TD><TD>100</TD>
  </TR>
  <TR>
    <TD>지역</TD><TD>서울</TD>
  </TR>
</TABLE>
```

<그림 3> HTML은 단지 데이터를 표시

```
<employee>
  <name>최우인</name>
  <salary>100</salary>
  <region>서울</region>
</employee>
```

<그림 4> XML은 단순히 데이터를 정의

2. XSL과 CSS

XSL은 XML의 스타일시트(Style Sheet)⁴⁾로 전술한 바와같이 XML은 문서의 출력구조는 가지고 있지 않기 때문에 XSL이 이 기능을 담당하게 된다. 즉 일반 편집기에서 여러 방식의 출력구조를 CSS(Cascading Style Sheets)가 담당하듯이 스타일시트 데이터의 포맷을 정의 할 수있다. 일반편집기와 같이 HTML문서의 출력양식도 CSS문법을 이용하여 아래와 같이 출력문서의 양식을 정의할 수도 있다.

3) 예를 들면 Electronic Data Interchange(EDI)가 있다.

4) 출력형식을 지정하는 언어

```

<HEAD>
  <STYLE TYPE="text/css">
    H1 {font-family:Comic Sans MS; font-size:36pt}
    P {font-family:Arial; font-size:16pt; color:red}
  </STYLE>
</HEAD>

<BODY>
  <H1>사용자 정의 헤더</H1>
  <P>패러그래프</P>
  <P>기존의 태그의 재정의 기능</P>
</BODY>

```

<그림 5> HTML문서에 CSS를 사용한 사례

그러나 XML의 경우는 XSL을 이용하여 XML문서의 요소들과 속성들을 다른포맷으로 변환하는 방식을 정의한다. 따라서 CSS 보다는 더 강력하고 유연하다.

```

CSS를 이용한 XML문서 포맷 사용 예
employee {display:block; margin:10px}
name {display:inline; font-family:Comic Sans MS;
      color:red}
salary {display:inline; font-family:Arial;
        font-size:24pt}

XML문서에 CSS 첨부
<?xml-stylesheet type="text/css" href="Employee.css"?>
<employees>
  <employee>
    <name>최우인</name>
    <salary>100</salary>
  </employee>
</employees>

```

<그림 6> XML문서에 CSS 사용의 사례

자료: Jump to.net 전국대학투어세미나, 「XML, Visual Studio.NET C#, SQL Server 2000」, 월간마이크로소프트웨어, 테브피아, KCC교육센터, 마이크로소프트, 2001, p55

```

XML문서에 첨부
<?xml-stylesheet type="text/xsl"
  href="URL for XSL file" ?>

예
<?xml-stylesheet type="text/xsl" href="Employees.xsl"?>
<employees>
  <employee>
    <name>최우인</name>
    <salary>100</salary>
  </employee>
</employees>
</employees>

```

<그림 7 > XSL Style Sheet의 적용 예

XSL의 기능은 새로운 요소와 속성 추가가 가능하고 또 이들을 정렬할 수도 있으며 순환, 조건문의 사용도 가능하다. 그리고 XML문서에 XSL을 첨부하여 사용할 수도 있다.

3. XML과 HTML의 연결

XML과 HTML의 연결은 우선 클라이언트에 데이터를 보내면 XML Data Island가 생성되며 XML 데이터와 HTML Elements 간에 연결이 형성되며 HTML문서에서 XML 데이터가 표시된다. 웹서버에서 클라이언트로 데이터를 보내는 방법으로는 첫째 HTML페이지⁵⁾를 보내거나 둘째 XML문서를 보내거나 셋째 XML data island를⁶⁾ 보내는 방법이 있다.

```

<xml ID = "MyXMLData">
<? xml version="1.0" ?>
<employees>
  <employee>
    <name>최우인</name>
    <salary>100</salary>
  </employee>
</employees>
</xml>

<INPUT TYPE ="TEXT"
  DATASRC="#MyXMLData"
  DATAFLD="salary">

```

<그림 8 >XML Data와 HTML Element의 연결 예

상기와 같이 XML data를 HTML element에 연결한 후 아래와 같이 XML data를 HTML에 테이블 형식으로 표시할 수 가 있다.

5) HTML elements와 HTML script를 보낸다.
6) XML data + HTML elements + HTML script를 보낸다.

```

<TABLE DATASRC="#MyXMLData">
  <TR>
    <TD><SPAN DATAFLD="name"></TD>
    <TD><SPAN DATAFLD="salary"></TD>
  </TR>
</TABLE>

```

<그림 9 > XML 데이터를 HTML 테이블로 표시한 예

4. DOM과 XML

Document Object Model(DOM)이란 XML문서에 대한 프로그램을 할 수 있는 인터페이스를 제공하는 것으로 XML문서의 객체와 인터페이스를 가지고 있다. 즉 XML트리를 가지고 프로그램화 할 수 있고 XML의 요소(Elements)와 속성(attributes) 등을 추가 삭제할 수 있으며 새로운 XML문서를 프로그램적으로 생성이 가능하다. DOM을 이용한 XML사용은 최상위 element를 사용하며 DOM트리의 다른 노드의 사용도 가능하다.

```

<xml ID="MyXMLData">
<?xml version="1.0" ?>
<employees>
  <employee> .....
  <employee> .....
</employees>
</XML>

<SCRIPT LANGUAGE="VBScript">
Sub SampleSub()
  Set doc = MyXMLData.XMLDocument
  Set rootNode = doc.documentElement
  .....
End Sub
</SCRIPT>

```

<그림 10> DOM을 이용한 XML Data Island의 사례

5. XML검사

XML문서는 원하는 형식에 맞게 작성 교환할 수 있도록 검사(Validation)가 필요하다. 이는 올바른 요소와 속성을 사용하였는지 요소와 속성간의 관계가 잘 정의되어 있는지 하부 요소의 숫자와 순서가 올바르게 정의되어 있는지 XML 스키마(schema)에서⁷⁾ 언급된 데이터 형식을 따르고 있는지를 검정한다. 이에 따라 Well Formed XML문서와 Valid 문서를 구분하게 된다. 전자는 XML의 기본문법을 준수하고 요소와 속성의 조합으로 구성되며 문법을 가지고 있다. 후자는 well formed 문서로 Document Type Definition(DTD) 또는 XML schema로 정의된 문법을 사용하는 XML 문서이다. DTD는 사실 단어의 일반적인 의미로는 스키마의 한 종류이다. DTD에서는 모든 요소가 포괄적으로 선언되는데 반해

7) XML스키마는 여러 데이터소스와 여러 데이터파일에서 데이터와 선언 모두를 가져올 수 있으며 이들 데이터의 일부는 데이터베이스, CGI, 스크립트 혹은 파일이 아닌 다른 소스에서 직접 가져올 수 있다.

XML 스키마에서는 지역적 선언과 포괄적 선언이 가능하기 때문에 XML스키마는 XML 문서의 콘텐츠에 대해 더 큰 장악력을 가지게 한다. 현재 DTD는 표준으로 정착된 단계며 Parser가 지원되고 있으나 복잡하고 데이터 형식에 대한 자세한 정보가 없고 추가적인 태그 사용을 허용하지 않는 제한점을 가지고 있다. XML스키마는 현재 DRAFT가 표준으로 되어 있고 기능이 보다 풍부한 것이 장점이다.

```

<! DOCTYPE myDocType
<! ELEMENT .....>
<! ATTLIST .....>
]>

```

<그림 11 > DTD의 보기

```

내부 DTD의 예
<?xml version="1.0"?>
<! DOCTYPE employees [
.....ELEMENT와 ATTLIST 정의.....
]>
<employees>
..... XML 자료 .....
</employees>
외부 DTD의 예
<? xml version="1.0"?>
<! DOCTYPE employees SYSTEM "Emploees.dtd">
<employees>
..... XML 자료 .....
</employees>

```

<그림 12 > DTD를 XML문서에 사용한 사례

XML 스키마는 최상부에 스키마(schema)를 표시하고 이름을 선언한 후 XML문서의 요소를 선언한다. 그리고 XML문서의 요소를 선언한다. 즉 XML문서에 대한 스키마를 정의 한 후 특정 요소에 대한 스키마를 정의하여 이름공간(namespace)를 사용하여 이를 적용한다.

```

<Schema name="mySchema" .....>
  <ElementType .....>
  .....
  </ElementType>

  <AttributeType .....>
  .....
  </AttributeType>
  <description>
  .....
  </description>
</Schema>

```

<그림 13> XML스키마의 보기

```
<book xmlns="x-schema:http://microsoft.com/myschema.xml">
  <title>C#프로그래밍</title>
  <isbn>72-80081-082</isbn>
  <author>이건복</author>
  <author>최현진</author>
</book>
```

<그림 14> XML스키마의 적용 예

III. XML변환의 과정

1. 기술적 개요

XML문서는 웹서버를 통해 Database로부터 생성된 여러개의 데이터 소스로부터 데이터를 가져와 CSS 또는 XSL등의 스타일시트를 적용해 사용자의 웹 브라우저에서 사용 가능한 HTML뷰로 생성된다. 다시말하면 데이터베이스에서 생성된 관계테이블이 데이터베이스의 XML export기능이나 XML Query에 의해 계층구조의 XML파일로 변환되어 HTTP를 통해 웹으로 전송되면 XSLT의 포맷변형을 통하여 새로운 모양으로 데이터베이스가 출력된다. 최근의 데이터베이스는 순수 XML저장소를 제공함으로써 완벽한 웹DB기능을 제공한다. 즉 XML문서의 저장, 검색 및 생성과 Open XML과 XML Updategram을 통한 데이터베이스 내의 관계 테이블을 수정, 삽입, 삭제가 가능하며 URL을 통한 쿼리, 영문쿼리(English Query)도 가능하다. XSLT를 이용한 XML파일의 변환과정을 보면 XSLT의 작업과정은 XML파일 해석기능을 가지고 있는 브라우저에서⁸⁾ XML파일을 열게되면 웹브라우저는 우선 내장파서를⁹⁾ 이용하여 XML문서의 각 요소를 분석하여 트리구조로 변형한다. 예를들면 아래와 같이 데이터베이스에서 XML문서로 변환된 Employee관계테이블이

```
<?xml version="1.0" encoding="UTF-8" ?>
<dataroot>
  <Employees>
    <EmployeeID>1</EmployeeID>
    <LastName>Davolio</LastName>
    <FirstName>Nancy</FirstName>
    <Title>Sales Representative</Title>
    ...
```

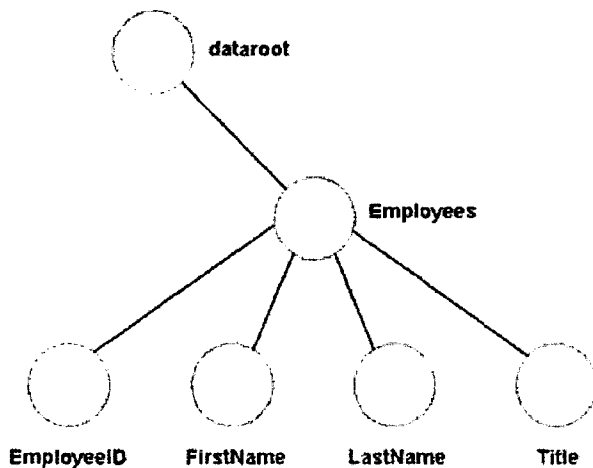
<그림 15 > Employee 관계테이블의 XML파일변환의 예

브라우저로 전송되면 다음 그림과 같은 트리구조로 분석된다.¹⁰⁾ 분석은 변형을 용이하게 하기 위한 것으로 관계테이블이 트리구조로 변하는 것은 계층구조데이터로 바뀌는 것을 의미한다.

8) 예를들면 Microsoft Internet Explorer 5.0 이상

9) 예를들면 Microsoft XML Parser와 같은 언어분석기

10) 즉 파싱(Parsing)된다.



<그림 16> Employee관계테이블의 계층구조 변환

자료:http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc2k2/html/odc_acc2xslt.asp

분석(Parsing)과 변형(Transformation)은 메모리에서 실행되며 입력파일에서 행하여지는 것은 아니다. 즉 입력파일의 당초의 물리적 구조는 변형되지 않으나 이를 메모리 상에서는 여러 형태로 변형하여 쓸 수가 있다. 입력데이터는 파싱된 후 메모리에 저장되어 변형된다. 변형을 위해 브라우저는 참조할 XSLT파일을 찾게되는데 만약 입력파일이 XSLT문서나 XSLT를 불러오는 스크립트 명령어를 가지고 있다면 브라우저는 그 XSLT파일을 메모리로 읽어들인 후 내장프로세스를 이용하여 XML데이터 트리구조를 XSLT트리구조로 변환한다.¹¹⁾ 만약 입력 XML데이터 파일에 XSL/XSLT참조문이 없다면 데이터는 원시XML(Raw XML)이나 다른 디폴트구조(Default Structure)로 출력된다.¹²⁾ XSL/XSLT파일이 XML을 변형 중에는 XSL처리기(Processor)는 XSLT명령들에 있는 패턴과 XML입력트리에 있는 특정 요소들과¹³⁾ 일치 시키면서 XSLT트리를 순환한 후 어떤 일치하는 패턴이 발견되면 브라우저는 XML요소나 그 요소의 자손이나 후손들에 적용된 기본규칙(Template Rules)들을 일치시키기 위해 XSLT트리를 탐색한다. 그 결과 데이터를 포맷하는데 필요한 요소(Elements), 데이터, HTML태그나 규정들을 가지고 있는 출력트리를 생성한다. 출력의 내용이나 표현은 데이터를 포맷하기 위한¹⁴⁾ HTML태그들과 데이터 요소들의 구조를 조정하기 위한 일련의 명령들을 제공하는 입력데이터와 스타일시트에 그 기초를 두고 있다. 변환과정을 완성하기 위해 브라우저는 출력트리를 XML파서를 이용하여 분석하고 HTML문서나 변형중에 규정된 어떤 형식으로든 이를 출력하게 된다. XSLT는 최하위수준에서 하나의 트리구조에서 다른 트리구조로 변환하는데 XSLT처리기는 입력XML과 데이터, XSLT스타일시트, 출력XML

11) 인터넷 익스플로러 5.0 이상의 최신 브라우저는 XML Parser와 XSL Parser 같은 처리기(Processor)를 내장하고 있다.

12) 인터넷 익스플로러 5.0 이상의 브라우저는 디폴트 XSL스타일시트(Style Sheet)를 내장하고 있어서 출력형식이 규정되어 있지 않으면 XML파일을 확장가능 트리 구조(Expandable Tree-Like Structure)로 보여준다.

13) nodes라고도 한다.

14) 모양을 규정하기 위한

과 데이터의 세가지 다른 트리구조를 조정하고 저장함으로써 이 기능을 수행한다.

2. 변환과정

조직의 관계데이터베이스를 전자상거래, 기업간(Business-to-Business), 웹응용에서 사용하기 위해 XML파일로 변환하는 것은 XML 변환기능을 내장하고 있는 전문데이터베이스용 프로그램이면 모두 가능하다. 예를들면 SQL Server 2000이나 Access 2002등 데이터베이스 전문프로그램에 XML변환기능들이 지원되고 있다. Xpath, XDR Schema, XSL Transformation, HTTP, OLE DB등과 같은 새로운 규칙을 가지고 XML자료와 관계데이터베이스를 매핑하는데 사용한다. 그리고 XML데이터의 검색, 삽입, 수정도 가능하다. 이들 전문데이터베이스프로그램과 XML을 사용하여 비즈니스데이터를 웹으로 이동하여 종업원, 고객, 파트너등을 위한 종합적 업무자료로 이용하는 것은 아주 유용하다. 이하에서 변환과정의 실행과정을 보면 다음과 같다.

1) XML로의 변환

데이터베이스로부터 관계테이블을 XML데이터로 변환 하는 과정은 SQL Server 2000의 경우 Transact-SQL SELECT문에 XML파일로의 변환 질의어인 FOR XML keywords를 제공하고 동시에 변환시의 포맷형식인 RAW, AUTO, EXPLICIT등을 제공한다. 이들의 기본 문법을 이용한 변환의 예를 보면 다음과 같다.

```
SELECT Name, Photo, Hiredate, Title, Region  
FROM [Employee]  
WHERE Region="WA"  
FOR XML RAW
```

<그림 17 > SQL Server 2000을 이용한 XML변환 질의어의 예

이 질의어의 결과 출력되는 XML파일은 다음과 같으며 루트요소(Root Element)인 <employee>는 추가로 삽입된 것이다.

```
<employee>  
  <row name="Nancy Davolio" photo="EMPID1" hiredate="01-May-1992"  
title="Sales Representative" region="WA"/>  
  <row name="Andrew Fuller" photo="EMPID2" hiredate="14-Aug-1992"  
title="Vice President, Sales" region="WA"/>  
</employee>
```

<그림 18 > XML파일로의 변환 예

Access 2002를 가지고 변환하는 경우는 우선 관계테이블을 열고 File메뉴의 Export서브메뉴에서 Export XML을 선택하면 Access로 작성된 관계테이블이 XML데이터로 변환된다. 예를들면 Employee관계테이블을 변환하면 <별첨 1>과 같이 XML데이터로 변환된다. Save as Type으로 이를 Employees.xml로 저장한다.

2) XSLT파일의 생성

문서편집기를 이용하여 XSLT파일을 만든다. 이때 파일의 확장자는 xsl이 된다. 이 파일은 자료를 원하는 대로 변형할 명령어(instruction)들로 구성하는데 출력하고자 하는 각 요소마다 처리명령을 주던지 요소의 그룹에다 처리 명령을 줄 수 있다. 예를들면 <별첨 2>와 같이 List_Style.xml 과 같이 XSLT파일을 만든다.

3) 브라우저로 보기

Internet Explorer5.0이상에서 <별첨 1>의 Employee.xml파일을 열면 익스플로러의 디폴트 XSLT 스타일시트에 의해 이 데이터를 자동 변환하여 <별첨 3>과 같이 출력한다. 문서편집기를 열고 <별첨 2>의 List_Style.xsl파일을 불러온 후 Employee.xml파일이 있는 동일 위치에 이를 저장한다. 그리고 문서편집기에서 Employees.xml파일을 열고 이 파일의 첫 번째 라인 다음에 아래의 명령을 삽입한다.

```
<?xml-stylesheet type="text/xsl" href="List_Style.xsl">
```

이것은 브라우저가 Employee.xml파일을 열 때 List_Style의 스타일시트를 적용하라는 내용이다. Employee.xml의 변경된 내용을 저장하고 이 파일을 브라우저에서 다시 열면 스타일시트가 적용되어 <별첨 4 >와 같이 브라우저에 나타난다. <별첨 5>는 Employee.xml에 적용될 또 다른 xsl파일로써 만약 이 Catalog_style.xsl과 List_Style.xsl파일을 동시에 적용하려면 <별첨 6>과 같이 VB Script를 이용하면 된다. 즉 똑같은 데이터를 서로 다른 형태로 표현하기 위해서는 XSLT와 VB Script를 동시에 사용하는 응용프로그램이 필요하다.

V. 결 언

E. F. Codd박사가 1970년에 제안한 관계형 데이터 모델의 관계형 데이터베이스는 2차원 테이블 형태이며 반면 XML문서는 불명확한 깊이를 가지는 트리구조이다. 그러므로 XML 관점을 관계형 모델로 혹은 그 반대로 표현하여야 웹에서의 데이터베이스 연산이 가능하다. 따라서 본 연구에서는 관계테이블의 한 예를 들어 XML파일로의 변환과 XSL을 이용한 포맷변경을 실행해 보았다. 실행 결과는 다소 성공적이었으며 계층구조의 XML과 관계데이터베이스간의 매핑구조의 기술적 관계를 이해할 수 있었다. XML은 무선인터넷과 전자상거래에 필수적인 기술로 사용되며 차세대 인터넷 언어로 확고히 자리잡고 있다. 이런 추세는 당분간 지속될 것으로 보이며 여러 분야에 걸쳐 영향을 미칠 전망이다.¹⁵⁾ 그리고 XML을 이용한 웹분야의 응용은 데이터베이스가 기반이 됨을 알 수 있다. 기업간에 전통적인 전자문서교환(EDI)분야, 웹기반의 공급망관리(Supply Chain Management), 기업간(Business-To-Business)전자상거래분야, 웹기반고객관리(Customer Realtion Management), 콘텐츠관리시스템(Content Management System), 기타 전자도서관, eBook등 그 응용 분야는 실로 다양하다 하겠다. 특히 앞으로의 Web통신이 기업간 전자상거래는 물론이고, 일찍

15) 이강찬, "지능형 인터넷 세상을 여는 XML", 마이크로소프트, 2001년 3월, p.196

이 유래가 없던 서로 다른 기종, 운영체제, 미들웨어간의 상호 운영을 무기로 기존의 원격프로시저콜(Remote Procedure Call), CORBA, DCOM, RMI등의 영역까지 넘보고 있으며 이러한 움직임의 시작점에 단순객체접근프로토콜(Simple Object Access Protocol)과 WSDL(Web Service Description Language)이 있으며 이들 미들웨어의 지원을 위한 XML-HTTP, XML-RPC, XML-IDL등이 소개되고 있으므로 향후 XML의 영향력은 클 것으로 사료된다. 하지만 실제로 데이터베이스에 사용되는 효율적 저장(efficient storage), 색인(indexes), 보안(security), 트랜잭션(transaction), 데이터무결성(data integrity), 다중사용자 접속(multi-user access), 트리거(triggers), 다중문서질의(query across multiple documents) 등의 많은 부분들이 부족한 실정이고 응용 소프트웨어를 통해 보강해야 할 부분이다. 때문에 대용량의 데이터나 많은 사용자 접근, 낮은 성능의 데이터 처리 등의 환경에서는 XML 문서를 데이터베이스로 사용하는데 있어 많은 부분을 고려해야 한다.

참고문헌

- 강민호, 김원배, "Q & A로 풀어보는 알쏭달쏭 XML기술", 마이크로소프트웨어 2001년 3월 pp.252-260
- 권준희, 윤용익, "지리정보시스템을 위한 XML" 정보처리학회지 제8권 제3호 pp.61-69
- 기명신, 전윤경, "XML을 활용한 유무선 홈페이지 구축사례", 마이크로소프트웨어 2001년 3월 pp.262-268
- 김달곤, 권병한, "비주얼 C++접근하는 XML프로그래밍", 마이크로소프트웨어, 2001년 3월, pp.224-237
- 김도형, "XML로의 실용적 접근:언어별 XML프로그래밍 I: 자바를 통한 웹 실크로드", 마이크로소프트웨어, 2001년 3월, pp.210-223
- 김석원, "데이터방송을 위한 XHTML", 정보처리학회지 제8권 제3호 pp.83-88
- 손원성, 고승규, 이경호, 김성혁, 임순범, 최윤철, "XML에 기반한 한국전자책 문서표준", 정보처리학회지 제8권 제3호 pp.27-37
- 싸오텍기획마케팅, "XML을 활용한 차세대 전자도서관 구축사례", 마이크로소프트웨어 2001년 3월 pp.269-274
- 양광모, "XML기반 문제은행시스템 구축사례, 마이크로소프트웨어 2001년 3월 pp.280-284
- 양광모, 최성, "XML기반 문제은행 시스템 구현", 정보처리학회지 제8권 제3호, 2001년 5월 pp.89-94
- 유정연, 이규철, "ebXML", 정보처리학회지 제8권 제3호 pp.38-53
- 이강찬, "지능형 인터넷 세상을 여는 XML", 마이크로소프트웨어 2001년 3월, pp.196-209
- 이봉규, "XML기반의 Mobile기술에 관한 연구", 정보처리학회지 제8권 제3호 pp.54-60
- 이승호, "XML기반의 전자카탈로그 구현사례", 정보처리학회지 제8권 제3호 pp.70-75
- 이영석, "XML표현과 변환을 위한 XSL과 XSLT" 마이크로소프트웨어 Jr. 2001년 7월호
- 이영석, "XSLT가 알고싶다", 마이크로소프트웨어 Jr. 2001년 9월 pp.34-41
- 이영석, "XSLT프로세서 활용하기" 마이크로소프트웨어 Jr. 2001년 10월 pp.40-46
- 이용규, "웹에서의 XML문서접근 제어시스템의 설계 및 구현", 정보처리논문지 제7권 제11S호 2000년 11월 pp.3623-3632

임순범, "XSL-FO을 이용한 XML문서의 스타일 정의", 정보처리학회지 제8권 제3호 pp.38-46

임영태, 한우용, 정희경, "XML에 기반한 EDI 문서교환 시스템 설계 및 구현, 정보처리논문지, 제7권 11S호, pp.3603-3612

정석영, 강선미, 정태의, "Voice XML", 정보처리학회지 제8권 제3호 pp.17-26

정승화, 최성호, "Xtream 2000 Editor", 정보처리학회지 제8권 제3호 pp.76-82

정정훈, "XML을 이용한 전자문서 검색시스템 구축사례", 마이크로소프트웨어 2001년 3월 pp.269-274

정채영, 이승호, 배종민, "XML에디터", 정보처리학회지 제8권 제3호, 2001년 5월, pp.10-16

정희경, "차세대 웹문서 표준 XML" 정보처리제6권, 제3호, 1999년 5월, pp.25-35

황병연, 김연혜, "XML스키마 발전동향", 정보처리학회지 제8권 제3호, 2001년 5월, pp.3-9

홍동현, "쉬운파이선으로 더 쉽게 알아본 XML", 마이크로소프트웨어, 2001년 3월, pp.238-250

Brian E. Travis, 「XML and SOAP Programming for BizTalk Servers」, Microsoft, 2000

Claude Seidman, 「Data Mining With Microsoft SQL Server 2000: Technical Reference, IT Professional and Developer」, Microsoft, 2000

David Cox, "Integrating Web Pages With Databases", Dr. Dobb's Journal, September 2000, pp.94-98

Elizabeth Castro, 「XML For The World Wide Web: Visual Quickstart Guide」, Peach Press, 2001

Elliotte Rusty Harold저, 김용권역, 「XML Bible」, IDG Books, 1999

Graeme Malcolm, 「Programming Microsoft SQL Server 2000 With XML」 Microsoft.net Enterprise Server, 2001

Jake Sturn, 「Developing XML Solutions」, Microsoft, 2000

Jump to.net 전국대학투어세미나, 「XML, Visual Studio.NET C#, SQL Server 2000」, 월간마이크로소프트웨어, 테브피아, KCC교육센터, 마이크로소프트, 2001

Michael J. Young, 「Step By Step XML」, Microsoft, 2000

Michael Morrison, 「HTML & XML for Beginners」, Microsoft, 2001

Roy Goldman, Jason McHugh and Jennifer Widom, "Lore: A Database Management System For XML", Dr. Dobb's Journal, April 2000, pp.76-80

William J. Pardi, 「XML In Action Web Technology」, IT Professional, 1999

<http://msdn.microsoft.com/code/default.asp>

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc2k2/html/odc_acc2xslt.asp

<http://schemas.xmlsoap.org/soap/encoding/>

<http://schemas.xmlsoap.org/soap/envelope/>

<http://trio.co.kr/webrefer/xml/xml10.html>

<http://www.oasis-open.org/>

<http://www.microsoft.com/korea/support/default.asp>

<http://www.microsoft.com/mspress/support/search.asp>

<http://www.microsoft.com/mspass/books/5178.asp>
<http://www.schema.net/>
<http://www.w3.org/>
<http://www.w3.org/1999/XMLSchema.dtd>
<http://www.w3.org/1999/XMLSchema.xsd>
<http://www.xml.com/>
<http://www.xmlsoftware.com/>

<별첨 1>

Employee_data.xml Data file 의 보기

```
<?xml version="1.0" encoding="utf-8"?>
<employees>
  <employee>
    <name>Nancy Davolio</name>
    <photo>EMPID1</photo>
    <hiredate>01-May-1992</hiredate>
    <title type="Commission+">Sales Representative</title>
    <region>WA</region>
  </employee>
  <employee>
    <name>Andrew Fuller</name>
    <photo>EMPID2</photo>
    <hiredate>14-Aug-1992</hiredate>
    <title type="Salaried">Vice President, Sales</title>
    <region>WA</region>
  </employee>
  ...

```

<별첨 2>

List_style.xsl XSLT style sheet file의 보기.

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">
<!-- This XSLT style sheet is used to transform the employee_data.xml
file to a simple list like format. It is passed as an argument to the
VBScript transformFile procedure in the default.htm file.
-->
<xsl:output method="html"/>
<xsl:template match="/">
<html>
  <head>
    <title>The XSLT list style sheet</title>
  </head>
  <body>
    <table border="0" cellpadding="10">
      <tr>
        <th align="left">Name</th>
        <th align="left">Region</th>
        <th align="left">Hire Date</th>
        <th align="left">Compensation</th>
        <th align="left">Title</th>
      </tr>
      <xsl:apply-templates select="/employees/employee">
        <xsl:sort select="name"/>
      </xsl:apply-templates>
    </table><p />
  </body>
</html>
</xsl:template>

<!-- Outputs the child elements of the employees element. -->
<xsl:template match="employee">
  <tr>
    <td><xsl:value-of select="name"/></td>
    <td><xsl:value-of select="region"/></td>
    <td><xsl:value-of select="hiredate"/></td>
    <td><xsl:value-of select="title/@type"/></td>
    <td><xsl:value-of select="title"/></td>
  </tr>
</xsl:template>
</xsl:stylesheet>
```

<별첨 3>

인터넷 익스플로러로 출력된 디폴트 XML 파일

```
<?xml version="1.0" encoding="UTF-8" ?>
- <dataroot>
-   <Employees>
      <EmployeeID>1</EmployeeID>
      <LastName>Davolio</LastName>
      <FirstName>Nancy</FirstName>
      <Title>Sales Representative</Title>
      <TitleOfCourtesy>Ms.</TitleOfCourtesy>
      <BirthDate>1968-12-08T00:00:00</BirthDate>
      <HireDate>1992-05-01T00:00:00</HireDate>
      <Address>507 - 20th Ave. E. Apt. 2A</Address>
      <City>Seattle</City>
      <Region>WA</Region>
      <PostalCode>98122</PostalCode>
      <Country>USA</Country>
      <HomePhone>(206) 555-9857</HomePhone>
      ....

```

<별첨 4>

상기 XSLT 스타일시트가 Employees.xml file 에 적용되어 출력된 결과

Name	Address	Home Phone
Steven Buchanan	14 Garrett Hill London, SW1 8JR	(71) 555-4848 Ext 3453
Laura Callahan	4726 - 11th Ave. N.E. Seattle, WA 98105	(206) 555-1189 Ext 2344
Nancy Davolio	507 - 20th Ave. E. Apt. 2A Seattle, WA 98122	(206) 555-9857 Ext 5467
Anne Dodsworth	7 Houndstooth Rd. London, WG2 7LT	(71) 555-4444 Ext 452
Andrew Fuller	908 W. Capital Way Tacoma, WA 98401	(206) 555-9482
Robert King	Edgeham Hollow Winchester Way London, RG1 9SP	(71) 555-5598 Ext 465
Janet Leverling	722 Moss Bay Blvd. Kirkland, WA 98033	(206) 555-3412 Ext 3355
Margaret Peacock	4110 Old Redmond Rd. Redmond, WA 98052	(206) 555-8122 Ext 5176
Michael Suyama	Coventry House Miner Rd. London, EC2 7JR	(71) 555-7773 Ext 428

<별첨 5>

Catalog_style.xml file: Employees.xml에 적용될 또 다른 스타일시트

```
<?xml version="1.0" encoding="utf-8" ?>
...
  <xsl:apply-templates select="/employees/employee">
    <xsl:sort select="name"/>
  </xsl:apply-templates>
...
<xsl:template match="employee">
  <table border="0" cellpadding="10" bgcolor="lightyellow">
    <tr>
      <td>
        
      </td>
      <td>
        <div>
          <b>Name: </b><xsl:value-of select="name"/>
        </div><p/>
        <div>
          <b>Title: </b><xsl:value-of select="title" />
          <br><b>Region: </b><xsl:value-of select="region"/></br>
          <br><b>Compensation:</b>
            <xsl:value-of select="title/@type"/> </br>
        </div><p/>
        <div><font color="GREEN"><b>Northwind Traders</b>
          </font></div>
        </td>
    </tr>
  </table>
</xsl:template>
</xsl:stylesheet>
```

<별첨 6>

Default.htm Web page file: VB Script를 이용하여 상기의 두 style sheet를 적용

```
<html>
<head>
<script language="VBScript">
<!--
Public sub transformFiles(strStylesheetName)
  ' This procedure transforms an XML data file by using an XSLT
  ' style sheet that is passed to it. It then stores the results
  ' of the transformation in a DIV tag on this page.
```

```

'
' Accepts:
'   strStylesheetName   The name of the XSLT style sheet.
'
Dim objResults 'object that holds the HTML results
Dim objXML     'object to hold XML data
Dim objXSL     'object to hold style sheet
On Error Resume Next
'   The DIV element will store the results.
'   set objResults = Document.all("divResults")
'   Create a new document instance for the data file and style sheet.
'   set objXML = CreateObject("MSXML2.DOMDocument.3.0")
'   set objXSL = CreateObject("MSXML2.DOMDocument.3.0")
'   Set the property to validate the file when its parsed.
'   objXML.validateOnParse = true
'   objXSL.validateOnParse = true
'   Load the files
'   objXML.load("employee_data.xml")
'   objXSL.load(strStylesheetName)
'   All must be OK so perform transformation
'   strResult = objXML.transformNode(objXSL)
'   Display the results in the DIV tag
'   objResults.innerHTML = strResult
'   Implement error handling
'   If Err.Number <> 0 Then
'       y = Err.description
'       Response.Write y
'   End if
End Sub
!-->
</script>
</head>
<!-- Call transformFiles procedure to load the simple list view when
the page is opened -->
<body onload="transformFiles('list_style.xml')">
<H2><font color="maroon">The Winning <font
color="green">Northwind</font> Sales Team!</font></H2>
<!-- Create the command buttons that pass the style sheet argument to
the transformFiles procedure. -->
<p>We are proud to present the information on
our team in a &nbsp;<button
onclick="transformFiles('catalog_style.xml')">Catalog</button>&nbsp; 
format or as a &nbsp;<button
onclick="transformFiles('list_style.xml')">Simple List</button>
<hr />
<!-- Insert the results of the XSLT transformation -->
<div id="divResults"></html></div>
</body>
</html>

```

A Case Study on the Web Publishing of Relational DB Via XML

Won-Taek Woo

Abstract

HTML revolutionized the way we specify the appearance of data on the Internet. Today, XML (the eXtensible Markup Language) is changing the way we specify the meaning of data. XML, lets document authors define their own markup tags and attribute names to assign meaning to the data elements in the document. Further, XML elements can be nested and include references to indicate data relationships, as Listing One. Unlike HTML, XML markup tags do not describe how to render the data. Rather, they provide descriptions of data, allowing software to understand the meaning of the data automatically. For publishing, instead, XSL, the eXtensible Stylesheet Language as a separate language , is in charge of specifying the presentation of XML documents. The purpose of this study is to discover how to transform your organizations relational data into potential e-commerce, business-to-business, and web application with XML and XSL documents. For this purpose, the literature survey, first of all, was undertaken to understand the basic structures of XML documents. Second, one case implementation was performed to understand how to transform Access 2002 XML Files into HTML with XSLT and VB script. The results come out to be successful, more or less. But the limitations of it still exist. One immediate limitation is that XML documents are essentially tree structure, as dictated by the nesting of elements. However, relational database tables are two dimensional matrix structure. In addition, real-world data often is graph structured-a single data element may be referenced in multiple ways. However, this study is useful for understanding how to convert relational database into XML documents and to publish them using XSL or VB script.