

# XML 웹 서비스 프로그램 개발을 위한 웹 기반의 XML Tree 생성

박 영 수\* · 장 덕 철\*\*

## 요 약

XML 웹 서비스를 하기 위해서 개발자는 사용자를 위한 클라이언트 프로그램과 서버 프로그램을 함께 개발하여야 한다. 최근 사용자들의 요구와 기대의 증가, 그리고 사회적인 기호와 트렌드 변화 등 내부·외부적인 변화가 자주 발생되고 있는 상황에서, 프로그램의 구조 변경이 자주 발생하게 된다. 따라서 XML을 기반으로 한 웹 프로그램 개발의 시간과 비용을 줄이기 위해 웹 프로그래밍을 위한 자동 생성 도구의 개발이 필요하게 된다. 본 연구에서는 개발 지원 서버의 통해 개발자가 XML 스키마, XML 스타일시트 그리고 XML 문서를 쉽고 빠르게 생성하기 위한 정보를 제공할 수 있는 웹 기반의 XML Tree를 설계하고 구현하는 방법을 제시한다.

키워드 : XML, DOM, XML Tree, XQuery, XPath Query, 관계형데이터베이스

## Web-based XML Tree Builder for XML Service Program Development

Young-Soo Park\* · Duk-Chul Chang\*\*

## ABSTRACT

For the excellent XML web services, the program developers should create both user's client program and the server program. Recently, the structural alteration of programs are often caused by internal and external environment concerned with social trends and preferences as well as user's requirements expectations. So, for reducing the development time and workload of XML web service programs, it is necessary to develop the web program generation tools. This paper suggest how to design and implement the web-based XML Tree builder, which was able to provide the information for promptly and easily generating the XML Schema, XML Stylesheet and XML Document through the XML Development Supporting Server System.

Key Words : XML, DOM(Document Object Model), XML Tree, XQuery, XPath Query, RDBMS

### 1. 서 론

최근 웹 기술과 정보 기술의 발달로 e-Commerce 뿐만 아니라 e-Business를 가능하게 되었고, 사회 전반의 생활 방식에 대한 패러다임을 바꾸어 놓았다[1]. 따라서 점차 고객의 요구가 다양하고 복잡한 형태로 진화하고 있고, 정보를 요구하는 매체 역시 다양한 형태로 개발되고 있다. 또한, 다른 여러 부처들 간에 업무통합이 요구되는 일들이 빈번하게 발생하는 기업에서는, 이를 관리하기 위한 웹 프로그램의 구조 역시 점점 복잡하고 세분화된 형태로 변화되고 있다[1, 2].

그리고 사회 모든 분야의 트렌드(trend) 변화 속도가 점차 빨라짐에 따라 기존의 콘텐츠(Contents)에 대한 생명주기(life cycle)도 짧아지고, 웹 프로그램의 구조 변경 및 유지

보수(maintenance)를 위한 작업은 증가하게 된다[1, 2]. 따라서 기존의 웹 프로그램 개발 방식(asp, jsp, cgi, php 등)을 계속 추구하게 되면, 점차 시간이 흐를수록 처음 계획된 의도와는 전혀 다른 방향으로 변모하게 되고, 프로그램의 전체적인 구조 역시 흔들리게 되어, 더 이상의 유지보수가 어렵게 되는 상황을 초래할 수 있는 문제점을 내포하게 된다.

본 연구에서는 이들 문제점을 최소화할 수 있는 방법 중 하나가 XML이라고 보고, XML 웹 서비스를 위한 프로그램 개발을 보다 쉽게 할 수 있는 방법으로, 웹기반 하에서 XML 스키마[3] 및 스타일시트[4]를 생성하는데 기초가 되는 XML Tree를 설계하고 구현하였으며, 이를 통해 XML 웹 서비스를 위한 프로그램의 자동생성[5, 6]을 하는데 필요한 정보를 제공하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 XML Tree 생성을 위해 필요한 XML 관련 기술 부분에 관해 살펴보고, 3장에서는 XML 서버 구조 및 XML Tree를 생성하기 위한 XML 개발 지원 서버의 구성 방법을 제시한다. 4장에서는

\* 정 회 원 : 광운대학교 대학원 컴퓨터학과

\*\* 정 회 원 : 광운대학교 컴퓨터학과 교수

논문접수 : 2005년 5월 16일, 심사완료 : 2005년 9월 13일

XML Tree를 생성하기 위한 정책을 제시하고 설계한다. 5장에서는 4장에서 설계한 내용을 기반으로 웹 개발자를 위한 XML Tree 생성을 위한 응용을 구현하며, 끝으로 6장에서 결론을 맺는다.

## 2. 관련 연구

본 장에서는 데이터베이스와 웹 서버간의 연동이 용이한 언어인 XML Query와 XML Tree를 구성하기 위해 사용되는 DOM, 그리고 이들 기술들을 이용하게 될 기반이 되는 관계형 데이터베이스에 대해 살펴본다.

### 2.1 XML 관련 기술

W3C의 XML과 관련된 많은 Working Group 중 본 논문과 관련된 기술인 DOM(Document Object Model)과 XML Query(XQuery)를 간단히 살펴보면 다음과 같다.

#### 2.1.1 XML Query (XQuery/XPath Query)

XML Query는 웹상에서 가상 또는 실제의 문서로부터 데이터를 추출하기 위해 질의를 쉽고 편하게 할 수 있도록 제공하기 위한 것으로, XML Query는 웹과 데이터베이스 간에 상호작용을 가능하도록 하며, XML 파일들을 데이터베이스처럼 접근이 가능하도록 한다[7]. 따라서 XML Query(XQuery)는 XML Working Group에서 1998년에 처음으로 웹 문서에 대한 질의를 할 수 있는 국제 표준을 개발하게 된 것이다[8, 9].

XML Query라고 하면 크게 XQuery와 XPath Query로 나눌 수 있는데, XQuery는 일반적으로 관계형 데이터베이스에서 사용하는 질의 형식을 XML문서 내에서 사용할 수 있도록 한 것으로, XML Tree를 생성하기 위해 본 논문에서 제안한 개발 지원 서버를 통해 관계형 데이터베이스로부터 각 테이블 및 컬럼들의 정보를 수집하기 위해 XQuery를 포함한 문서를 동적으로 생성할 수 있도록 하였다. 또한 XPath Query는 XQuery의 축약된 형태로 간단한 경로(path)와 매개변수를 통해 질의를 할 수 있도록 허용함으로써, XPath Query가 포함된 XML문서는 최종 사용자가 질의를 위해 호출할 때 사용할 수 있는 문서를 XML Tree를 기반으로 자동 생성할 때 사용하게 된다.

#### 2.1.2 DOM

XML문서의 요소(element)는 계층구조로 형성되어 있기 때문에, 문서는 모든 정보를 포함하는 트리로 나타낼 수 있다. 서브 요소(sub-element)는 상위 요소의 서브 노드로 나타나며, 속성 또한 속성이 정의된 요소의 서브 노드로 인식된다. 따라서 프로세스 과정에서 XML 파서는 그 문서의 요소들을 나타내는 노드들의 트리를 생성할 수 있다. 이때 사용되는 것이 DOM이다.

DOM(Document Object Model)은 XML의 구조와 정보의 접근을 용이하게 하고, 그것의 조작을 가능하게 하는 표준

인터페이스를 제공하며, XML문서를 계층적인 노드의 형태로 나타낸다. DOM은 1998년 10월 1일에 World Wide Web Consortium에 의해 권고(Recommendation)사항으로 제정되었으며, 크게 3단계로 제안되고 있다[8].

본 논문에서 사용하게 되는 단계는 1단계로서 이 부분은 HTML과 XML 문서 모델 중심으로 문서의 Navigation(특정 부분으로의 접근)과 Manipulation(조작)에 대한 표준 인터페이스 기능을 포함하고 있다.

기존의 연구에서 DOM Tree를 이용하는 주목적이 XML 문서의 계층구조를 Tree형태로 표현함으로써 검색의 효율성을 높이고[10], 스키마의 구조 변경을 용이하게 하며[11], 문서의 가독성 향상과, 유효성 검사를 통한 오류 탐지를 쉽게 하기 위한 것 이라면, 본 논문에서는 개발자가 프로그램을 설계하는 과정에서 사용하게 될 관계형 데이터베이스의 테이블 및 컬럼들 간의 위치 관계에 관한 정보 및 각 컬럼이 갖고 있는 속성 정보들을 보관하기 위한 목적으로 DOM Tree를 사용하게 된다.

### 2.2 XML을 위한 RDBMS

최근 대부분의 RDBMS에서는 XML을 지원하는 방향으로 발전하고 있다. XML 문서를 저장하고 검색하여, 계층적 구조를 갖는 질의에 응답이 가능하도록 설계되고 있고, XML 기반의 질의어 및 프로그램이 가능한 포맷을 지원함으로써, XML 기반의 웹 프로그램 개발에 힘을 실어주고 있다[9, 12, 13]. 그러나 (그림 3)의 (a)와 같은 관계형 데이터베이스 테이블에서 테이블 A,B,C를 XQuery를 이용하여 조인하는 질의를 하고, For XML문으로 결과를 반환하였을 때, (b)와 같은 구조를 갖는 것이 아니라 (c)와 같은 결과를 얻게 된다. 이것은 현재의 관계형 데이터베이스가 XML이 갖는 계층적 구조를 충분히 지원하지 못하는 결과를 단적으로 보여주는 것이고, 또한 웹 문서에 포함된 질의에 대한 다른 데이터베이스들과의 호환성 문제, 그리고 데이터 조인을 통한 질의를 할 경우 데이터베이스에 의존적인 형태로 제공됨으로 인하여, 개발자 측면에서 보면, 보다 자유로운 계층구조를 설계하는데 장애 요소를 갖고 있다. 따라서 최근 이를 해결하기 위한 방법으로 XML 스키마를 이용하여 데이터베이스 스키마를 통합하는 연구가 시도되기도 하였다[14, 15]. 그러나 본 연구에서는 관계형 데이터베이스로부터 얻을 수 있는 정보와 XML문서를 생성하는데 필요한 정보를 개발 지원 서버의 도움을 받아 쉽게 개발자가 원하는 형태의 XML Tree구조를 웹상에서 생성할 수 있도록 하고, 이렇게 생성된 XML Tree는 XML 웹 서비스를 위한 프로그램을 자동생성 하는데 필요한 정보를 제공하는데 목적을 두고 있다.

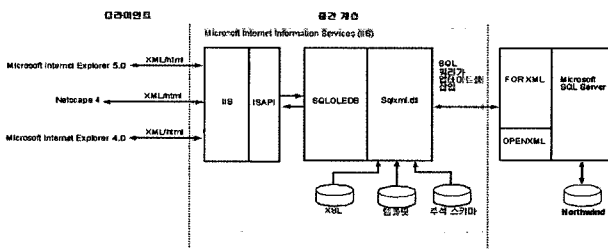
## 3. XML Server구조와 XML 개발 지원 서버 구조

본 장에서는 XML 서버 구조와 XML 웹 서비스를 위한 처리과정을 살펴보고, 본 논문에서 제안하는 웹 서버와 개

발자 시스템 사이에 존재하는 웹 기반의 XML 개발 지원 서버를 통해 개발자는 XML Tree를 생성하게 된다.

3.1 XML Server 구조

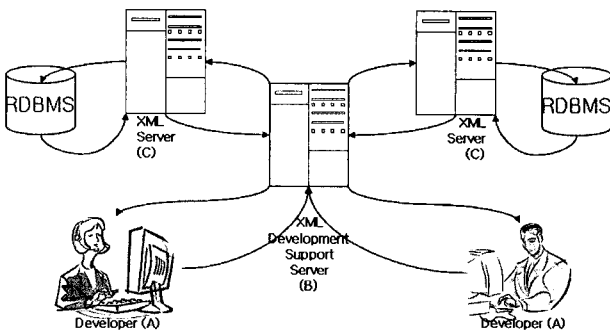
본 논문에서 사용할 XML 서버 구조는 (그림 1)과 같이 3계층 구조를 기본 구조로 갖는다. 우선 클라이언트에서 서비스 요청을 하면, IIS는 이 요청을 sqlisapi.dll에 전달하고, ISAPI는 SQLOLEDB와 통신을 하게 된다. 실제 XML에 대한 처리는 sqlxml.dll에서 담당하며, FOR XML을 사용하여 데이터베이스에서 데이터를 가져오거나 OPENXML을 사용하여 XML 데이터를 데이터베이스에 저장하게 된다[16].



(그림 1) XML 서버 구조

3.2 웹기반 XML 개발 지원 서버 구조

(그림 2)에서 개발자(A)는 자신이 관리하는 서버(C)에서 서비스해야 할 서버 프로그램을 생성하기 위해, 우선 개발 지원 서버(B)에 접속한다. 그리고 자신이 관리하는 XML 기반의 웹 서버(C)에 설정된 가상 디렉토리(template)와 데이터베이스 접속을 위한 password를 입력한다. 다음은 개발 지원 서버(B)가 요구하는 항목에 대해 개발자는 단계적(step by step)으로 선택 또는 설정을 함으로서 서버(C)에서 수행하게 될 XML 웹 프로그램에 필요한 정보를 수집하게 된다. 개발 지원 서버(B)는 수집된 정보들을 저장한 후, 이를 이용하여 XML 문서를 생성하기 위해 필요한 XML Tree를 생성한다. 이렇게 생성된 XML Tree는 XML 스키마 및 XML 스타일시트 생성을 위해 사용하게 된다.



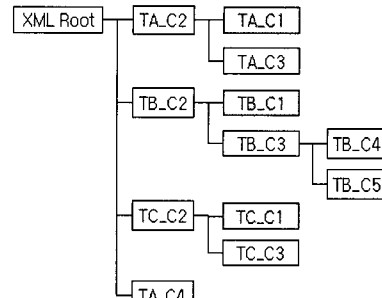
(그림 2) 개발자를 위한 웹 기반의 XML 개발 지원 서버 구성도

4. XML Tree 설계

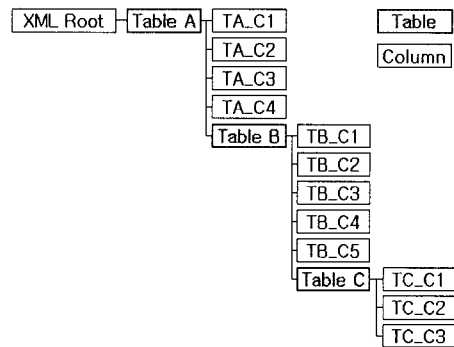
본 논문에서 생성하고자 하는 XML Tree는 웹 기반에서

Table A	TA_C1	TA_C2	TA_C3	TA_C4	Key	
Table B	TB_C1	TB_C2	TB_C3	TB_C4		TB_C5
Table C	TC_C1	TC_C2	TC_C3			

(a) 데이터베이스 테이블 구조



(b) XML 트리 구조



(c) RDBMS의 XML 지원 구조

(그림 3) RDBMS와 XML Tree의 관계

개발자가 개발 지원 서버의 도움을 받아 관계형 데이터베이스로부터 필요한 정보를 얻고, XML문서를 생성하는데 필요한 정보들을 결정하거나 설정할 수 있도록 설계하였다. 따라서 (그림 3)의 (a)인 RDBMS의 테이블 및 컬럼 구조를 (b)인 XML Tree 형태로 표현하기 위해서는 그에 합당한 정책이 수립되어야 한다.

그리고, 관계형 데이터베이스 테이블의 컬럼들 중에서 주 키 또는 외래키에 해당하는 컬럼들은 요소의 서브 노드들 중 속성으로만 사용되고, 그 외의 모든 컬럼들은 요소로만 사용된다는 조건에 부합되어야 한다. 또한, XML Tree를 구성할 때, 1레벨은 모두 요소로만 구성되고, 트리에서의 계층구조는 HTML문서로 변환되어 서비스될 때, 상위 노드와 하위 노드의 관계는 링크로 이루어진다는 전제조건하에서 시작한다.

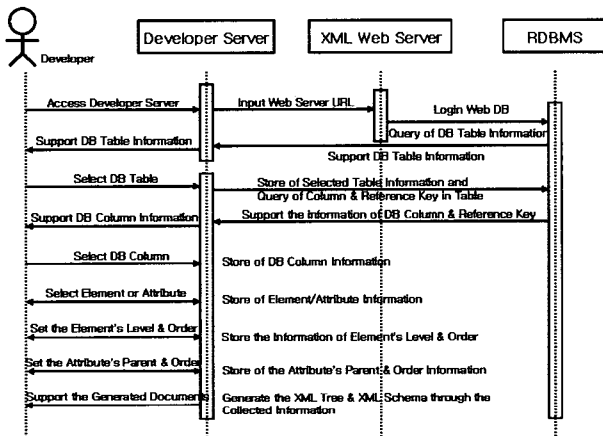
4.1 XML Tree 생성을 위한 정책 제시

본 논문에서는 XML Tree 생성에 대한 정보 수집을 위해서 <표 1>와 같이 7단계의 정책을 제시하였다.

위의 정책은 개발 지원 서버가 웹 프로그램에 필요한 정보를 수집하고, 필요에 따라 선택 또는 설정해야 할 사항들을 개발자에게 제시하며, 그 결과를 웹 서비스를 위한 프로그램 자동생성에 필요한 정보를 제공할 수 있도록 웹상에서 XML Tree형태로 저장하기 위해 본 논문에서 제안한 정책이다.

<표 1> 계층구조를 갖는 XML Tree 생성을 위한 정책

단 계	내 용
1단계	개발지원 서버 접속 및 DB 접속 인증 단계
2단계	RDBMS로부터 사용 가능한 테이블 선택 단계
2.1	사용할 테이블 선택
2.2	선택된 테이블간의 참조 키 선택
3단계	선택된 테이블내의 Column 선택 단계
3.1	사용할 Column 선택
4단계	선택된 Column의 편집 단계
4.1	선택된 Column에 대한 Element or Attribute 결정
5단계	Element의 세부 설정 단계(키 값으로 사용되지 않는 열)
5.1	상위 레벨 요소 결정
5.2	Element들 간의 순서 결정
5.3	자식 Element들의 부모 노드 및 순서 결정
6단계	Attribute의 세부 설정 단계(키 값으로 사용되는 열)
6.1	Attribute에 대한 소속 Element 및 순서 결정
7단계	각 Column들의 Alias 및 XML 데이터 타입 설정 단계



(그림 4) XML Tree 생성을 위한 시퀀스 다이어그램

4.2 XML Tree 시퀀스 다이어그램

<표 1>에서 제시한 정책을 기반으로 XML Tree 생성을 위해, XML 질의어를 통한 정보 수집 과정을 시퀀스 다이어그램으로 표현하면 (그림 4)와 같다.

<표 1>과 (그림 4)를 통해 수집된 정보는 데이터베이스의 테이블 내에서 선택된 각각의 컬럼(노드)들이 요소 또는 속성 인지에 대한 정보와, 트리 내에서 사용될 각 노드들의 레벨

및 동일 레벨간의 순서, 그리고 부모노드에 대한 정보 및 최상위 노드로부터의 각 노드들 간의 경로 등을 갖게 된다.

<표 2>의 XML Tree Design에서 Node Name에 사용된 8자리의 문자 코드 중 처음 3자리(L01)는 노드의 레벨을 나타내고, 다음 2자리(NE/NA)는 요소 또는 속성인지를 나타내며, 마지막 3자리(O01)는 레벨의 순서를 나타낸다. 그리고 Node Full Path를 통해서 현재 노드의 부모노드에 대한 정보를 얻을 수 있도록 설계하였다. 수집된 정보를 위와 같이 구성하고, Node Full Path의 값을 기준으로 오름차순 정렬을 하면 원하는 XML 트리 구조를 구성할 수 있는 순서를 갖게 된다.

5. XML Tree의 구현

5.1 XML Tree 생성을 위한 정보 수집 단계

개발자가 개발 지원 웹 서버를 통해 자신이 관리하는 서버 및 데이터베이스를 이용할 수 있게 하기 위해 개발자가 원하는 정보를 대신 질의할 수 있어야 한다. 따라서 <표 1>에서 제시한 정책에 따라 각 단계마다 XQuery를 활용하여 XML Tree를 구현하였다.

아래 XQuery의 FROM절에서 사용된 Information\_Schema를 통해, MS-SQL 서버에 있는 Table, Column, 그리고 Reference Key에 대한 정보를 얻을 수 있고, FOR XML문을 통해 XML 문서 형태로 반환할 수 있도록 하였다 [7, 9, 16].

```
<?xml version='1.0' encoding="EUC-KR"?>
<?xml-stylesheet type='text/xsl' href='xsl.xsl'?>
<root xmlns:sql='urn:schema-microsoft:com:xml-sql'>
<sql:query>
SELECT Table_Name, Column_Name, Ordinal_Position,
Constraint_Name, Data_Type
FROM Information_Schema.
{Tables|Columns|Key_Column_Usage}
WHERE Table_Name='Table Name'
AND Ordinal_Position in (Ordinal Positions)
{OR Join Table & Ordinal Positions}
FOR XML Auto, Element
</sql:query>
</root>
```

<표 1>에서 제시한 정책들을 개발 지원 서버에 의해 각 단계별로 동적으로 자동 생성되는 XQuery문들은 다음과 같

<표 2> XML Tree Design

Node Name	타입	Table & Column 이름	부모노드	Full Path Node Code	Column Ordinal Position	DB Data Type	Null 허용	키값	노드 Alias	XML Data Type	최소
L01NEO01	E	XMLFreeBoard.Title	XMLRoot	L01O01NE	XMLFreeBoard.2	nvarchar	No		XMLFreeBoard. 제목	string	1
L02NAO01	A	XMLFreeBoard.UserID	XMLFreeBoard.Title	L01O01NE/L02O01NA	XMLFreeBoard.1	int	No	PK	XMLFreeBoard.FreeUserID	id	1
L02NEO02	E	XMLFreeBoard.Content	XMLFreeBoard.Title	L01O01NE/L02O02NE	XMLFreeBoard.3	nvarchar	Yes		XMLFreeBoard. 내용	string	0
L02NEO01	E	XMLFreeBoard.WriteDay	XMLFreeBoard.Title	L01O01NE/L02O01NE	XMLFreeBoard.4	datetime	No		XMLFreeBoard. 작성일	date	1

은 형태로 생성된다.

### 1단계 :

개발자 : 개발 지원 서버에 접속하여 XML 웹 서비스를 할 수 있도록 설정된 가상 디렉토리의 경로를 입력한다.

### 2단계 :

[2-1]

서버 : 이 경로를 통해 DB에 있는 테이블들을 검색하여 개발자에게 보여주고 사용할 DB의 테이블을 선택할 수 있도록 제공한다.

```
SELECT      Table_Name
FROM        Information_Schema.Tables
WHERE      Table_Type='Base Table'
FOR XML Auto, Elements
```

개발자 : 사용할 테이블을 선택한다.

[2-2]

서버 : 개발자가 선택된 테이블을 보여주고, 선택된 테이블에 존재하는 주키 및 외래키를 검색하여, 사용할 키들을 선택할 수 있도록 한다.

```
SELECT Table_Name, Column_Name, Constraint_Name
FROM Information_Schema.Key_Column_Usage
WHERE Table_Name='Table1' (OR
      Table_Name='Table2') //선택된 테이블만큼 반복
FOR XML Auto, Elements
```

개발자 : 선택된 테이블 내에 있는 참조키 들을 확인한다.

서버 : 주키, 외래키, 주키와 외래키의 관계, 테이블간의 관계, 메인키(최상위 테이블의 주키)를 개발자가 확인할 수 있도록 보여준다.

### 3단계 :

서버 : 선택된 테이블 내에 있는 모든 Column들을 나열한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name
FROM Information_Schema.Columns
WHERE Table_Name='Table1' (OR
      Table_Name='Table2') //선택된 테이블만큼 반복
FOR XML Auto, Elements
```

개발자 : 사용할 Column들을 선택한다.

### 4단계 :

서버 : 개발자가 선택한 Column들을 나열하고, 요소(Element)로 사용될 Column들을 선택하도록 한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name
FROM Information_Schema.Columns
```

```
WHERE Table_Name='Table1' AND
      Ordinal_Position in (1,2,3) (OR
      Table_Name='Table2' AND
      Ordinal_Position in (1,2,3,4,5,6))//테이블만큼 반복
FOR XML Auto, Elements
```

개발자 : 요소(Element)로 사용할 Column들을 선택한다.

### 5단계 :

[5-1]

서버 : 개발자가 선택한 요소들을 나열하고, 최상위 레벨(Level 1)로 사용될 Column들을 선택하도록 한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name
FROM Information_Schema.Columns
WHERE Table_Name='Table1' AND
      Ordinal_Position in (2,3) (OR
      Table_Name='Table2' AND
      Ordinal_Position in (2,3,4,5,6)) //테이블만큼 반복
FOR XML Auto, Elements
```

개발자 : 최상위 레벨로 사용될 Column들을 선택한다.

[5-2]

서버 : 선택된 최상위 레벨의 Column을 나열하고, 동일 레벨에서의 순서를 결정하도록 한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name
FROM Information_Schema.Columns
WHERE Table_Name='Table1' AND
      Ordinal_Position in (2) (OR
      Table_Name='Table2' AND
      Ordinal_Position in (2)) //선택된 테이블만큼 반복
FOR XML Auto, Elements
```

개발자 : 최상위 레벨의 Column들에 대한 순서를 결정한다.

[5-3]

서버 : 상위 레벨을 제외한 Column들을 나열하고, 각 Column들의 부모 노드와 동일 레벨에서의 순서를 결정하도록 한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name
FROM Information_Schema.Columns
WHERE Table_Name='Table1' AND
      Ordinal_Position in (3) (OR
      Table_Name='Table2' AND
      Ordinal_Position in (3,4))//선택된 테이블만큼 반복
FOR XML Auto, Elements
```

개발자 : 부모노드와 동일 레벨에서의 순서를 결정한다.

서버 : 개발자가 부모노드를 결정하지 않은 Column들은 최상위 노드를 제외한 모든 요소들이 부모노드를 가질 때까지 [5-3]을 반복한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name
FROM Information_Schema.Columns
WHERE Table_Name='Table2' AND
      Ordinal_Position in (5,6)
FOR XML Auto, Elements
```

**6단계 :**

서버 : 요소(Element)들을 제외한 속성(Attribute)들을 나열하고, 개발자가 속성들의 부모노드(요소)를 결정하고, 속성들의 순서를 결정할 수 있도록 한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name,
FROM Information_Schema.Columns
WHERE Table_Name='Table1' AND
      Ordinal_Position in (1) {OR
      Table_Name='Table2' AND
      Ordinal_Position in(1) } //선택된 테이블만큼 반복
FOR XML Auto, Elements
```

개발자 : 각 속성들의 부모노드와 순서를 결정한다.

**7단계 :**

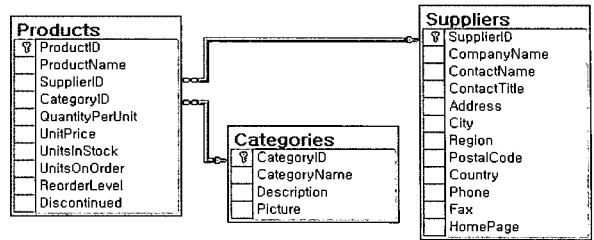
서버 : 각 속성 및 요소노드들을 나열하고, 각 노드들의 별칭(alias) 및 XML Data Type을 설정할 수 있도록 한다.

```
SELECT Table_Name, Ordinal_Position, Column_Name,
      Data_Type, Is_Nullable // for Attributes
FROM Information_Schema.Columns
WHERE Table_Name='Table1' AND
      Ordinal_Position in (1) {OR
      Table_Name='Table2' AND
      Ordinal_Position in(1) } //선택된 테이블만큼 반복
FOR XML Auto, Elements // for Attribute List
SELECT Table_Name, Ordinal_Position, Column_Name,
      Data_Type, Is_Nullable // for Elements
FROM Information_Schema.Columns
WHERE Table_Name='Table1' AND
      Ordinal_Position in (2,3) {OR
      Table_Name='Table2' AND
      Ordinal_Position in(2,3,4,5,6) } // 테이블만큼 반복
FOR XML Auto, Elements //for Element List
```

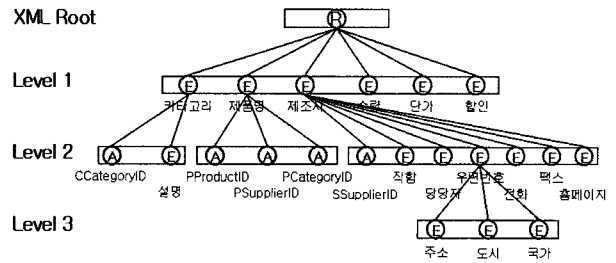
개발자 : 나열된 각 노드들의 별칭을 정하고, 데이터베이스에서 정의된 데이터 타입을 참조하여 XML이 지원하는 데이터 타입에 맞게 설정한다.

**5.2 XML Tree 구현**

본 논문에서의 구현은 앞의 4장에서 제시한 정책과 설계를 기반으로 개발자는 개발 지원 서버가 요구하는 사항들에 대해 선택하고 결정한 사항들에 대한 정보를 이용하여 MS사의 XML DOM 객체를 통해 해당 요소 및 속성들의 노드들에 대한 계층구조를 생성하였다. XML Tree 생성을 위해 javascript를 이용하여 해당 노드의 계층 구조를 생성할 수 있도록 하였다[9, 17]. 따라서 개발자는 본인이 선택하고 결



(그림 5) 데이터베이스 테이블 다이어그램



(그림 6) XML Tree의 계층구조

정한 내용을 기반으로 생성된 형태의 소스를 웹상에서 얻을 수 있도록 하였다.

본 논문의 구현 예로 사용한 데이터베이스 테이블간의 관계를 다이어그램으로 표현하면 (그림 5)와 같다.

XML Tree의 계층 구조는 개발자가 개발 지원 서버의 도움을 받아 구성하게 되는데, XML Root를 루트 노드로 하여 1레벨은 요소들로만 구성된다[9, 17]. 요소로 사용될 수 있는 항목은 각 테이블내의 컬럼들 중 외래키 또는 주키를 갖지 않는 컬럼에 해당되며, 속성항목들은 주키 또는 외래키 값으로 설정된 컬럼에 한하여 설정할 수 있게 하였다.

XML Tree의 계층 구조는 최종 웹 서비스가 시작되는 화면 구성을 토대로 설정하게 되는데, 1레벨은 첫 화면에 출력될 항목들, 2레벨은 첫 화면에서 출력된 항목 중 하나를 클릭할 경우 출력될 항목들로서, (그림 6)의 XML Tree의 계층구조와 같은 형태로 구성된다.

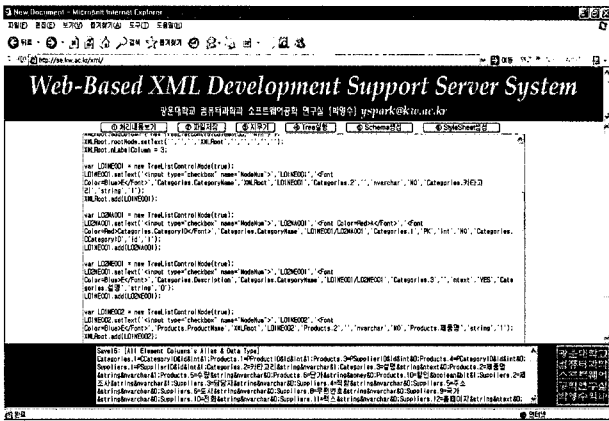
(그림 7)의 (a)는 javascript를 기반으로 생성된 결과 화면이다. 또한 (b)는 해당 소스를 이용하여 웹상에서 실행한 웹기반의 XML Tree에 실행 화면이다.

(그림 7)의 (b)에 나타난 정보가 XML 문서를 생성하기 위해 사용되는 부분을 요약하면 <표 3>과 같다.

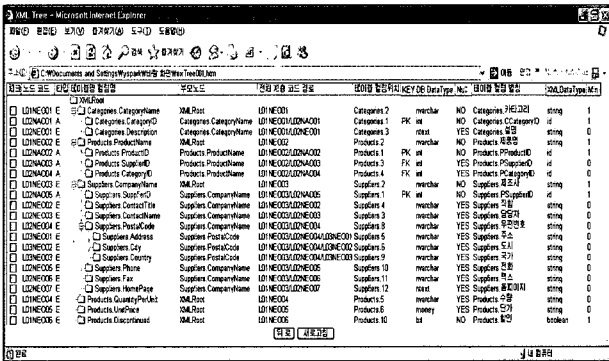
<표 3>에서 노드코드와 전체 계층 코드 경로는 XML Tree를 구성하기 위해 생성된 것이며, 테이블/컬럼명과 테이블내의 컬럼 위치 그리고 KEY는 데이터베이스 테이블 생성시 저장된 스키마 정보를 이용한 것이며, 컬럼 별칭 및 XML Data Type은 개발자가 데이터베이스 정보를 참조하여 설정한 값들이다. 이 값들은 단지 트리를 구성하기 위한 정보뿐만 아니라, 스키마와 스타일시트 그리고 Xpath를 포함하는 XML문서를 생성하는데 필요한 정보들을 포함하게 된다. 따라서 위의 정보들을 이용하면, XML 웹 서비스에 필요한 문서들을 자동 생성할 수 있는 기초 자료를 제공할 수 있도록 설계하고 구현하였다.

<표 3> XML Tree 구성 정보의 적용 부분

항목	내용	설명	사용목적
노드코드	L02NAO01	내부 노드코드명	Tree
타입	E or A	속성/요소	Schema
테이블 & 컬럼명	Categories, CategoryID	컬럼 이름 및 테이블 이름	Schema
부모노드	Categories, CategoryName	부모노드 이름	Schema
전체계층 코드경로	N01NEO01/N02NAO01	XML Tree 구성을 위한 경로정보	Tree
컬럼 위치	Categories.1	테이블내의 컬럼 위치정보	Tree
키	PK or FK	주키/외래키정보	Schema
DB Data Type	int, nvarchar	DB내 컬럼 데이터 타입	for XML Data Type
Nullable	No	null허용 여부	for Min
테이블 컬럼 별칭	Categories, 카테고리	컬럼 별칭(alias)	Schema, Stylesheet
XML Data Type	ID, string	XML 데이터타입	Schema
Min	1	최소 사용 여부	Schema



(a) Java Script 소스



(b) XML Tree 실행 화면

(그림 7) XML Tree 소스 및 실행 화면

5.3 기존 연구와의 비교 및 응용 예

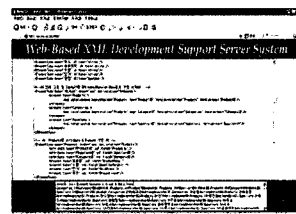
본 논문에서 생성한 XML Tree는 DOM Tree[8]가 XML 문서의 구조를 표현하기 위한 단계를 넘어 <표 3>과 같이 트리를 구성하는 노드들의 정보 및 노드들 간의 관계, 그리고 웹 서비스를 위한 프로그램 생성에 필요한 정보들을 저장하고 있다. 이것은 이전의 저작도구[18]들이 주로 로컬 컴퓨터상에서 응용프로그램 형태로 실행되고 있지만, 본 논문에서 구현한 XML Tree는 웹상에서 실행할 수 있도록 설계하고 구현 하였다.

또한, DOM Tree[8]가 갖는 고유의 특징인 XML문서의 가독성 향상 및 유효성 검사를 통한 오류탐지 목적 이외에 대부분의 연구들이 주로 효율적인 검색[10]과 접근제어[11] 및 저장[19] 등에 초점을 맞추고 있다면, 본 연구에서는 웹 서비스에 필요한 XML 스키마, 스타일시트, 그리고 문서를 생성하기 위한 정보를 웹상에서 수집하여 저장하고, 그 정보를 XML 개발 지원 서버를 통해 제공할 수 있도록 한 것이 기존 연구들과의 가장 큰 차이점이라고 할 수 있다.

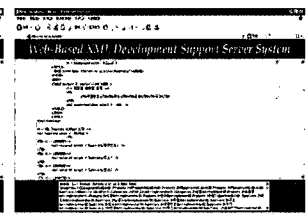
따라서 개발자는 특정한 개발도구의 도움 없이 인터넷이 가능한 곳이라면 언제 어디서나 쉽고 편리하게 XML 개발 지원 서버를 통한 아웃소싱 형태로 웹 프로그램 개발에 필요한 정보들을 수집하고 이를 활용할 수 있도록 한 것이 본 논문의 가장 큰 특징이라고 할 수 있다. 본 연구가 갖는 특징들은 비교하면 <표 4>와 같다.

<표 4> 본 연구와 기존 연구와의 비교분석

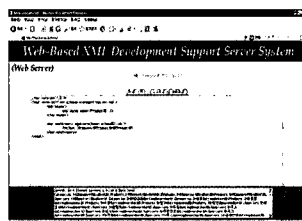
평가항목	기존 연구	본 연구
생성 기반	응용프로그램 기반	웹 기반
XML Tree 사용 목적	가독성향상/오류탐지/검색 향상/구조변경	XML 프로그램 생성에 필요한 정보 제공
문서 자동 생성 지원	스카마->문서 생성 문서->스카마 추출	XML 스키마, 스타일시트, 문서 자동 생성 지원
중간 오류 발견 시점	최종 테스트과정에서 발견	프로그램 생성과정에서 발견
유지보수/재사용성	가능(보통)/재작성이 편리함	가능(우수)/XML Tree의 재구성으로 간편함
프로그램 설치장소	데스크톱 /노트북	개발 지원 서버
개발 방법	로컬 컴퓨터를 통한 응용프로그램 중심	개발 지원 서버를 통한 아웃소싱 기법 적용



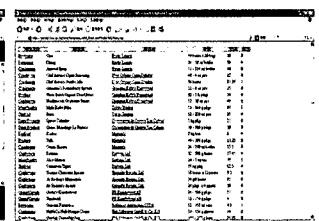
(a) XML스키마 생성화면



(b) XML스타일시트 생성화면



(c) XML문서 생성화면



(d) XML문서 실행화면

(그림 8) XML 웹서비스를 위한 원시코드 생성과 웹 서비스 실행 화면

(그림 8)은 XML Tree의 정보를 이용하여 웹상에서 개발자가 개발 지원 서버를 통해 XML 스키마, 스타일시트 및 문서를 생성하는 화면과, 이를 통해 생성된 문서를 실행한 화면을 보여주고 있다.

## 6. 결론

본 논문에서는 개발자가 XML 개발 지원 서버를 통해 빠른 시간 내에 관계형 데이터베이스로부터 개발자가 원하는 형태의 XML Tree를 생성하고, 이를 기반으로 XML 스키마 생성뿐만 아니라 XML 스타일시트를 쉽게 생성하기 위한 정보를 제공하는데 목적을 두고 있다. 이렇게 됨으로써 얻게 되는 장점으로는 사용자에게는 다양한 형태의 서비스가 가능하고, 개발자 역시 프로그램 개발 시간의 단축 및 XML Tree의 재구성을 통해 유지 보수를 보다 쉽고 편리하게 할 수 있고, 급격한 트렌드 변화에 유연하게 대처할 수 있을 것으로 기대된다.

향후 연구과제로는 XML Tree를 기반으로 하여, XML 스키마 및 XML 스타일시트를 자동 생성하는 것에 대한 개발과 연구가 필요하다고 본다.

## 참고 문헌

[1] Ravi kalakota, Marcia Robinson, "e-Business 2.0", Addison-Wesley, 2001  
 [2] Danny Ayers et al, "Professional Java Server Programming", Wrox Press, 1999  
 [3] Jon Duckett et al, "Professional XML Schemas", Wrox Press, 2001  
 [4] Neil Bradley, "The XSL Companion", Addison Wesley, 2000  
 [5] Kevin Dick, "XML A Manager's Guide" Second Edition, Addison Wesley, 2002  
 [6] J. Craig Cleaveland, "Program Generators with XML and Java", PH PTR, 2001  
 [7] Kevin Williams et al, "Professional XML Databases", Wrox Press, 2001  
 [8] W3C online, [http://www.w3.org/\(XML/DOM/XML Schema/XML Query\)](http://www.w3.org/(XML/DOM/XML Schema/XML Query))  
 [9] msdn online(XML), <http://msdn.microsoft.com/xml>  
 [10] 김영란, "효율적인 검색을 위한 Tree형태의 XML 구조 모델", 한국컴퓨터정보학회 논문지, 9권 제4호, 2004.  
 [11] 전재명 외 3인, "관계형 모델 기반의 XML 데이터에 대한 접근 제어 기법", 데이터베이스연구회KDBC(SIGDB-KISS), pp.64-71, 2003. 5.

[12] Ronald Bourret, "XML and Databases", <http://www.rpbouttet.com/xml/XMLAndDatabases.htm>, Last updated July, 2004.  
 [13] Ronald Bourret, "XML Databases Products", <http://www.rpbourret.com/xml/XMLDatabase Prods.htm>, Last updated Sep., 2004.  
 [14] H. Schoning, "Tamino-a DBMS Designed for XML" in Processing of IEEE ICDE, 2001.  
 [15] U-Chang Park, "A Database Schema Integration Method using XML Schema", Korean Society for Internet Information Vol.3-2, Apr., 2002.  
 [16] Deitel et al, "XML How to Program", Prentice Hall, 2000  
 [17] Nathan Ridley, "Jscript Tree List Control", <http://www.codeproject.com/jscript>, Dec., 2002.  
 [18] Altova, XMLSpy2005, <http://www.altova.com/manual2005/XMLSpy/SpyEnterprise>, 2005.  
 [19] 오동일 외 3인, "XDOM: 확장성 기반의 경량 XML 객체 정보 저장소", 정보과학회 논문지, 9권 3호, pp.332-340, 2003.



박 영 수

e-mail : yspark@kw.ac.kr

1996년 광운대학교 전산대학원 전자계산학과(이학석사)

1999년 광운대학교 대학원 컴퓨터과학과 수료

1998년~현재 경북대학 겸임교수

관심분야 : XML, 분산처리, 객체지향프로그래밍, 소프트웨어공학, 무선인터넷, 모바일컴퓨팅 등



장 덕 철

e-mail : dcjang@kw.ac.kr

1982년 고려대학교 대학원 경영정보학(박사)

1981년~1982년 버클리 대학교 객원 교수

1993년 광운대학교 전산사회교육원 원장

1996년 광운대학교 전산대학 원장

1997년 광운대학교 이과대학 학장

1997년~현재 광운대학교 컴퓨터과학과 교수

관심분야 : 소프트웨어공학, 버전 제어, 컴포넌트, 객체지향설계 방법론 등