

# 웹 서버 상에서 XML 트리를 이용한 XML 스키마와 스타일시트 생성

## (XML Schema and Stylesheet Builder using XML Tree on Web Server)

박 영 수 <sup>†</sup>      장 덕 철 <sup>\*\*</sup>  
(Young-Soo Park)      (Duk-Chul Jang)

**요 약** 최근 다양한 형태의 웹 서비스가 요구되는 상황에서, XML은 기존 다른 언어에 비해 프로그램 작성 및 관리가 쉽고, XML 관련 기술들에 대한 연구가 활발히 진행되고 있어, 개발자 측면에서는 많은 장점을 갖는다. XML문서는 데이터뿐만 아니라 문서를 구성하기 위한 구조를 포함하고 있기 때문에, XML 문서를 생성하기 위해서는 우선적으로 데이터 구조를 나타내는 DTD(Document Type Definition) 나 XML 스키마에 대한 정의가 선행되어야 한다. 본 연구에서는 웹 기반에서 생성된 XML 트리를 이용하여 XML 스키마와 XML 스타일시트를 자동 생성할 수 있는 방법을 제시하고 구현하였다.

키워드 : XML, XML 트리, XML 스키마, XML 스타일시트

**Abstract** At the present day, increasing the web service requirements, along with making XML program easy to create and manage, XML program which progress the research of related technologies enables developer the number of benefits. In order to create XML Documents which include structure that compose documents as well as data, it must have defined about DTD or XML schema that display structure of data. In this study, herewith suggest the design and implementation of method that can be automatically created XML Schema and XML Stylesheet using the web-based XML Tree.

**Key words** : XML, XML Tree, XML Schema, XML Stylesheet

### 1. 서 론

최근 W3C를 중심으로 XML을 비롯한 XML과 관련된 표준들의 지속적인 연구 개발 결과 cgi, asp, jsp, php 등 기존의 웹 서버에서 사용하던 응용 프로그램을 대체할 수 있을 만큼 비약적인 발전을 이루었다. 또한, 관계형 데이터베이스 역시 XML을 지원하는 방향으로 발전하고 있다[1-4]. 따라서 서버 측의 다른 특별한 응용 프로그램이 없이도 XML과 XML관련 기술들만으로 충분히 기존의 웹 서버가 수행하던 작업을 대체할 수 있을 만큼 발전되고 있다[5].

일반적으로 XML문서의 생성 유형을 살펴보면 다음과 같이 3가지 유형으로 분류할 수 있다. 첫째, 직접 또는 일반 XML문서 편집기를 이용하여 작성하는 방법으로 기존의 HTML문서 작성 방법과 비슷하며, 이 경우

사용자의 동적인 요구에 대응하기 어렵고, 간단하고 정적인 문서를 작성하는데 주로 사용하는 방법이다. 둘째, XML 질의를 이용하여 XML문서를 생성하는 방법으로, 질의어 자체가 기존의 관계형 데이터베이스에서 사용하던 질의어 형태를 취하고 있어, 동적으로 XML 문서의 생성이 가능하고, 따라서 사용자의 동적인 요구에 대응하기 쉬운 장점을 가지고 있다[1,6-8]. 그러나 이 유형은 지나치게 데이터베이스에 의존적이고, 자체의 구조를 지속적으로 유지하고 관리하는데 어려운 단점을 가지고 있다[9]. 셋째, XML 스키마를 기반으로 XML 문서를 생성하는 방법이다. 과거 DTD가 가지고 있던 복잡한 구조를 갖는 문서에 대한 정확한 정의를 내리기 어려운 문제점과, 제한된 데이터 타입을 사용해야 하는 단점을 보완한 XML 스키마는 계층적 구조를 갖는 XML문서의 구조를 정의하고 있기 때문에, XML 문서에 대한 계층구조를 항상 유지할 수 있을 뿐만 아니라 관리 및 유지보수가 쉽고 사용자의 동적인 요구에 대응이 가능한 장점을 가지고 있다[1,10].

따라서 본 논문에서는 세 번째 유형을 적용하여 웹

<sup>†</sup> 정 회 원 : 광운대학교 컴퓨터과학과  
yspark@kw.ac.kr

<sup>\*\*</sup> 정 회 원 : 광운대학교 컴퓨터과학과 교수  
dcjang@kw.ac.kr

논문접수 : 2005년 5월 17일

심사완료 : 2005년 7월 25일

프로그램 개발에 소요되는 시간과 인적 자원의 낭비를 줄이고, 웹 프로그램의 구조적 변경이나 유지보수가 쉽고 간편하며, 특정한 서버 측 프로그램 개발을 위한 특정한 응용 프로그램의 도움 없이 웹상에서 개발자가 단계적인 방법(step by step)에 의해 원하는 항목을 선택하거나 설정함으로써, XML을 기반으로 하는 서버 측 프로그램을 개발 할 수 있는 개발자를 위한 XML기반의 웹 서버 프로그램을 개발하는데 목적을 두고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 XML 관련 기술 부분에 관해 살펴보고, 3장에서는 XML 서버 구조, XML문서의 생성 과정, 개발자를 위한 웹 기반의 XML 개발 지원 서버 및 XML 트리에 대해 설명한다. 4장에서는 XML 트리를 기반으로 XML 스키마와 XML 스타일시트를 설계한다. 5장에서는 4장에서 설계한 내용을 기반으로 웹 프로그램 개발자를 위한 XML 기반의 서버 프로그램 생성을 위한 어플리케이션을 구현하고, 끝으로 6장에서 결론을 맺는다.

**2. 관련 연구**

본 장에서는 XML과 XML 스키마 그리고 XML 스타일시트에 관해 살펴본다.

**2.1 XML**

XML은 SGML의 복잡성을 제거하고, HTML의 한계를 뛰어넘어, 웹상에서 구조화된 문서의 전송이 가능하도록 설계된 표준화된 텍스트 형식의 문서이다[11].

SGML은 모든 Markup 언어의 모체라고 할 수 있으며, 특정한 타입을 기술하거나, 표준을 정해서 널리 사용되는 타입의 문서를 기술할 수 있고, 어플리케이션을 정의할 수 있는 웹 문서를 기술하는 가장 완벽한 언어일 것이다. 그러나 너무 복잡하고 정의하기 어렵다는 단점을 가지고 있다.

HTML은 웹을 위한 정형화된 문법을 가지고 표현되며, 사용법이 간단하고 편리하지만 확장이 불가능하고, 특수한 문서들을 표현하는데 많은 제약을 가지고 있다.

XML은 SGML이 갖는 복잡성을 해결하기 위한 목적으로 특별히 설계된 SGML의 하위 집합이다. 1996년 W3C의 XML Working Group에서 웹을 위해 최적화된 SGML의 단순화 버전인 SGML의 하위 집합을 XML이라고 이름 붙여 발표하게 되었다. 따라서 XML은 간단한 메모에서부터 복잡한 데이터베이스까지, 가상의 모든 종류의 정보를 기술하는데 사용할 수 있는 유연한 문법을 제공하게 되었다[1,6,11].

현재 XML은 Internet Explorer 5.0부터 HTML 페이지를 사용하지 않고, 스타일시트가 첨부된 XML 문서를 직접 열어서 볼 수 있게 되었다. 하지만 스타일시트인 XSL을 사용하게 되면, XML 데이터를 어떻게 표현해야

할지를 브라우저에게 알려주기 위해서 HTML을 사용해야 한다. 따라서 현재로서는 XML이 완전히 HTML을 대체하기보다는 HTML과 함께 사용되면서, 웹 페이지의 기능성에 대해 모든 타입의 문서를 전달하거나, 정보의 정렬, 필터링, 재배열, 검색 및 조작 등 고도로 구조화된 정보를 제공하는 방향으로 범위를 넓혀가고 있다.

**2.2 XML 스키마**

일반적으로 스키마란 객체의 클래스 특성을 정의하는 것이다. 따라서 XML 스키마 역시 XML 문서의 구조와 내용 그리고 의미 체계를 정의하는 광범위한 의미를 포함한다. DTD도 문서의 클래스를 정의하고 그 클래스의 요소와 특성의 형을 정의하기 때문에 하나의 스키마라고 할 수 있다. 그러나 기존의 DTD가 갖는 많은 문제점들 때문에 XML 스키마에 대한 제안이 나오게 되었다[6,10].

XML 스키마는 현재 W3C에서 권고(Recommendation)된 상태이며, XML 문서의 스키마를 정의하는 표준이다. DB 설계시 테이블의 구조 및 각 Attribute의 Data type을 정의하듯이, XML 스펙을 통해 사용할 수 있는 XML문서의 스키마와 Data type을 지정할 수 있게 하는 언어이다. 즉, XML 문서에 나타날 수 있는 Element와 그 Element의 Attribute를 표현할 수 있는 방법을 제공한다. XML DTD는 XML 표준에 따라 문서의 논리적 구조를 정의하는 방법을 제공하는 반면, XML 스키마는 XML을 사용하여 XML의 논리적 구조를 정의한다고 할 수 있다. 따라서 본 연구에서는 XML 스키마를 기반으로 XML 문서를 생성할 수 있도록 하고 있다. 표 1은 DTD와 XML 스키마에 대한 비교를 나타내고 있다.

표 1 DTD와 XML 스키마 비교

	DTD	XML Schema
문법	EBNF 기반 언어 :XML과 전혀 다름	XML 1.1 스펙
DOM지원	불가능	XML언어이므로 가능
Namespace	전역 이름만 사용	전역 및 로컬 가능
상속	불가능	가능
확장성	불가능	XML의 확장성과 동일
다중 어휘	XML문서에 하나의 DTD만 사용	XML문서 내에 참조 스키마는 네임스페이스 기반으로 무한 사용 가능
동적 연결	DTD는 읽기 전용(변경 불가)	Runtime시 동적으로 선택과 변경 가능

**2.3 XML 스타일시트(XSL)**

XML스타일시트인 XSL은 크게 2개의 부분으로 구성된다. 하나는 XML문서의 변환(Transformation)을 위한 언어이고, 또 다른 하나는 포매팅 구분(Formatting

Semantics)을 기술하기 위한 어휘(Vocabulary)로 이루어진다. 변환을 위한 언어인 XSLT(XSL Transformations)는 XPath(XML Path Language)와 함께 사용하여 XSL의 기본 골격을 구성한다[12]. 이러한 기본 골격 위에 포맷팅을 위한 어휘를 사용하여 XML문서를 표현하기도 하지만, 다른 형태의 문서를 생성하기도 한다. 그림 1은 XSL의 구조를 보여주고 있다.

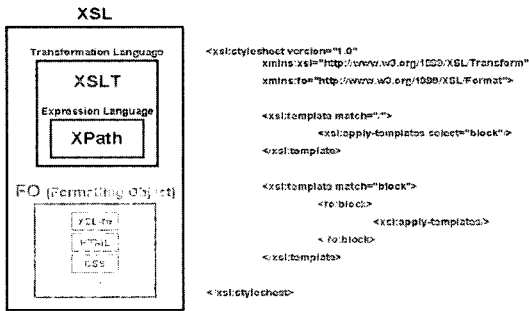


그림 1 XSL의 구조

XML문서의 처리과정은 클라이언트의 서비스 요청에 대해 XML 웹 서비스는 그림 3과 같이 수행하게 된다. 클라이언트가 요청한 내용을 템플릿 내에 있는 XML 문서의 매개변수로 읽어 들여, XML Query를 수행하게 된다. 수행된 결과는 XML 스키마에 의해 계층 구조를 갖는 XML 문서를 생성하게 되고, 클라이언트의 환경에 맞게 미리 작성된 XML 스타일시트인 XSL과 결합하여 클라이언트가 읽어 들일 수 있는 XML, HTML, WAP 등의 문서형태로 변환하여 전달하게 된다.

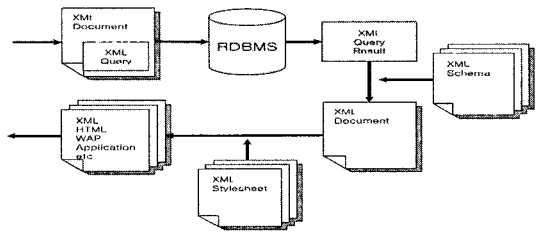


그림 3 XML 웹 서비스를 위한 처리 과정

3. XML 서버와 XML 웹 서비스

본 장에서는 XML 서버 구조와 XML 웹 서비스를 위한 처리과정, 그리고 본 논문에서 제안하는 웹 서버와 개발자 시스템 사이에 존재하는 웹 기반의 XML 개발 지원 서버의 구조 및 본 논문에 앞서 설계하고 구현된 XML 트리를 살펴보고, 이를 기반으로 XML 스키마와 XML 스타일시트를 생성할 수 있는 방안을 제시하고자 한다.

3.1 XML 서버 구조 및 문서 처리 과정

본 논문에서 사용할 XML 서버 구조는 그림 2와 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[13].

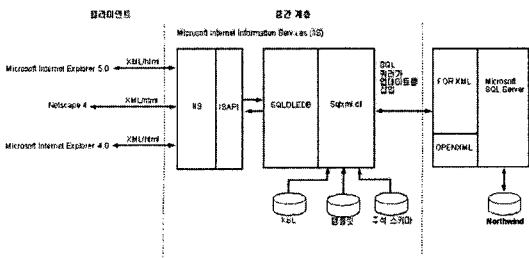


그림 2 XML 서버 구조

3.2 웹기반 XML 개발 지원 서버 구성

그림 4에서 개발자(A)는 자신이 관리하는 서버(C)에서 서비스해야 할 서버 측 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 패스워드를 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 트리를 생성한다 [14]. 이렇게 생성된 XML 트리는 XML 스키마 및 XML 스타일시트 생성을 위해 사용하게 된다.

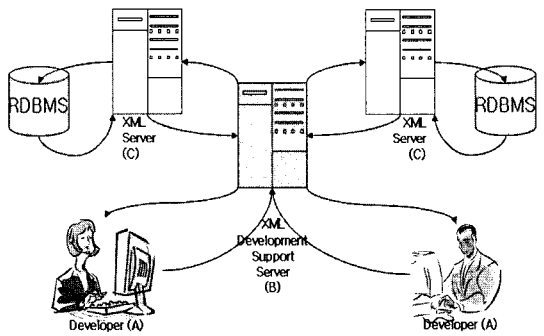


그림 4 개발자를 위한 웹 기반의 XML 서버 프로그램의 개발 구성

3.3 XML 트리

본 논문에서 사용되는 XML 트리는 그림 5에서 보는 바와 같이, 개발자가 개발 지원 서버의 도움을 받아 단계별로 필요한 값들을 설정 또는 선택함으로써, XML 문서 생성에 필요한 정보를 수집하여 그림 6과 같은 XML 트리를 생성하게 된다[14].

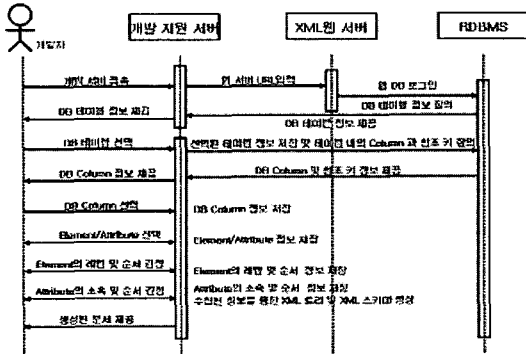


그림 5 XML 트리 생성을 위한 시퀀스 다이어그램

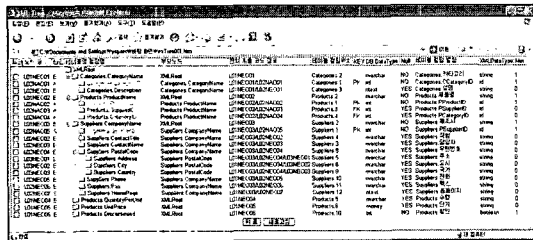


그림 6 XML 트리 실행 화면

그림 5를 통해 수집된 정보는 데이터베이스 테이블내의 각각의 컬럼들이 각각의 XML 트리 내에서 각각의 노드가 되어 저장 되는데, 그 구성을 보면 크게 3가지 유형으로 분류된다.

첫째, XML 트리를 구성하기 위한 부분으로 노드코드, 전체 계층 경로 코드가 이에 해당된다. 노드코드는 노드의 이름으로 사용되며, 여기에 사용된 8자리의 문자 코드 중 처음 3자리(L01)는 노드의 레벨을 나타내고, 다음 2자리(NE/NA)는 Element 또는 Attribute인지를 나타내며, 마지막 3자리(O01)는 레벨의 순서를 나타내는 정보를 갖고 있다.

둘째, 관계형 데이터베이스로부터 얻게 되는 정보로 컬럼위치, 키값, 컬럼의 Data Type, null허용여부가 이에 해당된다. 컬럼의 위치는 XML Query를 사용할 때, 컬럼이름 대신 사용하게 되고, 키 값(주키/외래키)은 이 노드가 XML 문서 내에서 요소(Element)의 속성(Attribute)으로 사용되며, 테이블 조인이 있을 경우에

사용되게 된다.

셋째, XML문서의 생성을 위해 개발자가 정의한 부분으로 컬럼의 별칭과 XML Data Type이 있는데, 이들은 XML 스키마를 생성하는데 사용된다.

이렇게 생성된 XML 트리내의 노드들이 XML문서를 생성할 때 사용되는 부분은 표 2와 같다.

표 2 XML Tree 구성 정보의 적용 부분

명칭	내용 예제	Tree	Schema	Stylesheet	XML
노드코드	L01NEO01	O		O	
테이블/컬럼명	Categories, CategoryID	O	O		
전체 계층 코드 경로	L01NEO01/L02NAO01	O			
컬럼 위치	Categories.2	O			
KEY	PK	O	O		O
테이블 컬럼 별칭	Categories, 카타고리		O	O	O
XML Data Type	string		O		
Min	1		O		

4. XML 스키마 & 스타일시트 설계

본 논문의 구현 예로 사용한 데이터베이스 테이블간의 관계를 다이어그램으로 표현하면 그림 7과 같다.

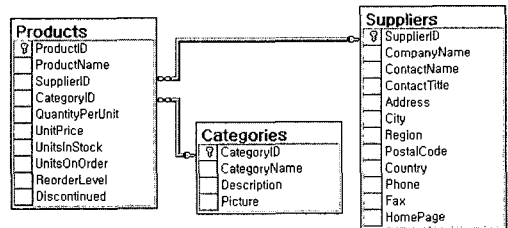
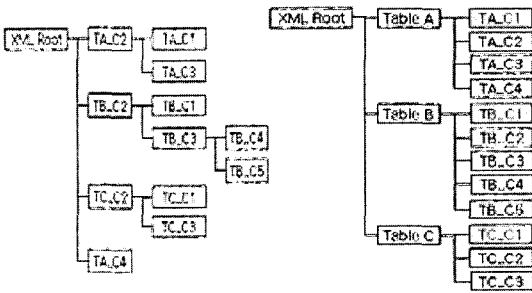
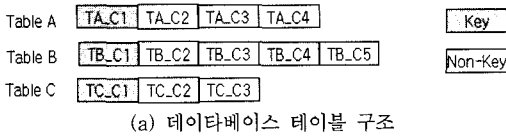


그림 7 데이터베이스 테이블 다이어그램

4.1 XML 스키마 설계

XML 스키마는 이미 생성된 XML 트리의 정보를 이용하여 생성하게 된다. 그리고 XML 스키마를 통해 생성되는 XML 문서가 갖게 될 계층구조는 그림 8의 (a)와 같은 데이터베이스 구조에서 XML 트리 구조를 갖는 (b)유형과 RDBMS 구조를 갖는 (c)유형 2가지가 가능하다.

그러나 XML 스키마를 구성하기 위해 사용된 테이블이 그림 8의 (a)와 같이 하나 이상일 경우, 각각의 테이블에 대한 참조키를 통해 테이블간의 관계를 설정하게 되지만, 이질적인(heterogeneous) 데이터베이스 테이블이 통합되지 않는 한, 하나의 부모 노드에 서로 다른 테



(b) XML 트리 구조 (c) RDBMS 구조  
그림 8 XML 스키마가 적용된 XML 문서의 유형

이들간의 컬럼들이 혼합되어 자식 노드로 존재하기는 어렵다[9]. 또한, 사용된 컬럼들의 변경 없이 계층구조만 변경되었을 경우 (b)유형은 XML 스키마를 새로 작성해야 되지만, (c)의 유형은 XML 스키마를 변경하지 않아도 된다. 따라서 본 연구에서는 (c)의 유형으로 설계 및 구현을 하고, XML 스타일시트를 통해 최종적으로 XML 트리가 갖는 형태로의 웹 서비스를 제공할 수 있도록 하였다.

XML 스키마를 생성하기 위해서는 XML 트리를 통해 얻은 정보 중 컬럼의 별칭(alias)과 XML Data Type, 데이터베이스의 테이블명과 컬럼명 그리고 키에 대한 정보가 사용된다.

#### 4.2 XML 스타일시트 설계

XML 스타일시트는 XML 스키마와는 달리 XML 트리의 계층구조 및 테이블 컬럼 별칭 정보를 활용하며, 그림 9에서 최상위 노드인 XMLRoot를 시작으로 부모 노드와 자식노드가 존재하는 만큼 여러 개의 문서를 생성하게 된다[15].

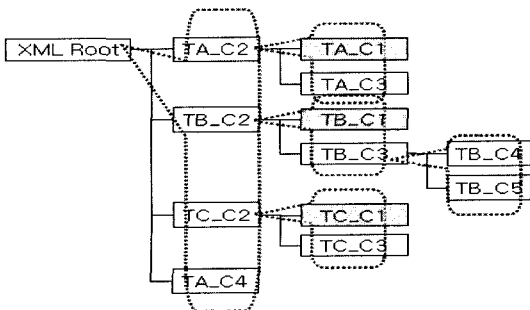


그림 9 XML 트리의 계층 구조

생성되는 모든 XML 스타일시트들은 그림 10과 같은 template 규칙에 의해서 매핑 된다.

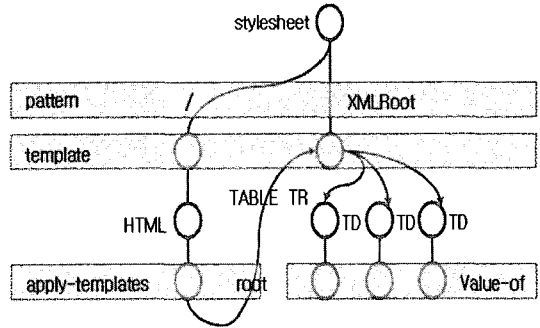


그림 10 template의 매핑 과정

### 5. 구현

#### 5.1 XML 스키마 생성을 위한 구현

최근 자주 사용되고 있는 XML 스키마 종류로는 W3C에서 2001년 표준으로 정하고 권고사항인 XSD(XML Schema Description)와 1999년 MS에서 개발자를 위한 표준으로 정하고, W3C에서 허용한 XDR(XML Data Reduced)이 있다. 본 연구에서는 XDR을 기준으로 구현하였다.

XML 스키마의 원시 코드는 다음과 같은 구조를 갖는다.

```

1. XML 선언
2. Schema 선언 (XML namespace 포함)
3. 사용될 테이블과 키관계에 대한 ElementType 정의
// ElementType의 name을 "XMLRoot"로 하고,
// 서브 노드로 element의 type은 "테이블명"으로
// 하는 테이블간의 키 관계를 정의
4. 앞에서 정의한 element에 대한 ElementType정의
//name은 "테이블명"으로 정의하고, 서브 노드로
//각각 attribute와 element에 대한 typed은 "별칭",
//DB field와 매핑을 위한 sql:field는 "컬럼명"으로
//정의
5. XML 문서 내에서 사용되는 attribute와 element의
AttributeType과 ElementType에 대한 정의
//속성으로 name은 "컬럼별칭", dt:type은 "XML Data
Type"을 사용
6. Schema종료 선언
    
```

위의 XML 스키마 구조 중에서 3번째 단계인 테이블 간에 조인하는 부분을 생성하는 알고리즘은 다음과 같다.

1. 외래키의 정보를 통해 주키의 테이블명 정보를 얻는다.  
(ex: Products.CategoryID=FK\_Products\_<strong>Categories</strong>)
2. 주키의 테이블명 정보를 통해 해당 테이블의 컬럼명을 얻는다.  
(ex: Categories.<strong>CategoryID</strong>=PK\_Categories)
3. 1,2번을 통해 테이블간의 2테이블과 키에 대한 관계를 설정하고, foreign-relation에 해당하는 테이블을 element의 type으로 설정한다.  
(ex: <element type="<strong>Categories</strong>">  
    <sql:relationship key-relation="Products"  
        key="CategoryID"  
  
    foreign-relation="<strong>Categories</strong>"  
  
    foreign-key="CategoryID"  
    </element>  
</>
4. 3번 과정을 반복한 후 주키 정보에는 있고, 외래키 정보에는 없는 테이블을 찾아 자신이 주키 및 외래키 관계가 되도록 설정한다.  
(ex: <element type="<strong>Products</strong>">  
    <sql:relationship key-relation="<strong>Products</strong>"  
        key="ProductID"  
  
    foreign-relation="Products"  
  
    foreign-key="<strong>ProductID</strong>"  
    </element>  
</>
5. 모든 테이블간의 관계 설정이 완료되면, ElementType, 의 name은 "XMLRoot", sql:relation은 4번에서 찾은 "테이블명"으로 설정하고, 앞에서 구한 element를 서브노드로 설정한다.  
(ex: <ElementType name="<strong>XMLRoot</strong>"  
    sql:relation="<strong>Products</strong>">  
    <element type="Product">  
        <sql:relationship정의 />  
    </element>  
    ..... // 나머지 element 반복  
    </ElementType>  
</>

**5.2 XML 스타일시트 생성을 위한 구현**

XML 스타일시트의 원시 코드는 아래와 같은 구조를 갖는다.

1. XML 선언
2. Stylesheet 선언 (XML namespace 포함)
3. xsl:template match="/" 정의  
    // root pattern에 대한 내부 내용 기술  
    // apply-templates로 'root' 선택 포함
4. xsl:template match='XMLRoot' 정의  
    //<strong>그림9</strong>의 XML Tree의 계층구조를 기반으로  
    // value-of의 값에 타입이 Element인 모든 자식  
    // 노드의 "테이블 컬럼 별칭"을 삽입한다.

```
// 만약 해당 노드의 손자노드가 존재하고,  
  
// 손자 노드 중 타입이 Element노드가 있을 경우  
// 링크를 위한 별도의 처리를 기술한다.  
  
5. Stylesheet 종료 선언
```

또한 위의 과정을 더 이상의 리프노드가 없을 때 까지 반복한다.

위의 XML 스타일시트 구조 중 4번째 단계에서 노드의 타입이 Element인 손자노드가 존재할 경우와 그렇지 않을 경우에 대한 알고리즘은 다음과 같다.

1. 부모노드에 대한 모든 자식노드를 검사한다.
2. 자식노드의 손자노드가 존재하지 않거나 손자노드가 존재하지만 타입이 모두 attribute인 경우에는 value-of의 select값으로 "테이블 컬럼 별칭"을 할당한다.  
(ex: <xsl:value-of select='Product/<strong>제품명</strong>' /> )
3. 2번이 아닌 경우에는 링크를 위해 자식노드의 "노드 코드"명을 파일명으로 하는 XML문서와 XSL문서가 생성되어야 한다.
4. 3번에서 자식노드 중 타입이 attribute인 노드가 존재하면, 링크를 위한 매개변수와 값으로 테이블 조인에 사용된 메인키를 갖는 노드와 attribute인 노드 모두의 "테이블 컬럼 별칭" 모두를 사용하고, 그렇지 않은 경우에는 메인키를 갖는 노드에 대한 "테이블 컬럼 별칭"만 사용한다.  
(ex:  
<A>  
<xsl:attribute name='href'>  
    L01NEO01.xml?PPProductID=  
<xsl:value-of select='Products/@PPProductID/'>  
&CCategoryID=  
<xsl:value-of select='Categories/@CCategoryID/'>  
&xsl:L01NEO01.xsl  
</xsl:attribute>  
<xal:value-of select='Categories/<strong>카테고리</strong>' />  
</A>  
</>

**5.3 XML 문서 생성을 위한 구현**

XPath 질의를 포함 XML문서의 원시 코드는 다음과 같은 구조를 갖는다.

1. XML 선언  
(ex: <?xml version='1.0'?> )
2. root 선언 (XML namespace 포함)  
(ex: <root xmlns:sql=  
    'urn:schemas-microsoft-com:xml-sql'  
>  
</>

```

3. 매개변수를 위한 sql:param정의
// (단, XMLRoot 노드인 경우 제외)
{ex: <sql:header>
  <sql:param name='PProductID' />
  <sql:param name='CCategoryID' />
</sql:header>
}

4. sql:xpath-query선언
// mapping-schema에 XML Schema링크
{ex: <sql:xpath-query mapping-schema
  ="schema001.xdr">
}
// 3에서 매개변수 선언된 경우,
// XPath형태의 질의 작성
{ /XMLRoot
  [Products/@PProductID=$PProductID]
  [Categories/@CCategoryID=$CCategoryID]
}

5. xpath-query 종료 선언
{ex: </sql:xpath-query> }

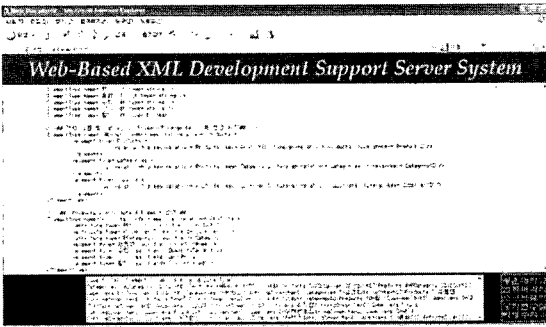
6. root 종료 선언
{ex: </root> }
    
```

위의 5-1,2,3과정을 통해 생성된 XML 웹 서비스를 위한 원시코드의 생성 과정은 그림 11과 같다.

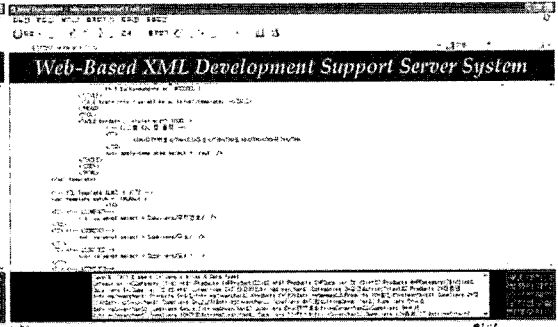
5.4 기존 연구와의 비교 분석

본 연구에서 생성되는 문서들은 개발 지원 서버를 중심으로 웹기반에서 이루어지고 있기 때문에, 특정한 응용 프로그램이 없이도 인터넷이 사용 가능한 곳이라면 언제 어디서나 프로그램 생성이 가능하다. 따라서 기존의 응용프로그램 위주로 생성되는 연구들과는 차별화가 된다.

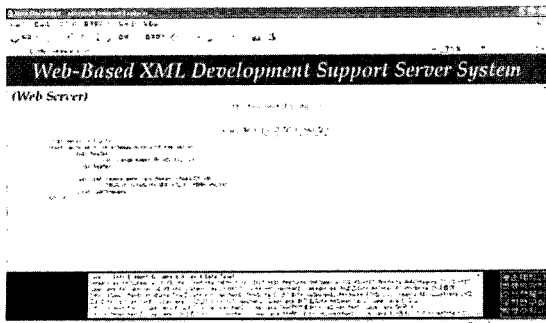
또한, 기존의 XML 트리는 XML 문서의 가독성을 높이기 위한 방법으로 사용되고 있지만, 본 연구에서의 XML 트리는 관계형 데이터베이스로부터 테이블들의 정보를 얻어 개발자가 XML 문서의 계층 구조를 형성하는데 사용되고 있다. 따라서 XML 스키마나 XML 문서에 대한 정의 없이도 XML 트리가 갖고 있는 계층구조 정보, 테이블 및 필드정보, 각 테이블간 조인 및 키값에 대한 정보를 활용하여 XML 스키마, 스타일시트, XPath 질의를 내포하는 XML 문서까지 한번에 자동생성이 가능하도록 설계하고 구현하였다. 따라서 본 논문에서의 연구 내용과 기존 연구와의 비교 분석한 내용을 정리하면 표 3과 같다.



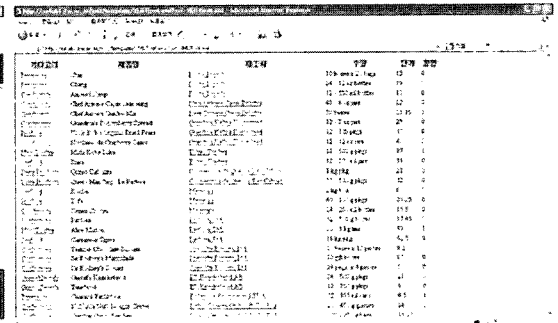
(a) XML스키마 생성과정



(b) XML스타일시트 생성과정



(c) XML문서 생성과정



(d) XML문서 실행화면

그림 11 원시코드 생성 & 실행 화면

표 3 본 연구와 기존 연구와의 비교분석

평가 항목	기존 연구	본 연구
생성 기반	응용프로그램 기반	웹 기반
XML Tree 사용 목적	가독성을 높이기 위한 목적	스키마, 스타일시트 생성에 필요한 정보 제공
문서의 자동생성 방법	XML문서->스키마추출 or 스키마->XML문서생성	XML Tree->XML 스키마, XML 스타일시트, XML문서 자동 생성
오류 발견 시점	응용프로그램이므로 최종 테스트과정에서 발견	웹기반 이므로 프로그램 생성과정에서 발견
유지보수/재사용성	가능(보통)/재작성이 유리함	가능(우수)/XML Tree의 재구성으로 간편함.
개발 방법	자체 개발 중심	아웃소싱기법 적용가능

6. 결론

본 논문에서는 개발자가 개발지원 서버를 통해 빠른 시간 내에 관계형 데이터베이스로부터 자신이 원하는 형태의 XML 트리를 생성하고, 이를 기반으로 XML 스키마 생성뿐만 아니라 XML 스타일시트를 쉽게 생성하는데 목적을 두고 있다. 이렇게 됨으로서 사용자에게는 다양한 형태의 서비스가 가능하고, 개발자 역시 프로그램 개발 시간의 단축 및 유지 보수가 쉬워지고, 급격한 트렌드 변화에 유연하게 대처할 수 있을 것으로 기대된다.

향후 연구과제로는 XML 트리의 기능을 향상시켜 무선 웹 서비스를 제공할 수 있는, XML 스키마, XML 스타일시트의 자동 생성을 위한 개발과 연구가 필요하다고 본다.

참고 문헌

[1] Kevin Williams et al, "Professional XML Databases," Wrox Press, 2001.  
 [2] H. Schoning, "Tamino-a DBMS Designed for XML" in Processing of IEEE ICDE, 2001.  
 [3] Ronald Bourret, "XML and Databases," <http://www.rpbouttet.com/xml/XMLAndDatabases.htm>, Last updated July 2004.  
 [4] Ronald Bourret, "XML Databases Products," <http://www.rpbourret.com/xml/XMLDatabaseProds.htm>, Last updated Sep 2004.  
 [5] Kevin Dick, "XML A Manager's Guide" Second Edition, Addison Wesley, 2002.  
 [6] W3C online, <http://www.w3.org/> (XML/DOM/XML Schema/XML Query)  
 [7] msdn online(XML), <http://msdn.microsoft.com/xml>  
 [8] J. Craig Cleaveland, "Program Generators with XML and Java," PH PTR, 2001.  
 [9] U-Chang Park, "A Database Schema Integration Method using XML Schema," Korean Society for Internet Information Vol. 3-2, Apr, 2002.  
 [10] Jon Duckett at al, "Professional XML Schemas,"

Wrox Press, 2001.

[11] Deitel et al, "XML How to Program," Prentice Hall, 2000.  
 [12] Neil Bradley, "The XSL Companion," Addition Wesley, 2000.  
 [13] Danny Ayers et al, "Professional Java Server Programming," Wrox Press, 1999.  
 [14] 박영수, 장덕철, "RDBMS로부터 XML 스키마 생성을 위한 웹 서버 기반의 XML Tree 설계 및 구현", 한국정보과학회, 학술발표대회, 2004. 10.  
 [15] Nathan Ridley, "Jscript Tree List Control," <http://www.codeproject.com/jscript>, Dec 2002



박 영 수

1996년 광운대학교 전산대학원 전자계산학과(이학석사). 1999년 광운대학교 대학원 컴퓨터과학과 수료. 1998년~현재 경북대학 겸임교수. 관심분야는 XML, 분산처리, 객체지향프로그래밍, 소프트웨어공학, 무선인터넷, 모바일컴퓨팅 등



장 덕 철

1982년 고려대학교 대학원 경영정보학 박사. 1981~1982년 버클리 대학교 객원 교수. 1993년 광운대학교 전산사회교육원 원장. 1996년 광운대학교 전산대학원 원장. 1997년 광운대학교 이과대학 학장. 1997년~현재 광운대학교 컴퓨터과학과 교수. 관심분야는 소프트웨어공학, 버전제어, 컴포넌트, 객체지향설계방법론 등