

Tag free XML 개발 툴을 이용한 XML문서 개발과 RDB 매핑에 관한 연구

(A Study on Developing XML Documents and RDB
Mapping Using Tag Free XML Development Tools)

우 원 택*
(Won-Taek Woo)

요 약 본 연구에서는 Tagfree의 XML개발 툴을 이용하여 XML, DTD, XML스키마 등의 문서를 실제로 개발하여 이들 간의 상관관계를 봄으로써 XML프로그래밍과 문법을 쉽게 해독하여 웹 문서를 보다 용이하게 개발하고 또한 XML데이터와 MS SQL과 같은 RDB 데이터의 저장관리와 매핑과정을 구현해 봄으로써 트리개념의 XML문서와 테이블개념의 데이터베이스가 어떻게 이론적으로 연동되며 실제 구현시의 한계와 장단점은 없는지를 규명해 봄으로써 전자상거래나 무선인터넷 구축 시 XML문서의 이용 효율성을 높이는데 그 목적이 있다.

핵심주제어 : XML프로그래밍, RDB 매핑, 태그프리 XML 개발툴

Abstract XML architecture is important to wireless or wired presentation. It is also important for data exchange between businesses. XML forces the separation of content from style and gives wired or wireless devices flexibility for interpretation. Separation of content, style, and logic is key to advanced architecture. XML can be exchanged among databases on multiple systems with presentation on wired or wireless devices. An XML schema might need to be defined, or an established DTD might need to be transformed to access a relational database on the server. The purpose of this study is to generate XML documents such as DTD, XML schema, RDB mapping using Tagfree's XML developer tools in order to experience whole processes mentioned above. Overall understanding of data structures of and database processing with XML documents, which is essential to XML programming and database processing, can be accomplished with this study without much endeavor to learn complex XML syntax. The future study can be extended on the subject of web programming with DOM or SAX API.

Key Words : XML programming, RDB mapping, Tagfree XML developer

1. 서 론

일반적으로 문서(Document)는 글자(Text), 그림(Images), 영상(Video), 소리(Sound) 등의 멀티미

디어 데이터로 구성되어 표현된다. XML문서에서는 문서를 구성하는 이와 같은 데이터들을 하나의 요소(Element)로 보고 이들 요소들의 조합으로 문서를 표현한다. 그러나 XML문서는 일반적인 문서들이 가지고 있는 문서의 글꼴이나 테두리, 배열형식과 같은 스타일(Style)에 대한 정보를 제외한 순

* 대구한의대학교 모바일콘텐츠학부

수 데이터 정보만을 요소나 속성으로 가지고 의미론적 태그(Semantic Tag)를 이용하여 문서를 표현한다. 따라서 XML문서는 일반문서와 같이 사람이 시각적으로도 인식할 수 있고 의미론적 태그를 포함하고 있기 때문에 컴퓨터도 인식할 수 있는 완전히 고급단계(High-level)의 축약문서(Abstraction Document)이다. 즉, XML문서는 콘텐츠와 스타일 명령만으로 정의된 축약된 문서로써 웹브라우저에 의해 해석되어 스크린이나 프린터로 출력된다. 이는 Java 나 C++ 같이 프로그램으로 실행되는 언어와는 달리 글자(text)와 스타일문만으로 구성된 단순 콘텐츠로써 저급단계(Low-level)의 축약문서인 비트맵이미지나 PostScript 문서, 중급단계의 축약문서인 HTML문서보다는 더욱 축약된 형태로 문서를 표현함으로써 문서의 무게나 사이즈가 가벼워 저서 데이터 트래픽이 빈번한 인터넷이나 모바일문서의 자료 전송에 적절하다.[1][2][3] 그리고 XML문서는 관계데이터베이스 자료로도 쉽게 변환할 수 있고 데이터만 내장된 관계데이터베이스의 자료를 XML문서로 변환하여 스타일을 적용함으로써 사람들이 쉽게 알아볼 수 있는 문서형태로 표현할 수도 있다. 그러나 XML문서의 구성은 XML, DTD, XSLT, XPATH, XPointer, XLink 등의 여러 언어로 표현되고 SAX, DOM 등의 XML 문서에 대한 인터페이스 언어도 다양하므로 실제로 XML의 문법을 이해하기란 간단한 일이 아니다. 본 연구에서는 XML문서가 다른 컴퓨터문서들과 어떠한 차이를 가지는지를 자료구조적인 관점에서 분석함으로써 복잡한 XML언어의 문법을 쉽게 이해하고 e-business나 모바일콘텐츠 개발에 쉽게 활용하고자 함에 그 목적이 있다. 이를 위해 본 연구에서는 Tagfree의 XML개발 틀을 이용하여 XML, DTD, XML스키마 등의 문서를 실제로 개발하여 이들 간의 상관관계를 봄으로써 XML프로그래밍과 문법을 쉽게 해독하여 웹 문서를 보다 용이하게 개발하고 또한 XML데이터와 MS SQL과 같은 RDB 데이터의 저장관리와 매핑과정을 구현해 봄으로써 트리개념의 XML문서와 테이블개념의 데이터베이스가 어떻게 이론적으로 연동되며 실제 구현시의 한계와 장단점은 없는지를 규명해 봄으로써 전자상거래나 무선 인터넷 구축 시 XML문서의 이용 효율성을 높이는데 그 목적이

있다.

2. 문서의 추상화와 XML 문서 특성

XML은 문서태그를 붙여 추상화(Abstract)한 문서로 문서의 내용과 표현이 분리되어 있으므로 문서의 사이즈가 적어서 많은 양의 저장 공간이 필요하지 않고 전송이 용이하며 문서의 내용과 표현이 분리되어 있기 때문에 전송된 문서는 유무선기기의 상태에 맞도록 플랫폼 독립적으로 변환될 수 있는 장점이 있다. 본장에서는 XML의 이러한 장점을 살펴보기 위해 문서의 추상화와 스타일의 분리 및 시각화에 대해 살펴보겠다.

2.1. 문서의 추상화와 XML

추상화란 말은 애매모호하게 들릴 지도 모르지만 아주 중요하고 실제적인 개념이다. XML마크업 어휘를 개발 한 후 적용하는 절차는 문서의 여러 부분을 추상화시켜 태그를 붙이는 작업을 말한다. 컴퓨터에서 문서가 표현되는 방법은 여러 가지가 있으나 여러 문서 표현에서 다음의 문서는 이전 문서로부터 얻어지도록 순서화하는 것이 자연스럽다. 그러나 역방향으로의 변환은 자동적으로 되거나 신뢰성 있게 되지는 않는다. 예를 들면 Tex문서는 PostScript 문서로 자동 번역되나 그 역 변환은 표준 틀로는 되지 않는다. 역변환의 일부를 자동화할 수는 있지만 항상 수동적인 체크와 수정을 요구하며 항상 신뢰성 있고 통일적으로 되지는 않는다. 이와 같은 경우 Tex문서는 PostScript 문서보다는 상위단계의 추상화 (higher level of abstraction)문서라고 한다. 비슷하게 PostScript 같은 문서는 JPEG이미지 같은 비트맵문서보다는 상위의 문서이다. 그리고 Tex는 LaTeX 보다는 하위이다.[4] LaTeX는 도널드 크누스 (Donald E. Knuth)의 식자언어(typesetting language)에 근거하여 1985년에 레슬리 램포트 (Leslie Lamport)에 의해 개발되었다. LaTeX로 <표 1>과 같은 워드 프로세스 문서를 작성하려면 문서작성자는 어떠한 배치(layout)로 제목은 18크기 Times Roman 체로 이름은 12크기 이탤릭체로 등의 스타일을 선택

<표 1> 워드프로세스문서

```

Cartesian closed categories and the
price of eggs
Jane Doe
September 1994
Hello world!

```

해야 한다. LaTeX는 이와 같은 문서디자인은 문서디자이너에게 맡기고 문서작성자는 문서작성에만 전념하게 하는 시스템이다. <표 1>의 문서를 LaTeX문서로 변환하면 <표 2>와 같다.[5]

<표 2> LaTeX문서

```

/documentclass{article}
/title{Cartesian closed categories and the
price of eggs}
/author{Jane Doe}
/date{September 1994}
/begin{document}
/maketitle
Hello world!
/end{document}

```

LaTeX는 주로 과학문서의 작성에 사용되었으나 어떤 유형의 출력문서에도 사용할 수 있다. 이와 같이 XML문서도 문서의 내용과 표현을 분리한 문서로 <표 1>을 XML문서로 변환하면 <표 3>과 같다.[6]

<표 3> LaTeX문서의 XML변환

```

<article>
<title> Cartesian closed categories and
the price of eggs</title>
<author>Jane Doe</author>
<date>September 1994</date>
<body><maketitle>Hello world!</maketitle></body>
</article>

```

만약 두 문서가 양방향으로 자동 변환될 수 있으면 같은 단계의 추상화라고 한다. 따라서 모든 컴퓨터문서(computerized document)는 최소 추상화 단계에서 최대 추상화 단계로 이어가는 추상화 문서로 만들 수가 있다. 높은 단계의 추상화문서일수록 아래 단계로 변환될 수 있는 더 폭 넓은

포맷을 가질 수 있다. 따라서 추상화의 맨 아래에 있는 문서형식은 막다른(dead ends) 추상화 단계로 이 문서는 단지 보이는(perusal) 역할만 있다. 그러나 추상화의 맨 위 단계의 문서는 편집(authoring), 저장(storage), 교환(exchange), 분석(analysis) 모두가 가능하다.[6]

2.2. 문서의 스타일 분리

문서의 표현 양식을 나타내는 스타일 정보는 추상화 단계의 상위 단계로 갈수록 그 정보의 양이 줄어든다. 예를 들면 비트맵이미지나 PostScript와 같은 하위단계의 추상화 문서는 단지 페이지의 기술만을 나타내므로 내용(what is being said)과 표현(how it is being said)을 분리하여 표현할 수가 없다. 따라서 이 단계의 문서를 스타일이 없는 일반의 텍스트형식문서로 쉽게 내보낼 수(exporting)가 없다. 그러나 HTML과 같은 중간단계의 문서는 스타일 정보를 인터프리트 명령어로 단순화하여 페이지 이미지를 나타내며 필요에 따라 CSS 스타일 시트를 만들어 스타일 정보를 문서로부터 분리할 수 있다. 이 단계의 문서는 대부분의 경우에 일반텍스트(plain text)로 내 보낼 수 있다. 최상위 단계의 추상화 문서는 스타일정보를 완전히 제거하므로 써 전혀 볼 수가 없으므로 문서를 보려면 스타일 정보를 첨가하여 약간의 변환과 통합 과정을 거쳐야 볼 수 있다. 최상위 추상화 단계의 문서의 예는 의미(semantic) XML이 있다. 여기서 의미라는 말은 스타일은 없고 내용만을 가지는 태그문서를 말한다. 실제로 스타일과 문서를 구분하는 경계선을 쉽게 그을 수는 없다. 단지 추상화 단계라는 정의는 문서의 내용과 스타일의 분리라는 개념을 쉽게 이해하게 해주는 역할을 한다. 만약 문서의 일부 정보를 스타일시트와 같은 외부데이터나 문서 내에서 자동으로 추출할 수 있다면 이 정보는 스타일정보가 되며 내용을 나타내는 것은 아니다. 예를 들면 번호가 붙여진 표제에서 번호는 표면상으로는 숫자라는 내용정보이나 실제로는 스타일을 나타낸다. 추상화의 단계가 높아질수록 문서의 구조는 더욱 다양하게 변화될 수 있다. 하위 단계의 표현은 약간의 유형(types)과 긴 설명의 엘리먼트를 사용하나 상위 단계의 문서는 아주

많은 종류의 엘리먼트 유형과 약간의 엘리먼트 내용(element instance)만을 가진다. 예를 들면 비트맵이미지는 단지 하나의 엘리먼트 타입인 픽셀(pixel)만을 가지나 PostScript는 다수의 기본 명령어를 가지고 상위 단계의 XML문서는 수백 개의 엘리먼트 타입을 가지고 있다. 모듈화(modularity)의 측면에서 보면 하위단계의 표현은 단조로운 경향을 띤다. 통상 하나의 큰 파일이 텍스트, 이미지, 스타일, 폰트 등 모든 것을 포함하고 있다. 그러나 상위 단계의 표현은 모듈화 형식이다. 문서의 구성 요소들은 명시적 또는 묵시적으로 별도의 파일로 존재하여 루트문서에 링크되어 있다. 이때 루트문서는 단지 문서의 내용부분만을 가지고 있다. 스타일, 구조, 모듈화 측면을 종합해 볼 때 추상화 단계가 올라갈수록 문서의 사이즈는 줄어들게 된다. 따라서 추상화의 목적은 실제세계의 다양성을 약간의 기본 명령어로 바꾸어 표현하기 때문에 많은 양의 저장 공간이 필요하지 않게 된다. 문서의 비트맵 표현은 저장 공간이 가장 크고 PostScript 파일은 약간의 저장 공간이 필요하고 XML은 가장 적은 공간을 차지하는 문서이다. 그러나 XML은 그 자체가 추상화가 아니고 구조적 데이터의 표현(notation for structured data)이다. 문서표현을 포함하고 있는 모든 컴퓨터데이터는 구조적이므로 원칙적으로 XML은 추상화의 어느 단계에서도 사용될 수 있다. 다시 말하면 XML은 반드시 의미만을 나타내는 것은 아니라고 할 수 있다. 예를 들면 각각의 픽셀이 XML엘리먼트로 표현되는 XML기반의 비트맵형식도 가능하며 비 의미(nonsemantic) XML인 XSL-FO 언어도 있다. XSL-FO문서는 하위단계의 추상화 문서인 PDF문서보다 더 크고 부피가 크다.

2.3. XML문서의 특징

XML의 주요기능은 문서의 내용(content)과 표현(presentation)을 분리하는 것이다. 문서에서 내용과 양식(style)을 분리한다는 개념은 컴퓨터의 탄생부터 시작되었다. 즉 컴퓨터는 인간이 개발한 언어에 비유될 수 있으며 양자 모두 인간에 의해 탄생되었으며 특별한 기호를 사용하여 정보를 전달하고 표현하는데 사용된다. 컴퓨터문서의 탄생초

기부터 표현과 내용의 분리라는 아이디어가 인지되었으며 1969년에 GML(generalized markup language)이라는 마크업 언어가 IBM에 의해 최초로 만들어 졌다. 이는 메인프레임기반의 출력시스템에서 여러 응용프로그램을 통합하여 서로 다른 종류의 문서를 저장하고 처리할 수 있도록 개발되었다. 이후 이 언어는 ISO에 의해 SGML(standard generalized markup language)로 표준화 되었고 W3C가 이를 더욱 단순화하고 효율적으로 만들어 1996년에 XML로 개발하였다. XML은 문서의 서로 다른 부분을 분리할 수 있게 하였고 컴퓨터가 이를 식별할 수 있게도 하였다. 이에 따라 컴퓨터는 아이디어를 표현하는 도구로서의 역할 뿐 아니라 아이디어를 실행하는 도구가 되었으며 이는 인간의 철학을 응용과학화(applied science)한 것이라 할 수 있겠다. 마크업은 원래 출판업계에서 편집자와 인쇄디자이너가 교정이나 식자를 위해 색인 및 문서 이외의 부분에 대해서 서체나 글자크기, 글꼴 및 단 등 레이아웃에 관한 지시를 수작업으로 원고에 기입하는 것에서 유래되었다. 컴퓨터가 발전하여 워드프로세싱에서 전자문서가 이용되게 되자 이런 전자문서를 처리하기 위한 마크업이 생겨나게 되었고 마크업은 양식적(Stylistic), 구조적(Structural), 의미적(Semantic)마크업으로 분류되는데 예를 들어 HTML의 <H1>, , <I>, , <U>태그 같은 것들이 양식적 마크업에 속하고 HTML에서 제목, 문단, 컨테이너 섹션을 뜻하는 <Hn>, <P>, <DIV> 태그 같은 문서의 구성 방식을 표현하는 것들이 바로 구조적 마크업에 속한다. 의미적 마크업이란 데이터의 내용 자체에 관한 것으로 HTML의 경우 <TITLE>이나 <CODE>등을 예로 들 수 있다. 마크업언어는 문서의 구조와 내용에 추가적인 의미를 부여하는 마크업 규칙들을 규정하는 언어로써 SGML, HTML, XML과 같은 언어들이 있다. 마크업 언어는 절차적 마크업과 일반화된 마크업으로 나눌 수 있는데 절차적 마크업은 문서를 재현하거나 출력하기 위해서 문서의 물리적 속성을 분석하여 특성화하는 것으로 이기종간의 호환이 불가능하며 다른 형태의 문헌에 대해서는 마크업의 변형이 필요하다. 또한 형태만을 규명하는 것이므로 문서의 구조에 대한 정보는 알 수 없다. 절차적 마크업의 대표적인 워드프

로세스의 예로 RTF(Rich Text Format)이 있으며 한글97에서 RTF파일의 구조를 살펴볼 수 있다. 범용(Generalized)마크업은 문서의 논리적 구조 및 속성을 표현하는 마크업으로서 이 기종간의 호환 유지가 가능한 방법이다. 이러한 마크업은 여러 가지 형태로 발전하였는데 특히 컴퓨터에서 호환성이 없는 디지털 문서의 문제점을 해결하기 위하여 1986년 세계 공통 문서형식인 SGML(Standard Generalized Markup Language)을 UN 산하 표준 기관인 ISO(International Organization for Standardization)에서 ISO8879 규격으로 제정하였다.[7]

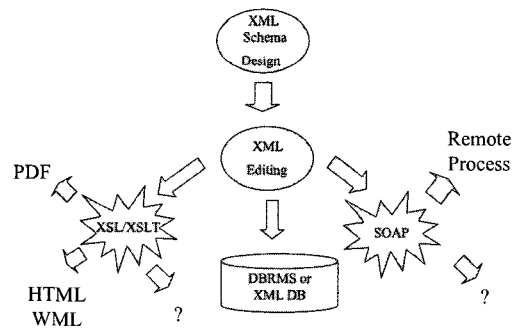
2.4. XML문서의 기술방법

XML 문서 내용을 기술하기 위해서는 두 가지 방법이 있다. 문स्ता입 선언(DTD, Document Type Declaration)과 XML 스키마이다. XML문서가 그 DTD 에 따를 경우 타입이 검증되었다고 말한다. 어떤 문서가 잘 구성되어 있기는 하지만 타입이 검증되지 않을 수 있는데 그 이유는 DTD 구조를 위반하고 있거나 또는 애당초 DTD를 갖고 있지 않기 때문이다. XML 스키마(Schema)는 XML 문서의 구조와 콘텐츠를 정의하는 파일을 가리키는 일반적인 용어이다. DTD도 이러한 스키마(Schema)의 일종이지만 많은 문제점을 가지고 있었다. DTD와의 가장 큰 차이점은 DTD는 EBNF라는 복잡하고 낯선 언어로 기술해야 하지만 XML 스키마(Schema)는 XML을 사용하여 기술한다는 점입니다. 또한, DTD에서 표현할 수 없었던 각종 데이터 타입과 엘리먼트 재사용 등을 XML 스키마(Schema)에서는 기본적으로 할 수 있다. 근본적으로 스키마(Schema) 파일의 역할은 DTD와 동일하다. 오히려 DTD 보다 발전된 형태라고 볼 수 있다. 훨씬 많은 데이터 유형을 표현할 수 있으며 네임스페이스(Namespace)를 사용하여 XML 태그들 간의 충돌을 해결할 수 있다. 또 DTD는 XML이 아니기 때문에 DTD 자체를 해석할 프로그램이 따로 존재해야 했지만, 스키마(Schema)는 XML 그 자체라는 것이 편리하다. XML 스키마는 XML 문서의 내용을 기술하기 위한 표준이다. XML 스키마는 개발자 자신만의 용

어를 정의하여 사용할 수 있도록 해준다. 주어진 XML 스키마에 일치하는 문서를 스키마 검증된 문서라고 말한다. DTD와는 달리 XML 스키마 문서는 그 자체가 또 다른 XML 스키마에 의해 검증될 수 있는 XML 문서가 된다. 이 최종적인 스키마는 W3C에 의해 관리된다. 스키마는 요소(element)와 속성(attribute)으로 구성된다. 요소에는 단순요소(simple element)와 복합요소(complex element)가 있다. 단순요소는 하나의 데이터 값을 갖는다. 복합요소 또한 속성을 갖고 있다. 일반적으로 요소는 데이터를 표현하기 위한 것이고 속성은 메타데이터를 표현하기 위한 것이다. 물론 이것은 일반 규칙을 말한 것이고 XML 표준에서 설명한 내용은 아니다. XML 스키마와 문서는 테이블보다 더 많은 구조를 표현할 수 있다. 전화번호(phone)나 주소(address) 같은 그룹이 정의될 수도 있다. 중복 정의를 회피하기 위해 요소는 전역적으로 정의된다. 중복된 정의는 변경될 필요가 있을 경우 하나만 변경되고 다른 것은 변경되지 않은 채 남아 있을 수 있어서 일관성이 깨어질 위험이 있으므로 바람직하지 않다.[8]-[11]

2.5. XML문서의 시각화

XML문서는 내용과 표현이 분리된 최상단의 추상화 문서이므로 XSL/XSLT(Extensible Markup Language Transformation)와 같은 스타일시트를 사용하면 웹은 물론 모바일기기에도 전송할 수 있다. <그림 1>은 XML문서의 시각화를 도시한 것



<그림 1> XML문서의 시각화

자료: Larry Kim지음, 유영배,조영임옮김, xmlspy따라하기, 홍릉과학출판사,2004, p.20

으로 XML문서는 XSLT에 의해 HTML이나 WML로 쉽게 변형될 수 있으며 SOAP에 의한 원격처리도 가능하다. 그림에서 ?는 XML문서를 또 다른 여러 형식으로 변환하거나 원격프로세스로 전송함을 표시한다.

그리고 XSLT외에 XML문서의 변환을 위한 응용프로그램 인터페이스로는 DOM과 SAX가 있다. XML 문서를 프로그래밍 적으로 처리하기 위해서는 일반적으로 각 요소(element)를 트리구조로 분리하는 파싱(parsing) 과정을 거쳐야 한다. 파싱한 자료를 트리구조로 분석, 저장하여 특정 요소에 대한 접근을 하게 되는데 이때 이용되는 API가 DOM이나 SAX이다. DOM이나 SAX를 이용하여 XML문서를 접근 할 경우 XML문서의 최상위 루트노드와 하위노드들 간의 관계와 구조가 명확히 이해되어야 정확한 프로그램이 가능하다. 유무선 인터넷 분야에서 XML이 적용되어 사용되고 있고 표준화가 완료된 상태이지만 구체적이고 상세한 트리 구조와 이의 접근 방법을 알기 위해서는 XML문법의 명확한 이해가 필요하다. XML 기술은 구조, 문법, 성격 면에서 기존의 절차적 프로그래밍 언어와 상당한 차이가 있다. 따라서 XML 애플리케이션 개발도 기존의 소프트웨어 개발과 다르다고 보는 것이 타당하다. XML애플리케이션 개발은 XML 문서들을 그 애플리케이션에서 이용하도록 규정하는 XML 스키마를 개발함으로써 시작된다. 그 다음에는 XML 스키마에 따라 XML 문서를 편집하고 그 유효성을 검사한다. 마지막으로 언어 바인딩(language binding)을 프로그래밍화하여 XML이 가능한 프레임워크로 XML 문서로 전환하고 처리할 수 있게 해야 한다. XML 문서 운용에는 주로 XML 문서를 또 다른 형식으로 변환하거나, XML 문서를 데이터베이스에 저장하거나 XML 문서를 원격 프로세스로 전송하는 일이 포함된다. XML을 개발하기 위해서는 XML 개발 도구를 능숙하게 이용할 수 있어야 한다. XML개발 도구에는 XML콘텐츠를 규정하는 XML스키마 모델링 도구, XML 문서를 만들어 내는 XML편집 및 유효성 검사 도구, XML 변환을 위한 XSL/XSLT개발 및 디버깅 도구, 데이터베이스 통합 애플리케이션을 위한 XML-데이터베이스 매핑 도구, 웹서비스를 확립하기 위한 SOAP 개발 및 디버

깅 도구 등이 있다.[12][13]

3. XML을 이용한 모바일 기술

모바일공간에서 통합적으로 사용될 수 있는 솔루션을 개발한다는 것은 매우 어려운 일이다. 최근 까지 모바일공급자들은 자신의 플랫폼과 언어들을 엄격히 구분해 왔다. 예를 들면, 노키아(Nokia)와 오픈웨어(Openware)는 WML과 WAP을 개발하여 북미지역에서 사용하였고 1999년에는 엔티티도코모(NTT DoCoMo)가 Compact HTML에 기반을 둔 i-Mode서비스를 일본에서 출시하였다. 모바일 공급자는 각각의 복잡한 플랫폼을 알아야 하는 전문가들이어야 했고 수백 가지의 기기에서 적용될 수 있는 콘텐츠를 개별적으로 만들어야 했다. 이러한 개발은 수 주 내지는 수개월이 소요되었고 이러한 프로젝트의 수행은 많은 비용이 수반되었다. 이러한 문제를 해결하기 위해 W3C는 업계표준을 지원하기 위해 플랫폼 독립적으로 상호동작이 가능한 XML기술을 개발하였다. 그리고 2001년 10월에는 XML과 HTML이 결합된 XHTML(Extensible HTML)이 소개되었다. XHTML은 NTT DoCoMo, Nokia, Ericsson, Openware와 같은 통신회사들의 지지를 받고 있고 여러 모바일기기를 위해 통합적으로 개발되었다. 이들 회사들은 공개모바일 연합(OMA: Open Mobile Alliance)을 구성하였다. XHTML은 XML기준의 멀티서버플래임워크로 XML이 HTML과 WML로 변환되듯이 XHTML은 하나의 코드 변경없이 WML로 변환할 수 있다.[14]-[18]

3.1. Mobile 인터넷과 XML

W3C가 1998년 XML 권고안을 발표한지 3년이 지난 2001년 5월에 XML을 새로운 인터넷 표준으로 인정하였다. 따라서 Gartner 그룹이 향후 10년간 IT시장을 주도할 12대 기술 가운데 하나로 선정한 XML은 mobile인터넷과 전자상거래의 핵심적인 기술로 사용되는 차세대 인터넷 언어로 확고히 자리 잡게 되었다. HTML이 메타언어가 아니라 메타언어를 통해 만들어진 애플리케이션이기

때문에 HTML을 mobile인터넷 등에 적용할 경우 웹 브라우저를 개발하는 측과 HTML관련 프로그램을 제공하는 측이 각기 상이한 HTML을 사용하여 브라우저 사이의 호환성 문제가 발생할 수 있다. 메타언어인 XML을 사용하게 되면 상기한 문제를 줄일 수 있는데 mobile인터넷의 경우 XML기반의 WML(Wireless Markup Language)을 사용하여 기존의 HTML계열의 인터넷 사이트에 접속하고 있다. XML을 이용하면 실제데이터와 프레젠테이션을 분리하는 새로운 사이트를 구축함으로써 기존의 유선인터넷과 무선인터넷을 통합하는 방법이 활성화되고 있다. 즉, XML을 이용하여 사이트의 기반 데이터들을 데이터베이스에 저장하고 사이트에 접근하는 클라이언트의 종류에 따라 적절한 XSL(eXtensible Stylesheet Language)로 단말기에 사이트 콘텐츠들을 제공하는 방안이 제공되고 있다. 물론 XML기반의 이러한 방식은 유. 무선인터넷 뿐 만 아니라 휴대폰에서 WebTV 등에 이르는 제반 PvC(Pervasive Computing) 단말기의 콘텐츠 제공에도 적용되고 있다. 그리고 J2ME의 MIDP 환경에서 동작하는 XML파서인 KXML이 Sun에 의해 개발되었고 Window CE플랫폼을 위해 pocketSOAP이 소개되어 SOAP과 XML이 모바일 기기에서 모두 가능하게 되었다. HTML과 WML로 사이트를 코딩하는 대신에 사이트를 XML로 코딩한 후 사이트를 접속하는 기기의 유형에 따라 XML문서를 XSLT와 XPath를 사용하여 HTML이나 WML의 적절한 마크업 언어로 변환하면 된다. 이상적으로는 웹브라우저이든 WAP기기이든지 모든 기기에 사용될 수 있는 하나의 마크업언어를 가질 수는 있다. 그러나 향후 몇 년 동안은 이것은 꿈에 지나지 않을 것으로 보인다. WAP 2.0 이 표준마크업언어로 XHTML을 선언하기 전까지는 개발자들은 여러 기기들과 자신들의 사이트에 맞는 메커니즘들을 별도로 개발하여야 했다. 오랫동안 하나의 표준 기술로 간주되어 온 XML은 최근까지 큰 호응을 얻지는 못하였으나 콘텐츠 코딩을 위한 완벽한 후보 언어로 보인다.[19]-[21]

3.2. XML-RPC와 SOAP

객체기반 분산컴퓨팅 환경의 ORB(Object Request

Broker)들은 개방형 mobile 서비스에서 상호운영성의 제약과 방화벽에 의한 메시지 차단 등 여러 가지 한계점이 드러나는데, 특히 데이터 저장이나 처리가 한정적인 palmOS나 Symbian핸드폰장치와 같은 다양한 OS로 운영되는 클라이언트와 서버들 간에 mobile 서비스를 할 경우에는 RPC(Remote Procedure Call)나 데이터 교환에 문제들이 발생한다. 즉, 객체기반 분산컴퓨팅하에서의 mobile클라이언트는 CORBA, DCOM, Java RMI등의 ORB 소프트웨어에 의존하여 서버와 데이터교환을 하기 때문에 데이터 저장이나 처리가 한정적인 mobile 클라이언트와 서버 간에 서비스를 할 경우에는 RPC나 데이터 교환에 여러 가지 문제가 내재하고 있다. 원격 프로시저를 호출하기 위한 프로토콜로 설계된 것이 아닌 HTTP(Hypertext Transfer Protocol)만을 사용하여 원격지 응용프로그램과 상호 작용할 경우에는 많은 문제가 있고 DCOM이나 IIOP(Internet Inter-ORB Protocol)등을 사용할 경우에도 방화벽에 의해 분산개체 프로토콜의 효과적인 사용이 난이 할 뿐만 아니라 특정회사의 제품에 편향될 수도 있기 때문에 일반적인 접근방법이라고 할 수 없다. 기존의 HTTP를 확장해 XML 메시지를 주고받을 수 있는 프로토콜로 XML-RPC와 SOAP(Simple Object Access Protocol) 등이 있다. XML-RPC를 개량한 SOAP은 클라이언트와 서버 모두에게 간편하고 편리한(simple and lightweight) 인터페이스를 제공하는 프로토콜이기 때문에 Service Oriented Access Protocol의 약어로도 사용하고 있다. SOAP은 XML-RPC와 마찬가지로 기본 전송수단으로 HTTP를 사용하고 호출 요청 및 응답 인코딩에 XML을 사용하는 메커니즘을 통해 클라이언트-서버간의 인터랙션을 한다. 이를 통해 인터넷 표준을 기반으로 하는 표준 개체 호출 프로토콜을 제공하고, 확장 가능한 프로토콜과 페이로드 형식을 만든다. 모바일응용프로그램은 분산컴퓨팅환경의 최종적 목표로 볼 수 있고 이들 클라이언트들은 수많은 운영체제에서 실행되고 어떤 곳이든지 위치할 수 있어야 하며 이러한 제약조건에도 불구하고 모든 모바일기기들은 데이터를 얻고 갱신하기 위해서는 서버들과 상호 동작할 필요가 있다. 진술한바와 같이 CORBA, DCOM, Java RMI와 같은 ORB기술들은 원격프로시저호출

이나 데이터교환에는 적합하지 않다. XML기반 기술인 XML-RPC와 SOAP은 이에 적합하도록 개발되었다. SOAP을 이용하면 서버로부터 XML스트림을 검색하고 XML의 연재(serialization)를 가능하게 하므로 모바일응용프로그램 개발을 간략하게 할 수 있다. 최근 Sun에 의해 개발된 J2ME환경에서의 KXML파서와 Window CE를 위한 pocketSOAP은 XML과 SOAP을 핸드폰에서 이용할 수 있게 하였다. 이와 같이 XML은 원격프로시저호출과 데이터교환의 수준을 높이고 유무선 기기를 아우르는 웹서비스의 새로운 세대를 구축하였다. [22]

3.3. XML문서와 RDB 매핑

XML문서에는 데이터 지향적 XML과 문서 지향적 XML이 있다. 데이터 지향적 XML문서는 주로 데이터베이스에 상주하는 트랜잭션데이터와 같고 문서 지향적 XML은 사람이 생성한 문서와 같은 출판데이터 형식으로 이들은 XML시스템에서 서로 다르게 처리된다. 데이터베이스는 데이터구조를 제한함으로써 빠른 접속과 처리를 가능하게 한다. 전형적으로 데이터베이스는 관계테이블로 구성되어 있고 각 테이블은 일련의 레코드들을 저장하고 있고 각 레코드는 일정의 필드들의 집합을 가지고 있다. 필드들은 다른 테이블들의 레코드들에 링크될 수 있으며 링크의 정도는 연결의 중첩 정도를 나타낸다. 그러나 테이블들의 전체 세트는 데이터베이스 스키마로 고정되어 있으며 동적으로 수정할 수는 없다. 따라서 주어진 데이터베이스는 고정 깊이를 가진다고 한다. XML문서나 XML문서의 각 엘리먼트들이 데이터베이스로 표현되기 위해서는 XML의 문서 구조가 관계테이블의 모델로 쉽게 매핑 될 수 있어야 함은 물론 매핑 자체가 아주 정확하게 예측가능(strongly predictable)하여야 한다. 정확한 매핑을 위해서는 XML문서의 각 엘리먼트들은 수량(quantity), 부품번호(item-number), 특성(description), 가격(price)등과 같은 데이터베이스테이블의 속성과 같이 데이터 특성이 정의되어야(constrained) 한다. 데이터베이스테이블로의 매핑이 가능한 엘리먼트는 데이터베이스테이블과 같은 형태의 폼으로 쉽게 시각화 할 수 있다. 즉,

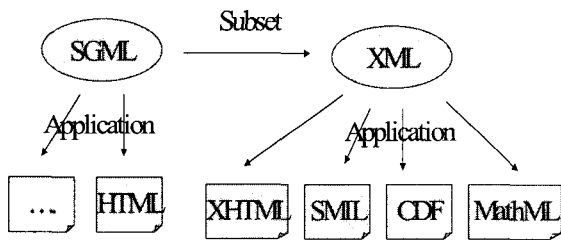
주문서와 같은 상거래용 문서는 일반 메모와 같은 문서보다는 더 정확하게 데이터베이스의 테이블로 매핑이 가능하므로 더욱 쉽게 하나의 폼으로 나타낼 수 있다. 일반적으로 XML 데이터는 DTD나 스키마에 의해 정의된다 하더라도 마치 데이터가 자유형식인 것처럼 여러 가지 변형이 가능하므로 데이터베이스데이터 보다는 훨씬 예측이 어려운 데이터이다. 예를 들면 한 엘리먼트 유형의 콘텐츠 모델은 여러 개의 서로 다른 하위 엘리먼트를 가질 수 있고 무순서로 여러 번 이용될 수도 있다. 또한 하위 엘리먼트는 상위 엘리먼트와는 다른 유형의 데이터가 될 수 있으므로 문서의 구조를 예측하기란 매우 어렵다. 게다가 이와 같은 콘텐츠모델의 문자데이터는 자신의 하위 엘리먼트들과 상호 혼재되어 혼합콘텐츠(mixed content)로 표현할 수도 있다. 혼합콘텐츠의 보기로는 문단 안에 강조를 위해 몇몇 단어들만 마크업(marked up)되거나 참조(reference) 또는 내장엘리먼트(in-line element) 유형으로 표시되는 문단을 볼 수 있다. 유효성의 측면에서 볼 때 혼합콘텐츠는 특별한 경우로 볼 수 있다. 왜냐하면 혼합콘텐츠에서는 어떤 엘리먼트 내에서 무슨 유형의 엘리먼트들을 사용할 것인지만 컨트롤을 할 수 있고 몇 번이나 어떤 순서로 사용할 것인지는 제어가 불가능하기 때문이다. XSLT의 측면에서도 혼합콘텐츠는 다소 다루기에 불편한 면을 가지고 있다. 왜냐하면 혼합콘텐츠를 가진 한 엘리먼트는 여러 개의 자식 문자 노드를 가지고 이들은 또 하나의 별개 노드가 되기 때문이다. XML의 장점 중 하나는 XML문서는 데이터베이스데이터와 출력데이터를 모두 가질 수 있고 문서처리와 데이터처리가 모두 수행될 수 있다는 것이다. 웹사이트는 주로 출력데이터 처리에 주목하고 있지만 동적사이트의 경우에는 데이터베이스 데이터도 처리할 필요가 있다. 그러나 XML기술은 데이터 유형에 따라 더 잘 작동하기도 하고 그렇지 않기도 한다. XML문서는 데이터베이스로부터 자동으로 생성될 수 있으며 데이터베이스용 데이터 또한 XML문서로부터 자동으로 생성이 가능하다.

4. XML문서 개발과 RDB 매핑 구현

본 연구에서는 지금까지 서술한 XML문서에 대해 Tagfree의 XML Developer를 이용하여 DTD, XML Schema, XSL등의 XML문서를 개발하고 MS SQL을 통하여 이를 매핑 하는 과정을 그래픽 에디터로 개발 구현하고자 한다.

4.1. XML문서 개발

2000년 10월에 제안된 XML1.0이 발표되어 지금까지 사용되어 오고 있으며 실제로 XML표준의 경우 SGML표준사양이 500페이지에 달하는 것을 단 20페이지로 표현할 수 있게 되었다. SGML과 HTML, XML의 관계를 보면 <그림 2>와 같다. XML은 문서의 내용에 관련된 태그를 사용하여 직접 정의할 수 있고 정의된 태그를 다른 사용자가 사용할 수도 있다. 따라서 XML은 본질적으로 다른 언어를 기술하기 위한 언어, 즉 메타언어 이다.

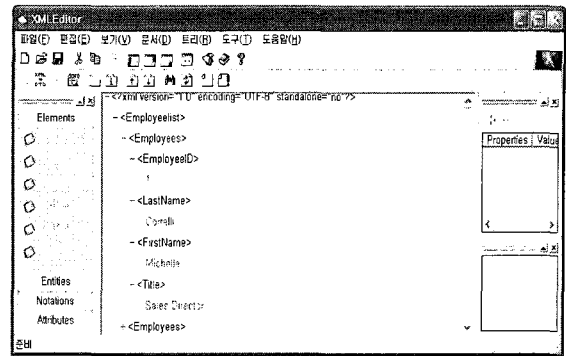


<그림 2> SGML/HTML/XML 의 관계

Tagfree의 XML Developer는 XML DTD에서부터 XSLT에 이르는 고급기술 및 E-Business에서 적용되는 XML의 다양한 응용력을 배가할 수 있는 솔루션으로 XML Editor, DTD Editor, XML Builder, X2X Mapper, DXD의 제품군으로 구성되어 있다.[23][24]

4.2. 정형문서의 작성

Tagfree XML Editor 은 XML 의 특징인 문서의 구조를 트리형식으로 제공함으로써 사용자로 하여금 쉽게 문서의 구조를 파악하고 편집할 수 있도록 한다. <그림 3>은 Employee.xml이라는 간단한 예제 파일을 정형문서(Well Formed Document)로 작성한 결과화면이다.



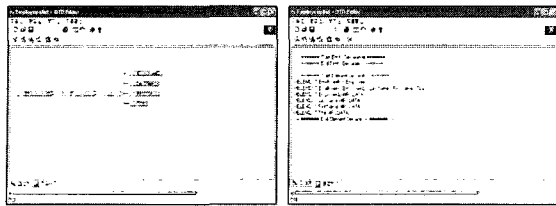
<그림 3> XML Editor로 작성한 정형문서의 결과 화면

XML문서의 각각의 Element 들에 대하여 속성의 표시, 텍스트 데이터, 하위 엘리먼트의 구조 등을 잘 나타내 주고 있고 현재 선택되어진 Element 는 파란색(Blue)으로 표시가 되며, 오른쪽 마우스 클릭 시 수행 가능한 명령들이 팝업 메뉴로 제공이 된다. 좌측의 .Element List Window 는 XML 문서가 DTD 를 포함하는 경우에 유용하게 쓰일 수 있고 DTD 에 정의되어 있는 요소(Element)들을 나열하여 XML 문서 편집 시, Element들을 표시할 수 있고, 또한 개체(Entity), 표기(Notation), 속성(Attributes) 들도 표시할 수 있다. 그리고 Element 추가 시 DTD 구조를 파악하여 하위 Element 들까지 자동적으로 생성시킬 수 있다. Element 는 종류에 따라서 속성 값을 가질 수 있다. 우측의 Element Properties Edit Window 에서 는 Element 가 포함하는 속성 값을 편집할 수 있다. Properties 에는 속성의 이름이, Values 에는 속성 값을 표시할 수 있다.

4.3. DTD의 설계

Tagfree DTD Editor는 DTD 문서 전문 편집기로서, 문서의 구조를 트리 형식으로 표현함으로써 XML문서의 구조를 쉽게 파악하게 한다. 텍스트 편집 창은 마치 윈도우즈의 메모장과 같이 직접 DTD 문서의 텍스트 원본을 편집할 수 있는 창이고 그리기 창은 문서의 구조를 트리 형식으로 보여준다. 사용자는 언제든지 텍스트 편집 창과 구조 편집 창(그리기 창)을 전환하면서 DTD 문서를 편집할 수 있다. <그림 4>는 Employeeslist.xml파일

을 위한 DTD설계로써 좌측은 그리기 창이며 우측은 텍스트 편집창이다.



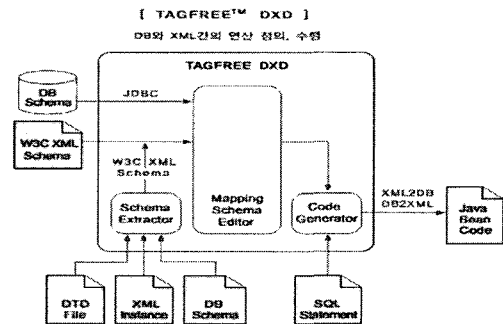
<그림 4> Employeeslist의 DTD설계와 텍스트 정의

새로운 노드를 선언하고자 할 때 DTD Editor의 개체 도구 바의 버튼을 선택하고 Structural View에서 노드를 생성하고자 하는 위치에 클릭하면 트리구조의 DTD파일을 생성할 수 있다. DTD Editor는 현재 선택된 DTD 노드의 속성 값을 편집할 수 있는 동일한 인터페이스로 속성 편집 창을 제공하며 속성 편집 창에는 현재 선택된 DTD 노드의 모든 속성 값이 리스트 컨트롤에 표시되며, 바로 속성 값을 편집할 수 있다.

4.4. Schema의 설계

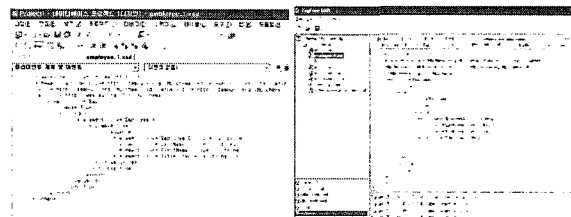
XML 문서의 구조와 데이터 타입과 같은 조건들을 정확히 염수하기 위해서는 XML문서에서 DTD 또는 Schema를 참조해야 한다. 이와 같은 XML문서를 유효한(Valid)문서 또는 정형(Well Formed)문서라고 한다. DTD를 사용하여 정형문서를 만들려면 DTD editor에서 작성한 DTD파일을 XML editor에서 이를 import하여 정형문서를 자동으로 생성하고 난후 XML editor에서는 데이터만 추가하면 된다. 이렇게 작성된 것이 <그림 3>의 DTD를 내장한 Employeeslist.xml파일이다. 여기서 앞서 설계해 놓은 DTD를 DXD 1.0을 이용하여 Schema로 변환시킨 후 변환된 Schema를 XML문서에 참조시켜 정형 XML문서를 생성하였다. Tagfree DXD는 W3C XML Schema와 관계형 데이터베이스 간의 데이터 변환을 위한 기능을 제공하는 도구로 첫째 W3C XML Schema를 추출할 수 있고, 둘째 XML Schema 구조를 이용하여 테이블을 생성할 수 있는 도구를 제공하며, 셋째 DB2XML과 XML2DB 변환을 위한 매핑 스키마를 작성할 수 있는 도구를 제공하며, 넷째

DB2XML과 XML2DB 변환을 수행하는 변환 엔진을 제공한다. 참고로 DXD의 구조를 보면 <그림 5>와 같고, 매핑 스키마란 XML과 DB간 변환 규칙을 정의한 파일을 말한다.



<그림 5> TagFree DXD의 구조도

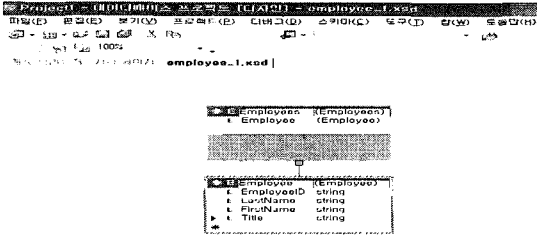
DXD는 관계형 데이터베이스의 데이터와 XML 문서내의 데이터를 시스템에 관계없는 유연한 데이터로 제공하기 위하여 설계되었다. DXD는 어플리케이션 개발 시 시스템에서 유통되는 데이터의 공통된 형태를 제공함으로써 새로운 시스템과의 통합, 구 시스템과의 호환 등의 장점을 제공한다. DXD는 데이터베이스의 구조와 XML문서의 구조 사이의 매핑을 제공함으로써 사용자의 뜻에 맞는 데이터의 형태를 구성한다. DXD를 이용하여 <그림 4>의 DTD파일을 Schema파일로 변환하면 <그림 6>과 같다.



<그림 6> DXD를 이용한 Employee Schema 파일

DXD를 이용하면 xml파일, dtd파일, Database Table들로부터 스키마파일을 만들 수 있다. <그림 7>은 Visual Studio.Net에서 작성한 employee_1.xsd 라는 스키마 파일의 그림이며 이는 <그림 4>의 DTD파일의 그림과 같다. <그림 4>의 DTD파일은 DXD에서 <그림 6>과 같은 스키마로 변경할 수 있다. 스키마로 변경이 되면 데이터베이스와

매핑이 이루어 질 수 있다.

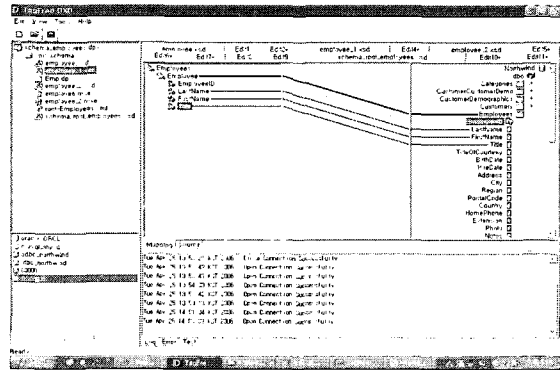


<그림 7> VS. NET을 이용한 XML 스키마 작성

4.5. 매핑 스키마 생성

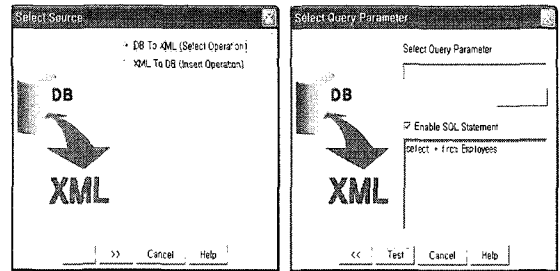
Schema Tree의 아이템들을 DB Tree로 드래그 앤 드롭 방식을 사용하여 매핑을 할 수 있다. XML Schema의 아이템은 성격에 따라 자식을 가지는 엘리먼트, 자식 엘리먼트, 애트리뷰트로 나뉜다. DB Tree의 아이템은 성격에 따라 데이터베이스, 사용자, 테이블, 일반 칼럼, 주키를 가지는 칼럼, 외래 키를 가지는 칼럼, 주키와 외래 키를 동시에 가지는 칼럼으로 나뉜다. 매핑은 Schema와 DB의 매핑 될 아이템들의 성격에 따라 매핑 형식이 나뉘게 되는데 테이블 매핑(데이터베이스의 테이블과의 매핑), 일반 칼럼 매핑(데이터베이스의 일반 칼럼과의 매핑), PK 칼럼 매핑(데이터베이스의 주키를 가지는 칼럼과의 매핑), FK 칼럼 매핑(데이터베이스의 외래 키를 가지는 칼럼과의 매핑), PKFK 칼럼 매핑(데이터베이스의 주키 이면서 동시에 외래 키를 가지는 칼럼과의 매핑), 관계 형 매핑(데이터베이스의 관계 형 칼럼과의 매핑으로 현재 지원 되지 않음)이 있다. 변환 엔진은 첫째 DB로부터 XML 도큐먼트(DB2XML)를 생성하는 것으로 매핑 정보, SQL문, DB 연결 정보를 바탕으로 DB에 있는 데이터를 XML로 추출한다. 둘째 DB로 XML 도큐먼트를(XML2DB) 저장하는 것으로 매핑 정보, SQL문, DB 연결 정보를 바탕으로 XML에 있는 데이터를 DB로 저장한다. 그리고 DXD에서는 개발 하게 될 시스템에 데이터베이스 테이블이 설계되어 있지 않을 때 제공되는 Schema의 구조에 맞는 테이블을 자동으로 생성할 수도 있다. Schema의 데이터 타입, 부모 관계를 바탕으로 테이블을 구성하고 칼럼의 데이터 타입

을 설정하면 된다. 매핑에는 Standard Mapping과 Relational Mapping이 있으며 전자는 매핑 될 아이템들의 부모를 설정해 주어 매핑하고 후자인 관계 형 매핑은 두 개 이상의 테이블과 매핑 될 때 각 테이블간의 관계를 유지하면서 이루어지는 매핑을 의미한다. <그림 8>은 <그림 7>의 employee스키마를 MS SQL 데이터베이스에 있는 NorthWind 데이터베이스와 매핑 한 것으로 NorthWind 데이터베이스의 Employees 테이블의 18개 필드 중 employee스키마와 이름이 일치하는



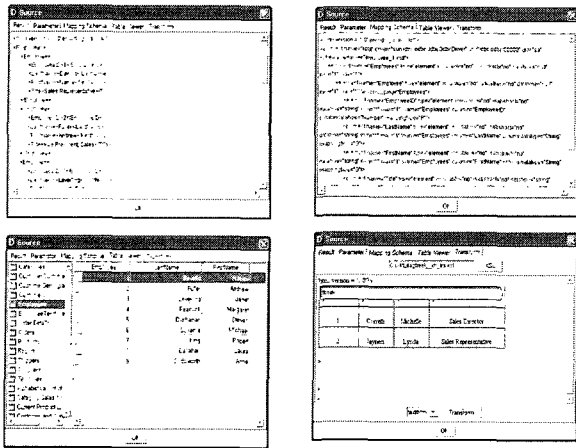
<그림 8> Employee스키마와 노스윈드 DB와의 매핑

4가지 필드를 매핑 하였다. 매핑은 좌측의 employee스키마의 각 필드를 우측의 NorthWind의 해당 필드로 드래그 하여 드랍하면 된다. 매핑 파일을 employee_1.mse로 저장한 후 이를 오른쪽 마우스로 클릭하여 preview를 선택하면 <그림 9>의 좌측 화면이 먼저 뜬다. 이는 데이터베이스와



<그림 9> 매핑 Preview와 Query 선택

XML문서의 변환 옵션이며 우측은 연결테이블의 필드를 선택하는 Query화면이다. <그림 10>은 매핑변환의 결과 화면으로 employee XML스키마에

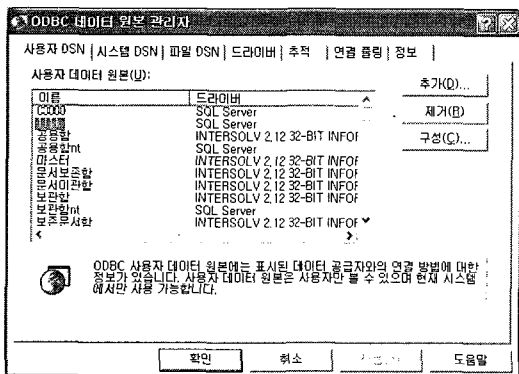


<그림 10> employee스키마-NorthWind DB 매핑의 결과 화면

해당하는 NorthWind 데이터베이스의 employees 테이블의 해당 자료들이 XML파일로 변경된 결과를 보여주고 있으며 매핑 시 사용된 ODBC driver, url, uid, schema filename에 대한 정보를 제공하며 결과 화면의 Table View에서는 NorthWind 데이터베이스의 전체 테이블들의 목록과 입력된 자료들을 볼 수가 있다. 결과화면의 Transform 에서는 XSL을 조합하여 XML문서를 HTML파일로 변환할 수 있다.

4.6. ODBC 설정과 JDBC 연결

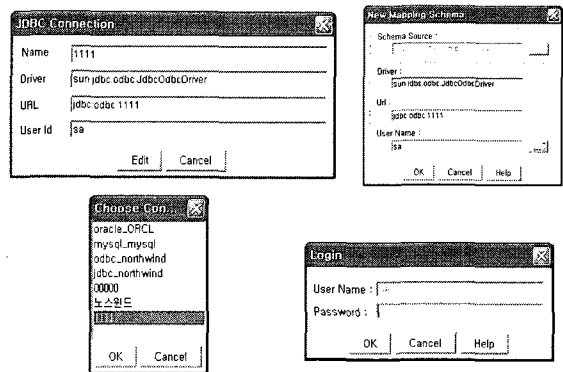
본 연구에서는 매핑스키마를 생성하기 위한 전제 조건으로 우선 MS SQL을 설치하고 ODBC드라이브를 설정하였다. 이는 윈도우XP의 제어판-데이터원본-ODBC데이터원본관리자-사용자DSN-추



<그림 11> SQL Server ODBC 드라이브 생성

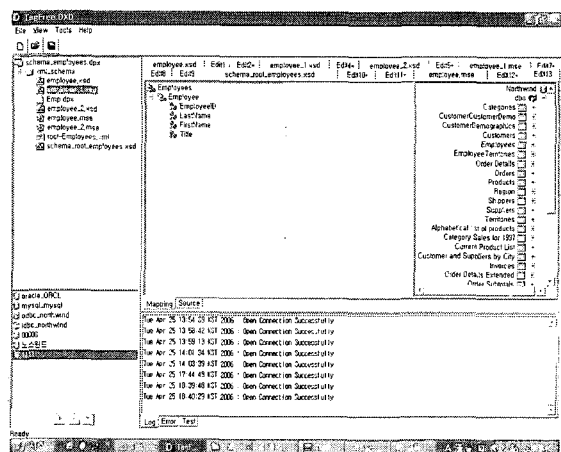
가-SQL Server-SQL Server에 새로운 데이터원본만들기(이름:1111, 서버:local)-사용자가 입력한 로그인 ID및 암호를 사용하는 SQL Server 인증 사용(로그인 ID: Sa 암호: 1234)-기본데이터베이스를 NorthWind로 선택하면 1111이란 이름의 SQL Server의 ODBC 드라이브가 <그림 11>과 같이 생성된다.

ODBC드라이브가 설정이 되면 DXD에서 JDBC 연결을 해 주어야 한다. DXD의 맨 좌측아래의 JDBC Connection Window에서 +의 Add 버튼을 누르면 JDBC Connection 화면이 나타나는데 해당 정보를 <그림 12>의 맨 좌상화면과 같이 입력하고 1111이라는 JDBC연결 드라이브를 만든 후



<그림 12> DXD에서의 JDBC연결과 매핑 설정

employee_1.xsd파일을 프로젝트윈도우에서 선택하여 오른쪽 마우스를 클릭하여 new mapping

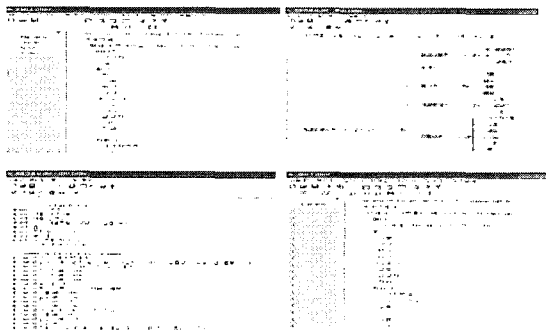


<그림 13> employee스키마-Northwind 디비의 매핑 초기 화면

schema를 선택하면 <그림 12>의 우상 측의 화면이 뜬다. 여기에서 <그림 12>의 좌하 측과 같이 JDBC 드라이브를 1111로 선택하면 <그림 12>의 우하 측과 같은 Login화면이 나타난다. ID와 MS SQL의 데이터베이스 암호를 입력하면 <그림 13>과 같은 매핑초기 화면이 형성된다. 여기에서 왼쪽의 employee 스키마트리에서 오른쪽의 데이터베이스 트리로 해당 필드를 드래그 하여 연결하면 <그림 8>의 완성된 매핑 화면이 나타난다.

4.7. Tagfree XML 개발툴을 이용한 실무사례

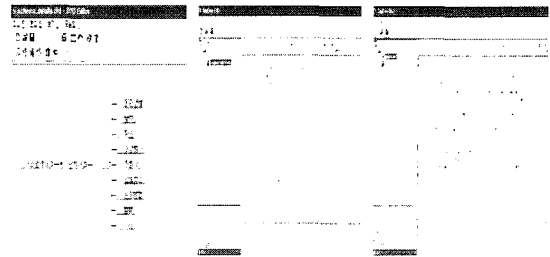
Tagfree XML개발툴은 이미 상용화 되어진 개발툴로써 XML활용을 쉽게 해준다. 상용화 된 개발툴로는 xmlspy, visualstudio.net 등 여러 가지가 있으나 XML의 중요성에 비해 실무적 활용이 다소 부족한 것 같다. Tagfree XML개발툴은 XML을 쉽게 이해하여 DTD문서와 XML문서를 쉽게 작성하고, XSL/XSLT를 이용한 XML의 프레젠테이션 방법과 XML자체만이 아닌 JAVA/DOM을 이용한 XML프로그래밍 테크닉을 제공하고 있으나 실무적 활용을 위한 매뉴얼이나 활용서가 충분하지 못해 응용사례개발의 어려움이 있다. Tagfree XML개발툴은 XML빌드, DTD설계, 스키마제작, XSLT등의 편집기들이 개별적으로 제공되어서 복잡한 XML문서와 RDB매핑의 각 단계를 쉽게 이해할 수 있는 장점이 있다. Tagfree XML개발툴은 XML문서에 대한연구용으로는 활용이 되고 있으나 상업성은 다소 떨어져 실무적 활용은 활발하지 않은 것 같다. 무선인터넷 기술의 급속한 발전으로 HTML기반의 compactHTML이나



<그림 14> 거래명세서XML문서편집, DTD그리기, DTD소스, DTD내장XML그리기

XHTML등의 해석기가 핸드폰 등에 탑재되어 이용되고 있는 것도 이의 이용을 지연시키고 있다.

본 장에서는 실무적 활용의 응용시스템의 사례로 거래명세서문서를 XML문서로 편집하고 복잡한 DTD문서를 Tagfree XML개발툴의 DTD그리기 기능을 이용하여 생성해 보았고 이에 따르는 DTD소스와 이를XML문서에 내장하여 DTD의 엘리먼트별로 편집이 가능한 그리기 화면을 구성해 보았다. <그림 14>는 이의 결과화면으로 네 가지 화면으로 구성되어 있으며 시계방향으로 제시되었다.



<그림 15> 거래명세서스키마그리기, DXD생성, 스키마소스보기

<그림 15>는 RDB매핑을 위해 거래명세서스키마를 재설계한 것으로 XML스키마에서는 DTD와 같이 Entity가 사용되지 않으므로 Entity선언이 생략되어서 DTD그림보다는 간단하다. DXD생성은 XML파일이나 스키마파일에서 생성될 수 있으며 이를 RDB로 매핑하면 된다. 본 연구에서는 Employee.xml 이라는 XML문서를 Tagfree XML개발툴과 노트패드와 같은 일반편집기로 프로그래밍 하여 양자를 동시에 개발하여 이의 차이점과 일치 관계를 규명해 봄으로써 프로그래밍 코딩 시에 발생할 수 있는 문법적 오류를 감소시키고 거래명세서와 같은 복잡한XML문서와 대형데이터베이스 파일의 구조와 연결 관계도 쉽게 이해 할 수 있었다. 그리고 SQL Server, Access, Oracle과 같은 데이터베이스와 XML문서의 매핑구조와 각 에트리뷰트의 설정관계도 파악할 수 있었다. XML파서는 XML 문서의 각 엘리먼트를 계층적 트리구조로 파싱하는데 데이터베이스 테이블이 XML문서로 변환되었을 때의 파싱 결과에 대한 문법적 구조와 차이점을 이해함으로써 개발툴이 있을 때나 없을 때 모두 프로그래밍 코딩을 이용하는 방안을 제시하였다.

5. 결 언

XML은 유무선 웹 콘텐츠에서 애플리케이션 데이터에 이르기까지 모든 종류의 데이터를 구성하는 가장 대중적인 형식이 되었다. 응용서버는 XSL이나 DOM을 이용하여 유무선 장치가 쉽게 읽을 수 있는 페이지로 변환할 수 있다. XML Style Language Translation (XSLT)를 사용하는 경우 응용서버는 XML콘텐츠와 XSLT 스타일을 연결하기 위해 SAX라는 파서(parser)를 이용하여 이를 페이지로 표현한다. 반면 DOM(Document Object Model)을 이용하는 경우 응용서버는 DOM을 XML 데이터에 삽입(plug-in)하여 유무선 장치에 필요한 HTML, XHTML, cHTML, WML과 같은 마크업 페이지로 변환시킨다. XSLT는 해독하기는 쉬우나 프로그램을 만들기가 어렵고 반면 DOM은 언어 중립적이고 전체문서를 트리구조로 메모리에 적재하여 사용하므로 XML문서구조를 이벤트기반으로 파악하여 사용하는 SAX나 XSLT에 비해 프로그래머가 처리할 부분이 적다. 프로그램을 통하여 XML문서에 접근할 수 있는 방법은 DOM이나 SAX(Simple API for XML)와 같은 API를 통하여 C, C++, JavaScript, Java, VBScript, C# 등과 같은 언어로 문서를 다룰 수가 있다. 내용과 스타일이 분리된 XML문서를 프로그래밍언어로 접근하기 위해서는 XML문서의 자료구조적 개념을 명확히 이해하고 있어야 하며 데이터베이스와의 연동을 통한 XML문서 처리를 위해서도 이러한 개념의 이해는 필수적이라 할 수 있다. [25][26][27] XML은 이미 90년 후반부터 지금까지 무수히 많은 개발툴이 소개되어 왔으며, XML의 특성을 비롯하여 장단점, 개발 프로세스, 타 시스템과의 호환성, XML개발의 어려움과 해결방안이 제시되었다.[28] 산업계에서는 산업 부분별로 XML을 응용할 수 있는 산업-상세 표준(Industry-Specific Standards)을 만들었으며 이 산업 표준들은 모두 XML 스키마나 DTD에서 사용되어 정의된 콘텐츠 모델을 가지고 있다. 산업분야별로 보면 문서, 과학, 출판, 금융, 생활과학, 인적자원 등이 있으며 문서 분야에 속하는 WML의 예를 들면[29] WML은 정보의 빠른 전송과 모바일 사용자의 서비스를 위한 XML용어이며 핸드폰

과 같은 무선의 디지털 장치와 호출기, 라디오, 스마트폰, palmOS와 EROC, Window CE, FLEXOS, OS/9, JavaOS 등의 운영시스템으로 만들어진 통신기기들에서 사용될 산업표준에 포함된다. 비즈니스 분야에서는 eb-XML이 표준화되었고, XML-EDI, WIDL(Web Interface Definition Language)[30] BIPS[31] 등 크고 작은 프로젝트에서 개발프로세스, 타 시스템과의 호환성, XML개발의 어려움 등의 내용들이 표현되었다. 본 연구에서는 XML문서를 자료 구조적 관점에서 이해하고 프로그램 처리나 데이터베이스 처리에 있어 반드시 알아야 할 데이터아일랜드, 루트엘리먼트, 엘리먼트, 속성, 데이터바인딩, XML 문서의 트리구조 변환, DTD, 스키마 등을 Tagfree의 XML 개발툴을 이용하여 직접 실현해 보았고 따라서 향후의 MS SQL이나 Oracle을 이용한 XML 문서처리나 DOM이나 SAX를 이용한 유무선 프로그램 처리에 도움이 될 수 있을 것이다.

참고문헌

- [1] Michael Yuan Juntao, *Enterprise J2ME: Developing Mobile Java applications*, Prentice Hall PTR. pp.285-304, 2004
- [2] Mark Beaulieu, *Wireless Internet: Applications and Architecture*, Addison Wesley. pp.431-446, 2002
- [3] Micab Dubinko, "XForms Essentials," O'reilly & Associates. ISBN 0596003692 Available online at www.dubinko.info/writing/xforms, 2003
- [4] Leslie Lamport, *LaTeX : User's Guide and Reference Manual*, Addison Wesley, 1994
- [5] Michel Goossens, and Sebastian Rahtz, *The LaTeX Web Companion: Integrating TEX, HTML, and XML*, Addison-wesley, 1999
- [6] Michel Goossens, Frank Mittelbach, and Alexander Samavin, *The LaTeX Companion: Includes Newly Revised LaTeX Standard*, Addison Wesley, 1994
- [7] 송정길저, XML 정복을 위한 지름길 XML프

- 로그래밍, 생능출판사, pp.16-19, 2004
- [8] David Hunter 외 5인 공저/윈도우사용자그룹 (안성욱, 공준혁, 천주석)공역, *Beginning XML*, 정보문화사, 2001
- [9] Richard Blair 외 12인 공저/윈도우사용자그룹 (안성욱, 최효진, 최기린, 최영규)공역, *Professional ASP XML*, 정보문화사, 2001
- [10] 고범석, 공정석, 권용재, 임수미, 윤수진공저, 초보자를 위한 다이내믹 웹설계 XML, 영진.COM, 2002
- [11] Ama Formica, "Legality of XML-Schema Type Hierarchies", *The Computer Journal*, Vol. 47, No.5, pp.591-601, 2004
- [12] Kirsanov Drinity *XSLT 2.0 Web Development*, Pearson Education Inc., 2004
- [13] Travis, Brian E. *XML and SOAP Programming for BizTalk Server*, Microsoft, 2000
- [14] Tomohisa Kamada, Takuya Asada, Masayasu Ishikawa and Shinichi Matsui, "HTML 4.0 Guidelines for Mobile Access," *W3C Note*, 15 March 1999 Available at <<http://www.w3.org/TR/NOTE-html40-mobile/>>
- [15] T. Kamada, "Compact HTML for Small Information Appliances", *W3CNote*, 9 February 1998 available at <<http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/>>
- [16] Jean-Loc David, "Developing Wireless Content Using XHTML Mobile," April 14 2004 available at <<http://www.xml.com/pub/a/2004/04/14/mobile.html>>
- [17] Wei Meng Lee, "An Introduction to XSLT and XPath", available at <<http://www.wirelessdevnet.com/channels/wap/training/xslt.html>>
- [18] Wei Meng Lee, "Transforming XML into WML", available at <http://www.wirelessdevnet.com/channels/wap/training/xslt_wml.html>
- [19] 이봉규, "XML기반의 Mobile기술에 관한 연구," 정보처리학회지 제8권 제3호, 2001년 5월, pp.54-60
- [20] C. F. Chen "The Connection between Mobile Communication and Information Access: a Study of the Wireless Application Protocol(WAP) and XML", *Journal of Information, Communication, and Library Science*, v.7 no.3 pp.93-104, 2001
- [21] Tzu-Han Kao and Shyan-Ming Yuan, "Designing an XML-Based Context-Aware Transformation Framework for Mobile Execution Environments Using CC/PP and XSLT", *Computer Standards & Interfaces* 26, pp.377-399, 2004
- [22] Delier WDN, "Mobile services Using XML and SAOP", February 8, 2001 available at <<http://www.wirelessdevnet.com/columns/feb2001/editor14.html>>
- [23] 강익태, 김광휘, 오선진 공저, XML Programming, Com & Book, 2003
- [24] (주) 다산기술부설기술연구소공저, XML 이론과 실무활용, DASAN Technology, 2002
- [25] Rawlins, Michael C., *Using XML with Legacy Business Applications*, Addison Wesley, 2004
- [26] Deitel, H. M., Deitel, P. J., Nieto, T R., Lin, T. M. and P. Sadhu, *XML How to Program: Java 2, perl/CGI and Active Server Pages, 피어슨에듀케이션코리아*, 2002
- [27] 권해운저, XML & Java for Webmaster, 성안당, 2002
- [28] XML Journal available at <<http://xml.sys-con.com/read/filter/xmllatest.htm>>
- [29] Open Mobile Alliance(former wapforum) available at <<http://www.wapforum.org/>>
- [30] Web Interface Definition Language available at <<http://www.w3.org/TR/NOTE-widl-970922>>
- [31] Financial Service Technology Consortium available at <<http://www.fstc.org/projects/bips/>>



우 원 택 (Won-Taek Woo)

- 1979년2월 성균관대학교통계학
과학사
- 1981년8월 연세대학교경영대학
원 석사
- 1991년2월 영남대학교상경대학 박사
- 1993년3월 ~ 현재 대구한의대학교 모바일콘텐츠학부
- 관심분야 : MIS, 모바일상거래, 모바일게임