# 시각적 XML 문서 편집기 구현

황현숙[0] 오지훈[1] 최병규[2] 전양승[3] 한성국[3]
원광대학교 컴퓨터공학과 시맨틱웹 연구그룹[0,2,3]
(주)유엠텍 종합기술연구소 선임연구원[1]
{hhs2103[0], opt[1], nix-j[3], skhan[3]}@wonkwang.ac.kr, nowwhere@hotmail.com[2]

# An implementation of Visual XML Document Editor

Hyeun-Sook Hwang[0] Ji-Hoon On[1] Byung-Gyou Choi[2] Yang-Seung Jeon[3] Sung-Kook Han[3]
Semantic Web Research Group, Dept. of Computer Engineering, Wonkwang University[0,2,3]
Senior Researcher ,Total Technique Research Institute, u-MTec Co. Ltd.[1]

**요 약**

XML은 어떠한 정보든지 구조화할 수 있으며, 데이터와 스타일 정보를 철저히 분리함으로써 컴퓨터가 처리할 수 있는 메타데이터를 제공하여 정보검색의 정확성을 높이는 획기적인 계기를 마련하였다. XML 데이터와 스타일의 분리를 위해 XML 기반의 문서를 표현하기 위한 스타일시트인 XSLT가 제안되었지만, 이는 전문적인 지식이 없는 일반인이 사용하기에 매우 어려울 뿐만 아니라, 작성하는데도 많은 시간과 비용이 필요하다. 이에 대한 대안으로 현재 XSLT 문서 편집 시스템이 상용화되어 출시되고 있지만, 사용이 너무 복잡하고 어려워 문서 편집 시스템의 요구조건을 충족하기에는 미흡하다. 이에 본 논문에서는 일반인도 쉽게 XML 기반의 문서 구조와 스타일시트를 생성할 수 있는 시각적 환경 기반의 XML 기반 문서 편집 시스템을 구현하고, 이를 실제 적용하여 기존 시스템과의 성능 평가를 통해 시스템의 성능을 검증하였다. 또한 문서 구조에 따른 데이터베이스 테이블을 동적으로 생성하여 XML 기반 문서 관리의 효용성을 다각도로 제시하였다.

## 1. 서 론

현재의 웹 기술의 주류를 이루는 마크업 언어(Markup Language)인 HTML은 많은 단점이 있다. 첫째, 사용할 수 있는 태그(Tag)가 고정되어 있고, 사용자가 필요로 하는 태그를 정의할 수 없다. 둘째, 문서의 구조 형식을 표현하는 태그와 표현형식을 지정하는 태그가 혼합되어있다. 셋째, HTML의 태그는 문서의 외형적인 모양을 지정하는 스타일시트(Stylesheet)의 성격 위주로 구성 되어있어 조직구성의 표현이 제한된다[1,3,4].

이와 같은 HTML의 한계점을 극복하기 위하여 XML이 등장하였다[2]. XML[8]은 데이터(Data)와 스타일(Style)이 분리되어 스타일시트에 따른 다양한 결과물을 만들 수 있으며, 사용자 정의 의미태그 정의를 통한 메타데이터(Meta Data) 특성을 갖는다. 이와 같은 XML의 특징은 현재 단순한 문서나 정보의 구조화뿐만 아니라 독특한 구조를 갖고 있는 정보 모델링에 널리 이용되고 있으며 그 활용 범주는 급격히 확대되고 있다[5,7].

XML의 핵심적인 특징인 데이터와 스타일의 분리는 순수 데이터의 측면에서는 매우 유용하나 사용자 중심의 출력을 하기 위해서는 XML 표현 언어인 XSLT[6,9]에 대한 전문지식과 함께 많은 시간과 비용을 필요로 한다는 어려움이 있다. 이런 문제점을 해결하기 위해 현재 많은 XSLT 에디터가 개발되었지만 여전히 그 사용은 지나치게 복잡하고, 실용적인 활용에 문제가 있다.

이에 본 논문에서는 XSLT에 대한 전문적인 지식이 없는 일반인도 쉽게 데이터베이스와 상호연동이 가능한 XML 기반의 문서를 직접 설계하고 제작 할 수 있는 XML 기반의 문서 편집 시스템을 구현한다.

## 2. XML 문서 편집기 개발 현황

현재 상용화된 XML 문서 편집기능을 제공하는 많은 XSLT 에디터가 있지만 그 중 대표적인 XML 문서 편집 기술사항을 분석하면 다음 표 1과 같다.

표 1. XML 문서 편집 시스템 분석

| 구 분 | 장 점 | 단 점 |
|---|---|---|
| XML SPY STYLEVISION | •WYSIWYG 방식에 의한 화면 편집과 다양한 스타일 편집 가능<br>•템플릿기반 화면정보 생성<br>•데이터베이스 스키마를 이용한 데이터 출력 | •기능이 복잡하여 사용자가 작업하기 어려움<br>•XML Schema 에디터와 XSLT 에디터의 이원화 통합 개발 환경을 제공치 않음. |
| Whitehill <xsl> Composer | •WYSIWYG 방식에 의한 화면 편집과 다양한 스타일 편집 가능<br>•템플릿기반 화면정보 생성<br>•화면구성 정보 출력으로 화면 편집용이 | •XML 문서 편집을 위한 통합 환경을 제공하지 않는다. (XSLT 에디터 기능만 제공)<br>•외부 시스템과의 상호 운용 불가능 |
| 향후 연구방향 | •일반인도 쉽게 사용할 수 있는 쉬운 인터페이스 제공을 통한 일반화 유도<br>•XML Schema 에디터와 XSLT 에디터 제공으로 통합 개발 환경 제공<br>•데이터베이스와 상호 운용 가능한 편집 시스템 제공 | |

Altova의 XML 스타일시트 편집기인 STYLEVISION은 XML SPY 또는 기타 Schema 에디터로 생성된 Schema 파일 또는 데이터베이스의 Schema를 분석하여 정해진 템플리트에 선택적으로 매칭시켜 속성정보를 변경함으로써 스타일 편집을 가능하게 한다.

Whitehill <xsl> Composer는 원본 XML 파일을 로드하여 트리뷰로 구성하고 프레임이나 테이블을 이용하여 WYSIWYG방식으로 기본 화면을 설계한다. 그리고 원본 XML 파일을 노드단위로 Design 창의 출력할 위치에 Drag & Drop으로 위치시키고 속성정보를 추가 변경함으로써 스타일을 편집한다.

## 3. XML 기반의 문서 편집 시스템 설계

### 3.1 개념적 설계

본 논문에서 구현한 시스템은 다음의 세 가지 기능을 제공한다. 첫째, 전자문서의 구조 생성을 위한 XML 기반의 Schema 에디터를 제공한다. 둘째, 생성된 Schema 파일을 웹 페이지에서 보여주기 위한 XSLT 에디터 기능을 제공한다. 셋째, SQL 서버와의 상호연동이 가능하다. 각 기능별 업무흐름은 다음과 같다.

Schema 에디터는 만들고자 하는 전자문서의 구조를 이루는 XML 기반의 Schema 파일을 WYSIWYG 방식으로 쉽게 생성하는 기능을 제공한다. 이때 생성되는 Schema 파일은 문서의 레이아웃과 데이터베이스 테이블 생성을 위한 기본정보로 사용되어야 하기 때문에 Schema 파일은 Table과 Column 요소로 구성된다.

XSLT 에디터는 Schema 에디터에 의해 만들어진 Schema 파일을 로드하여 패턴 분석을 통해 기본 레이아웃을 디자인 창에 출력한다. 그 후 사용자에 의해 WYSIWYG방식으로 디자인 편집이 이뤄진다. 디자인 편집은 속성정보의 변경에 의해 이뤄지며 수행절차는 요소 선택후 해당 요소에서 속성정보를 변경한후 가능하다.

Schema 정보를 통한 SQL 서버에 Table을 생성을 하기 위해서는 먼저Schema 파일이 로드된 상태에서 데이터베이스 테이블 생성 기능을 선택함으로써 가능하다. 이때 시스템은 SQL 서버 연결을 위해 서버 및 데이터베이스 이름, 사용자 아이디와 패스워드를 요청하며 사용자 인증후 데이터베이스에 Table을 생성한다.

### 3.2 논리적 설계

#### 3.2.1 Object Interaction Diagram

객체추출시 도출된 클래스를 바탕으로 각 시나리오의 업무절차별 이벤트 처리를 검증함과 동시에 서비스를 통해서 클래스간의 상호연관 관계를 도출한다(그림 1).

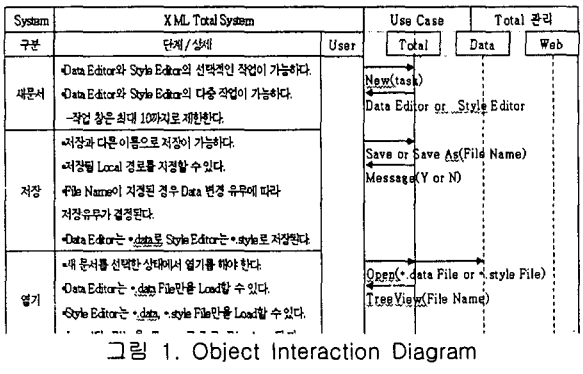


그림 1. Object Interaction Diagram

#### 3.2.2 Class Diagram

Object Interaction Diagram을 통해서 클래스 간의 연관관계와 클래스의 구성 속성 및 메소드가 파악되면 전체 클래스간의 상호 연관관계 즉 상호작용, 일반화, 포함 관계를 구성한다. 그림 2는 Class Diagram의 일부이다.

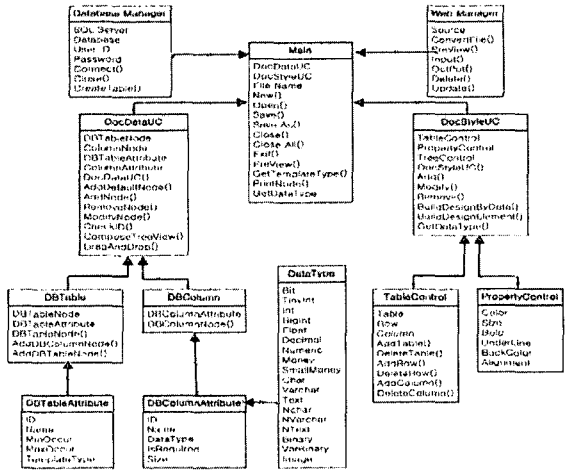


그림 2. Class Diagram

## 4. XML 기반의 문서 편집 시스템 구현

### 4.1 Schema 에디터

전자문서의 구조적 설계를 위해 Schema 에디터를 선택하게 되면 동적으로 TabPage가 생성되며 그 위에 Schema 에디터 사용자 정의 컨트롤이 위치하게 된다. Schema 에디터의 구성은 전자문서의 구조를 계층적으로 보여주기 위한 TreeView 컨트롤과 문서구성 요소인 Table과 Column을 선택적으로 사용하기 위한 ToolBox 컨트롤, 그리고 각 요소에 대한 속성정보를 추가, 수정, 삭제하기 위한 PropertyGrid 컨트롤, 마지막으로 노드를 추가, 수정, 삭제하기 위한 Button 컨트롤로 구성되어 있다. 아래의 그림은 Schema 에디터를 보여준다.

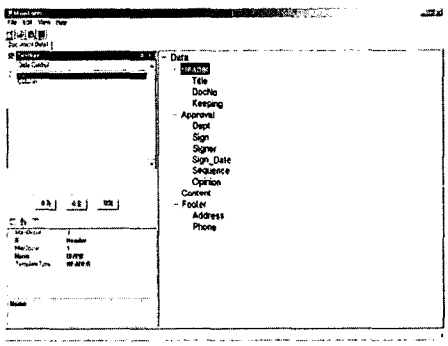

그림 3. Schema Editor

전자문서의 구조적 설계를 위한 Schema 생성은 다음과 같은 원칙을 갖는다. 첫째, 노드는 테이블과 컬럼 노드로 구성된다. 둘째, 모든 노드는 식별을 위한 고유한

ID를 가져야 한다. 셋째, 테이블 노드 하위에는 모든 요소가 자식으로 추가될 수 있지만 컬럼 노드는 하위노드를 가질 수 없다. 넷째, 요소가 동일할 경우 이름 속성이 같아서는 안 된다. 그리고 마지막으로 루트 노드는 삭제될 수 없으며 노드 추가시 노드 선택이 선행되어야 한다.

### 4.2 XSLT 에디터

Schema 에디터로 전자문서의 구조 설계를 완성한 후 전자문서의 화면 레이아웃을 생성하기 위해 XSLT 에디터를 선택하게 되면 동적으로 TabPage가 생성되며 그 위에 XSLT 에디터 사용자 정의 컨트롤이 위치하게 된다. XSLT 에디터의 구성은 크게 로드된 Schema 정보를 보여주기 위한 TreeView 컨트롤과 각 노드에 대한 상세 속성정보를 보여주기 위한 ListBox 그리고 화면 디자인 정보를 보여주고 편집하기 위한 TextBox와 디자인 속성 변경을 위한 PropertyGrid로 구성된다(그림 4).

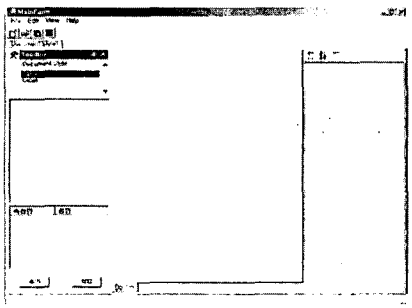

그림 4. XSLT Editor 구현결과

XSLT 에디터는 사용자 편의를 제공하기 위하여 Schema 에디터로 생성한 *.data 파일 로드시 Table과 Column 요소의 종첩관계를 분석하여 기본 레이아웃을 디자인 창에 출력한다. 또한 Schema 정보를 TreeView로 구조화하여 보여주고 ListBox를 통해 속성정보를 제공함으로써 전체적인 문서 정보를 한 눈에 파악할 수 있어 보다 효율적인 전자문서 생성이 가능하다.

Schema 파일 로드 후 시스템이 제공하는 기본 레이아웃 정보를 바탕으로 사용자는 드래그 앤 드롭 또는 속성정보 변경을 통해 WYSIWYG 방식으로 레이아웃을 쉽게 편집할 수 있다. 시스템이 제공하는 스타일 속성정보는 전자문서에 있어 필수 속성정보인 폰트와 정렬 그리고 색상정보이며 차후 기능 확장이 용이하도록 클래스화 되어 있다. 화면 디자인이 완료된 후 파일 저장 시 내부 프로세서에 의해 웹 페이지에 보여주기 위해 HTML 파일 생성을 위한 XSLT 스타일시트가 만들어진다.

### 4.3 Table 에디터

본 시스템에서 생성된 Schema 파일은 Schema 에디터에서 문서 구조 생성 시 요소와 속성정보를 데이터베이스 Schema 정보와 서로 일치시켜 생성했기 때문에 Schema 파일을 구성하는 요소와 속성정보의 분석을 통해 설계된 문서 구조를 그대로 반영한 데이터베이스 테이블 생성이 가능하다. 이 기능을 그룹웨어의 전자결재 시스템에 도입할 경우 전자문서의 생성과 동시에 데이터베이스 테이블을 만들 수 있어 개발속도 향상과 함께 비용절감 효과를 얻을 수 있다. 아래의 그림은 Schema 정보를 이용한 테이블 생성과정을 보여준다.

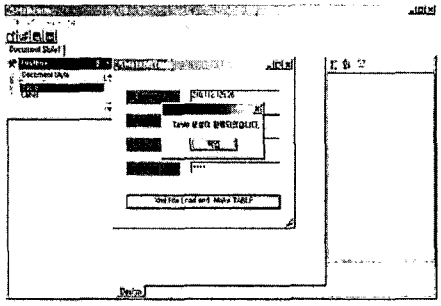

그림 5. Schema에 의한 데이터베이스 테이블 생성

## 5. 결 론

현재의 XML 문서 편집 시스템은 XML 관련 응용프로그램에 전문지식이 없는 일반 사용자가 사용하기에 그 기능이 어렵고 복잡하며, XML 또는 XSLT에 대한 단편적인 편집기능 만을 제공할 뿐 이 기종 시스템과 상호 운용이 가능한 통합 개발 환경을 제공하지 않고 있다. 이는 전자문서의 일반화, 대중화에 장해요소로 작용하고 있다.

이에 본 논문에서는 XML 기반의 전자문서 제작과정을 단순화하여 누구나 쉽게 사용할 수 있는 시스템을 개발하고자 하였으며, 이 기종 시스템 즉 데이터베이스와의 상호 운용을 가능하게 하여 통합 개발환경을 제공함으로써 전자문서의 활용가치를 높이고 XML의 일반화, 대중화를 위해 노력하였다.

본 시스템을 그룹웨어의 전자결재 시스템에 도입할 경우 문서정보와 문서내용이 분리되는 특징을 갖는 HTML 기반 전자문서 사용으로 인해 발생되는 "불필요한 이중 데이터 입력"의 문제점을 해결할 수 있을 뿐 아니라 전자문서의 제작과 동시에 데이터베이스에 그 정보가 반영됨으로써 관리적인 측면에서도 그 효율성은 매우 크다 할 수 있다. 이를 통해 사용의 편의성뿐 아니라 전자결재 시스템 구축 속도를 향상시켜 시간과 비용 절감의 효과를 기대할 수 있다.

### 참 고 문 헌

[1] 한성국, 정영식, 주수종, 이현창, "XML 워크샵", 정익사, 2004
[2] XML·SGML 모임, 원두영·이차미 옮김, "표준 XML : 데이터 전달과 저장을 위한 차세대 인터넷 언어", 대청, 2001
[3] Peter G. Aritken, "XML- the Microsoft Way", Addison-Wesley, 2002
[4] Frank Boumphrey 외 11인, 류광 역, "Professional XML Applications", 정보문화사, 1999
[5] Robert Eckstein, "XML Porket Reference", O'REILLY, 1999
[6] Michael Kay, "XSLT Programmer's Reference", Wrox, 2000
[7] Richard Anderson, "Professional XML", Wrox, 2000
[8] www.w3.org/TR/REC-xml/, W3C, 2004
[9] www.w3.org/TR/xslt, W3C, 1999