

자바 기반의 XML 스키마 편집기의 설계 및 구현

지석진, 박정환, 임두욱, 장우영, 신동규, 신동일
세종대학교 컴퓨터공학과
e-mail:jeesj@gce.sejong.ac.kr

A Design and Implementation of XML Schema Editor Based on JAVA

Seok-Jin Jee, Jung-Hwan Park, Doo-Wook, Im Woo-Young Jang,
DongKyo Shin, Dongil Shin
Department of Computer Engineering, Sejong University

요약

컴퓨터의 사용이 보편화되고, 정보화 사회로 발전해감에 따라 이 기간의 문서 정보 교환 및 인터넷과 네트워크 인프라의 향상으로 인한 문서 처리량은 계속 증대되고 있다. 또한 문서의 표현 방법 및 문서량의 증대로 인해 사용자의 요구는 다양해져가고 있다. 기존의 웹(Web)에서 사용되고 있는 HTML(Hyper-Text Markup Language)은 고정된 태그를 사용해서, 사용자가 직접 논리적인 구조를 갖는 문서를 작성할 수 가 없었다. 이에 W3C에서는 XML을 발표하여, 사용자가 논리 구조를 가진 문서를 작성할 수 있도록 하였다. 이러한 XML 문서의 논리적인 구조를 표현하려면, XML 문서 안에 어떤 것을 담을 수 있는지에 대한 규칙들의 집합인 스키마(Schema)가 필요하다. 현재 여러 종류의 스키마가 사용되고 있으며, 이를 이용한 다양한 문서들이 사용되고 있다. 본 논문에서는 현재 가장 많이 사용되고 있는 스키마인 DTD를 보다 쉽게 작성할 수 있는 자바 기반의 스키마 편집기를 설계 및 구현하였다.

1. 서론

네트워크 인프라의 발달 및 인터넷의 보급으로 인하여 문서의 처리량을 기하급수적으로 증가하고 있다. 또한 다양한 운영체제가 생겨남에 따라, 이 각종 시스템간의 정보교환이 문제점으로 대두되고 있으며, 이를 충족시키기 위해서는 특정 구조에 국한되지 않으며 다양한 방식으로 문서를 표현 할 수 있는 방법을 필요로 한다.

기존의 HTML(Hyper-Text Markup Language)은 태그(Tag)를 사용하는 고정된 표현 방법으로 문서를 표현하였다. 이에 W3C에서는 이러한 문제점을 해결하기 위해 문서의 구조를 사용자 직접 작성할 수 있도록 하는 XML(eXtensible Markup Language)이란 언어를 발표하였다 XML의 출현으로 문서 작성시 문서의 논리적인 구조를 표현 할 수 있게 되었으며, 문서를 다양하게 표현 할 수 있게 되었다[1][4].

XML은 자체적인 논리 구조를 가지고 있지만, 그 표현에 있어서는 XSL(extensible StyleSheet Language)이라는 스타일 언어를 차용해서 표현하며 HTML과 같은 링크(Link)를 표현을 위해서는 XML Linking Language)이라는 언어를 사용한다 [1][2][3][6][7].

XML 문서를 작성함에 있어서 그 논리구조를 표현하기 위해서는, XML문서 안에 어떤 것들을 담을 수 있는지에 대한 규칙들의 집합을 필요로 한다[8]. 이런 스키마가 없다면, XML 문서는 의미가 없는 단순 문서에 불과할 뿐이다. 현재 다양한 종류의 스키마가 사용되고 있으며, 스키마의 대표적인 것으로는 DTD(Document Type Definition)가 있다. [7][8].

본 논문에서는 여러 스키마 중에서 가장 많이 사용되고 있는 DTD를 작성할 때, 보다 쉽게 작성할 수 있는 스키마 편집기를 설계 및 구현을 하였다. 2절에

서는 XML 및 스키마와 관련된 연구에 대하여 상세하게 기술하며 3절에서는 스키마 편집기의 설계 및 구현에 기술한다. 마지막으로 4절에서는 향후 연구과제에 대하여 기술하며 본 논문을 마친다.

2. 관련연구

2.1 XML

XML(extensible Markup Language)은 W3C에서 제정한 표준이며 SGML(Standard Generalized Markup Language)의 부분집합(Subset)이라고 할 수 있으며 SGML보다 훨씬 간결화된 문서를 표현하는 표준이다. 즉, XML은 SGML의 방대한 스펙 중 복잡하고 사용하지 않는 것들을 없애고, 고정된 DTD를 사용하는 스타일 중심의 HTML의 단점을 보완하고자 개발되었다.[5][7]

XML은 HTML과 유사한 방법으로 문서를 표현하는데 HTML에서 오직 지정된 태그(Tag)만을 사용할 수 있지만, XML에서는 엘리먼트(Element)을 사용자가 자유롭게 지정할 수 있다. 즉, 문서의 논리적 구조를 표현하기 위해 작성자가 문서에 의미를 부여할 수 있다는 것이다

XML은 그 자체만으로는 HTML과 같이 웹(Web) 상에 표현이 되지 않으며 XML을 지원하는 브라우저로 볼 경우, 그 논리 구조만이 브라우저 상에 나타난다. 이것은 HTML과 달리 스타일 언어를 따로 분리해두었기 때문이다. 따라서 XML을 HTML과 같은 형식으로 표현하기 위해서는 XSL이란 스타일언어가 필요하다[2]. 또한 XML은 HTML에서 제공하는 링크보다 다양한 링크 방법을 제공하기 위해 XLL이란 언어를 차용해서 사용한다. XLL은 기존 HTML에서 제공하는 방식의 링크의 개념을 벗어나 단순링크(Simple-Link), 확장 링크(Extended-Link) 등과 같은 다양한 링크를 제공한다[3].

2.2 DTD(Document Type Definition)

DTD는 XML문서를 표현함에 있어서 XML문서의 마크업(Markup)의 특징을 정의하는데 필요한 것이다. DTD를 사용하지 않는다는 것은 이런 마크업(Markup)에 의미를 부여하지 않는다는 것과 다름없다. 이런 DTD는 문서를 설명하는데 필요한 모든 요소(element), 엔티티(entity), 노테이션(notation), 그리고 기타 선언들로 구성된다[4].

DTD는 그 쓰임새에 따라서 다양하게 나뉘어 진다. XML에서 DTD가 적용되는 방식에 따라서 적격문서(well-form)와 유효한(valid) 문서로 나뉘게 된다. 적격 문서의 경우, XML 문서를 구성함에 있어서 DTD 형식이 없는 문서를 말하며, 유효한 문서의 경우

DTD형식에 맞게 XML을 구성한 문서를 말한다. 또한 DTD는 문서 내부에서 정의되느냐 혹은 외부에서 정의되느냐에 따라서 내부 DTD와 외부 DTD로 나뉘게 된다. 이러한 DTD의 특징은 다음과 같다

- DTD는 DTD를 참조하는 XML문서의 구조와 유효성을 체크하여 서로 다른 독립된 응용프로그램들끼리의 데이터를 공유하기 위한 방법을 제공한다.
- 표준화된 DTD를 이용하여 외부 시스템과의 상호 데이터 교환을 하는데 사용하기도 하고 자체 문서 형식의 데이터에 대한 유효성을 체크하기 위해서 사용한다.
- DTD 외부 파일로 정의되거나 인라인(inline)형식으로 정의되어 XML 문서에서 참조한다.

2.3 스키마

스키마란 문서가 마크업되는 방식을 기술하는 것이며 DTD는 스키마에 포함될 수 있다. 비록 DTD가 스키마의 일종이기는 하지만 엘리먼트의 데이터형 정의 및 문서의 반복횟수를 정의 할 수 가 없다[8]. 현재 이런 단점을 해결하기 위해서 DCD, SOX, XML-DATA 등의 다양한 스키마가 사용되고 있다 [7].

• XML-DATA

DTD라는 관점에서 보면 XML-DATA는 일종의 구문 스키마이고, 데이터 객체라는 관점에서 보면 개념 스키마이다.

• DCD

수정된 RDF의 구문구조를 사용하며 RDF를 만족하는 방식으로 XML-DATA 제언의 부분집합을 포함한다.

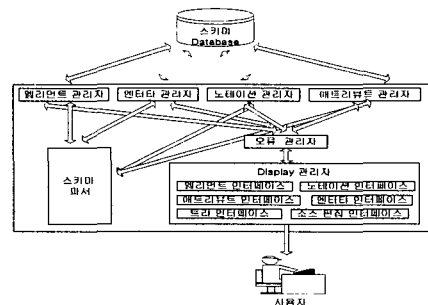
• SOX

객체지향적 XML을 위한 스키마이다.

3. 스키마 저작편집기의 설계 및 구현

3.1 스키마 편집기의 설계

스키마 저작편집기의 구성은 그림 1과 같다.



<그림 1. 스키마 편집기의 구성>

사용자가 입력하는 데이터는 엘리먼트 관리자, 엔터티 관리자, 노테이션 관리자 및 애트리뷰트 관리자를 통해 Display 관리자로 넘겨지며 각각의 인터페이스를 통해 사용자 컴퓨터에 나타나게 된다. 각각의 데이터는 스키마 파서(Parser)를 통해서 처리되며 데이터베이스에 저장된다. 또한 기존에 작성된 스키마 파일을 읽어 올 경우, 파서를 통해서 각각의 관리자로 넘겨지며 이는 해당 인터페이스를 통해서 사용자에게 보여진다.

스키마 편집기의 구성요소로는 엘리먼트관리자, 엔터티 관리자, 노테이션 관리자, 애트리뷰트 관리자, 스키마 파서, 오류 관리자, 스키마 데이터베이스로 구성된다.

● 엘리먼트 관리자

사용자가 입력하는 엘리먼트를 받아서 처리하는 부분으로서 엘리먼트 인터페이스를 통해 사용자에게 보여지며 삽입, 삭제, 수정 후 스키마 데이터베이스에 저장된다.

● 엔터티 관리자

사용자가 입력하는 엔터티를 받아서 처리하는 부분으로서 엔터티 인터페이스를 통해 사용자에게 보여지며 삽입, 삭제, 수정 후에 스키마 데이터베이스에 저장된다.

● 노테이션 관리자

사용자가 입력하는 노테이션을 받아서 처리하는 부분으로서 노테이션 인터페이스를 통해 사용자에게 보여지며 삽입, 삭제, 수정 후 스키마 데이터베이스에 저장된다.

● 애트리뷰트 관리자

사용자가 입력하는 엘리먼트의 애트리뷰트를 받아서 처리하는 부분으로서 애트리뷰트 인터페이스를 통해 사용자에게 보여지며 삽입, 삭제, 수정 후 스키마 데이터베이스에 저장된다.

● 스키마 파서

사용자가 데이터를 입력하거나 혹은 기존의 파일을 불러올 경우 사용되며, 각각의 엘리먼트, 애트리뷰트, 엔터티, 노테이션의 값을 파싱해서 각각의 입력 값을 스키마 데이터베이스에 집어넣는다. 또한 기존의 DTD 파일을 불러들이는 경우, 엘리먼트는 해쉬 테이블을 통해서 트리의 형태로 보여지게 된다.

● 오류 관리자

각종 오류에 대해 처리하는 부분이다.

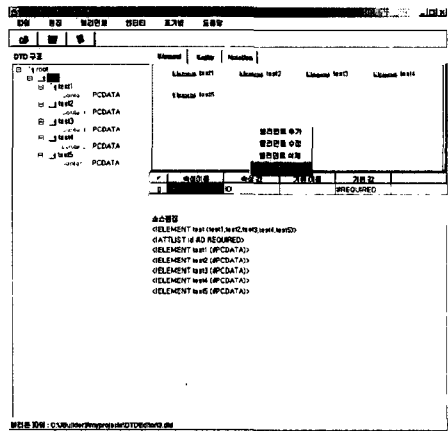
● 스키마 데이터베이스

사용자가 입력한 데이터를 저장하며 저장된 데이터를

갱신하기 위해 삽입, 삭제, 수정할 경우 스키마 데이터베이스를 이용한다.

3.2 스키마 편집기의 구현

본 스키마 편집기의 구현은 Window2000 환경 하에서 Jubiler3.0을 사용하여 개발되었으며 사용자가 쉽게 사용할 수 있도록 WYISWIG 기능 및 드래그 앤 드롭(Drag & Drop)기능이 지원된다. 왼쪽의 패널(Panel)은 사용자가 입력한 엘리먼트를 트리(Tree) 형태로 보여주기 위한 부분이며, 오른쪽 상단의 패널은 엘리먼트, 엔터티, 노테이션을 입력하기 위한 부분이다. 오른쪽 하단의 패널은 사용자가 입력하는 내용을 보여주며, 오른쪽 중간부분의 패널은 애트리뷰트를 설정한다. 스키마 편집기의 전체적인 인터페이스는 그림 2과 같다.

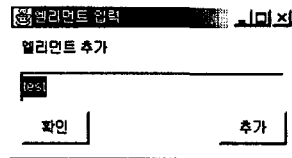


<그림 2. 스키마 편집기의 전체적인 인터페이스>

스키마 편집기 세부적인 인터페이스는 엘리먼트 입력 인터페이스, 노테이션 입력 인터페이스, 엔터티 입력 인터페이스 및 노테이션 입력 값 인터페이스 등이 있으며 세부적인 인터페이스들은 그림 3과 같다.

● 엘리먼트 입력 인터페이스

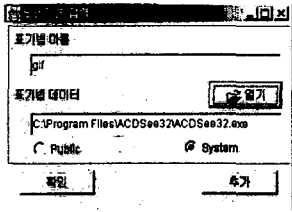
엘리먼트를 입력하기 위한 부분으로 사용자가 입력하는 엘리먼트의 이름을 입력받아 처리하는 부분이며 엘리먼트 인터페이스는 그림 3과 같다.



<그림 3. 엘리먼트 입력 인터페이스>

● 노테이션 입력 인터페이스

노테이션을 입력하기 위한 부분이며 사용자가 입력한 노테이션의 이름 및 노테이션 데이터를 설정 할 수 있으며 그림 4과 같다.

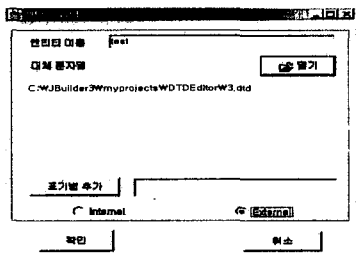


<그림 4.

노테이션 입력 인터페이스>

● 엔터티 입력 인터페이스

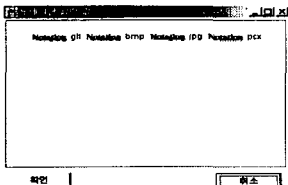
엔터티를 입력받기 위한 부분이며 사용자가 입력한 엔터티의 이름 및 노테이션 입력 인터페이스에서 생성된 데이터들은 표기법 추가를 통해 설정할 수 있으며 그림 5과 같다.



<그림 5. 엔터티 입력 인터페이스>

● 노테이션 입력 값 인터페이스

엔터티 입력 인터페이스에서 표기법 추가 버튼을 누르게 되면 나타나는 인터페이스로써, 입력된 노테이션 값들은 노테이션 입력 값 인터페이스에 나타나며, 엔터티 입력 인터페이스에서 이를 선택을 해서 사용할 수 있다. 노테이션 입력 값 인터페이스는 그림 6과 같다.



<그림 6. 노테이션 입력 값 인터페이스>

4. 결론 및 향후과제

컴퓨터를 이용한 문서의 처리 및 이 기종 시스템간의 정보 교환은 그 중요성이 계속 증가하고 있으며 인터넷이 보편화되고 정보화 사회가 되어감에 따라 문서 정보를 손실 없이 효율적으로 상호 교환 및 검색할 수 있는 문서의 처리 및 문서 구조에 대한 중요성이 대두되고 있다.

이에 따라 본 논문에서는 문서의 논리구조를 표현할 수 있는 집합을 제공해주는 스키마 중 DTD에 대한 초점을 두고 스키마 편집기를 설계 및 구현하였다.

본 스키마 편집기는 시각적인 면과, 기능적인 면에서 기존 상용 제품들과 견주어 부족함이 없으며, 스키마를 보다 쉽게 작성할 수 있도록 대화 방식으로 구성되어 있다. 또한 드래그 앤 드롭(Drag & Drop)을 지원하고 있으며 오류에 대한 정보를 사용자에게 보여주고 있다.

향후에는 스키마를 보다 효율적이며 세부적으로 작성할 수 있는 SOX, DCD, XML-DATA와 같은 스키마를 이용한 스키마 편집기를 연구할 것이며, 스키마 편집기를 포함한 시스템으로의 확장을 위한 연구를 계속할 것이다.

7. 참고 문헌

[1] W3C, "Extensible Markup Language", <http://www.w3.org/XML/>
 [2] W3C, "Extensible Stylesheet Language", <http://www.w3c.org/Style/XSL>
 [3] W3C, "XML Linking Language" <http://www.w3c.org/XML/Linking>
 [4] W3C, "Hyper-Text Markup Language" <http://www.w3c.org/MarkUp>
 [5] 정회경 외 2인 공저, "SGML 가이드". 사이버 출판사, 1997
 [6] 정회경, "차세대 웹 문서 표준 XML", 한국 정보처리 학회지, 제6권 제3호, pp25-35, 1999
 [7] Frank Boumphrey, Olivia Direnzo, Jon Duckett, etc "XML Application", Wrox Press Ltd, 1998
 [8] W3C, "XML Schema" <http://www.w3.org/XML/Schema>