XML DTD 기반의 구문지향 문서 작성기

김영철† · 김성근†† · 최종명†††

요 약

XML은 문서의 정적인 요소나 확장성을 해결할 수 있는 차세대 웹 문서 표준 언어이다. 그러나 XML 구조의 복잡성과 문법의 제약 때문에 일반 사용자는 잘 설계된(Well-formed) 문서나 유효한(valid) 문서를 만들기가 어렵다는 문제를 가지고 있다. 본 논문에서는 XML 구문지향 문서 작성기를 설계하고 구현한다. XML 구문지향 문서 작성기는 편집기에 제시되는 구문에 따라 쉽게 문서를 작성할 수 있으며, 작성된 문서는 모두 유효하다(valid)는 특징을 가지고 있다. 본 편집기는 XML 초보자에게 도움을 줄 수 있으며, XML 문서를 작성하는데 높은 생산성을 줄 것이다.

키워드 : 키워드 : XML, 구문지향문서 작성기, 구문트리

Syntax-Directed Document Editor based XML DTD

Young-Chul Kim† · Sung-Keun Kim†† · Jong-Myung Choi†††

ABSTRACT

XML is being accepted as a standard for the next generation web documents, as it enables to extend the document structures. However, general users have difficulties in writing valid and well-formed XML documents, since the documents should satisfy the grammatical constraints of XML. In this paper, we present a syntax-directed XML document editor which will ease users in writing valid XML documents. The editor will help users, and increase productivity in writing XML documents.

Keywords : XML, Syntax-Directed Document Editor, Syntax Tree

1. 서 론

HTML은 배우기 쉽고 작성이 용이하여 많이 사용되었다. 그러나 웹 문서가 보다 거대해지고 복잡해짐에 따라 그 구조가 고정되어 있고 데이터 검사기능이 없는 HTML에 대하여 한계를 느끼게 되었다. 이에 웹을 통한 정보 작성자들은 대규모 상업용 웹에서 사용할 수 있는 새로운 문서처리 언어가 필요로 하게 되었고 World Wide Web Consortium (W3C)에서는 HTML보다 향상되고 문서구조의 확장성과 검사기능을 가진 새로운 XML(eXtensible Markup Language)를 개발하였다. XML은 W3C에서 SGML(Standard Generalized Markup Language)를 WWW 환경에 적합하도록 만든 언어이다[1,2]. XML은 기존의 SGML의 기능 중 확장성과 구조성 그리고 문서의 검증기능 등 다양한 장점을 그대로 유지하면서 WWW 환경에서 요구되는 URL 링크의 삽입과 StyleSheet등의 기능을 추가하여 기존의

† 준회원: (주) 뉴스텍시스템즈 (교신저자)
†† 준회원: 가톨릭상지대학 컴퓨터정보계열 조교수
††† 준회원 : 목포대학교 정보공학부 컴퓨터공학전공 교수
논문접수: 2004년 3월 25일, 심사완료: 2004년 5월 19일

HTML의 비확장성과 문서의 비검증성의 단점을 해결하였다. 또한 확장성, 이식성 등의 특징 때문에 문서의 표현, SOAP, XML Query 등의 프로그래밍 언어로 활용되고 있으며, MathML, VML, CML[3] 등의 특수한 목적으로 활용되고 있다[4]. XML은 문서의 확장성을 제공하기 위해 문서의 구조를 표시하는 DTD(Document Type Definition)와 문서를 제공하며 하나의 문서로 여러 문서형태를 제공하기 위해 Style Sheet을 제공하며 다양한 URL링크들을 제공한다. 그러나 문서의 구조를 나타내는 DTD나 StyleSheet 등은 초보자들이 사용하기에는 어려움이 많다[3]. 따라서 DTD에 익숙하지 않은 문서 작성자가 DTD에 맞게 XML 문서를 작성하는 것은 결코 쉬운 일은 아니다. 이처럼 DTD에 익숙하지 않은 문서 작성자에게 빠르고 정확한 문서 편집 환경을 제공하는 XML 구문 지향 문서 작성기가 필요하다. XML은 HTML이나 SGML과 그 특성이 상이함으로 기존의 파서나 브라우저를 활용할 수 없다. 그러므로 XML 문서를 적극적으로 활용하기 위해서 문서의 유효성을 검사하는 파서, 문서를 시각화해주는 브라우저, 문서의 생산성을 향상시키는 문서 작성기가 절실히 요구되고 있다. 따라서 XML 어플리케이션을 위한 구문 기반 문서 문서 작성기의 설계 및 구현은 단계별 XML 어플리케이션 개발 방법 제시로 문서 생산성에 크게 기여할 것으로 기대된다.

본 논문은 다음과 같은 형태로 구성되어 있다. 제 2장에서는 XML 편집 시스템에 대한 관련 연구를 제시하였으며, 제 3장에서는 XML 편집 시스템의 설계에 대해서 기술하였다. 또한 제 4장과 5장에서는 각각 구현과 결론 및 향후 연구에 대해서 기술하였다.

2. 관련 연구

XML을 위한 편집 시스템은 현재 여러 가지 언어로 제작되고 있는데, DTD를 가지고 유효성 검사(validating)를 하는 문서 작성기와 DTD없이 잘 설계되었는지(well-formedness)만을 검사하는 문서 작성기로 나뉘어 진다. 또한 현재의 문서 작성기들은 텍스트 편집 환경보다는 트리와 같은 그래픽 사용자 인터페이스를 사용해서 보다 직관적으로 편집이 가능하다. 그러나 이들은 XML 문서처리를 완벽하게 처리하지 못하거나, XML의 변형된 형태의 문서를 처리하거나, SGML에 관련된 소프트웨어를 수정한 형태를 가짐으로서 순수한 XML 문서를 처리하는데 적합하지 못하며, XML 어플리케이션 사용자를 위한 소프트웨어로 부족한 점이 많다. 현재 특정 XML 어플리케이션을 위한 문서 작성기는 찾아보기 힘든 실정이다.

Techno2000에서 개발한 CLIP! XML Editor[5]는 트리 기반, 텍스트 기반의 편집을 지원하며 새문서 작성 마법사를 통해서 처음부터 마지막 단계까지 단계별로 가이드 해주는 마법사를 지원한다. 또한 DTD 트리 보기 기능, 에러 출력 및 수정 기능을 제공하며 잘 설계된(well-formed) XML 문서에서 DTD를 추출, 생성해 준다. 생성된 DTD는 비슷한 구조를 갖는 문서의 저작에 이용할 수 있어 다수의 문서를 쉽게 작성할 수 있다. 기본적인 텍스트 검색과 엘리먼트 검색 외에도 컨텐트에 기반한 엘리먼트 검색과 컴포넌트 검색 등 확장 검색이 가능해 원하는 정보를 쉽게 찾을 수 있다. 오류 수정 기능으로는 문서의 유효성 검사 시 옵션에 의해 첫 번째 오류 감지에서 검사를 멈출 수도 있고 문서전체의 오류를 출력하게 할 수도 있다. 출력된 오류 메시지를 선택하면 오류가 발생한 곳으로 바로 이동할 수 있어 쉽게 오류를 수정할 수 있다. 마지막으로 다른 XML 문서들의 엘리먼트를 불러와 재사용할 수 있어 다량의 XML 문서 저작 시 문서 작성 시간을 최소화할 수 있다.

Vervet Logic에서 개발한 XML Pro v2.01[6]은 순수 자바 애플리케이션으로 쉽고 직관적인 그래픽 사용자 인터페이스, 문서 구조를 유지하는 문서 트리 편집 뷰, DTD를 통한 XML 유효화 검증을 지원한다. 또한 편리한 엘리먼트 생성 및 관리를 위한 엘리먼트 마법사(element wizard), 속성 생성 및 관리를 위한 속성 마법사(attribute wizard)도 제공된다. 기본적으로 엔티티(entity), CDATA, 주석을 포함한 W3C의 XML 1.0 스펙을 지원한다.

Cuesoft의 EXml[7]은 well-formed XML 문서 문서 작성기로서 트리나 소스 뷰를 지원한다. 트

리 뷰를 통해서 엘리먼트 편집이 가능하다. 다른 문서 작성기와 마찬가지로 well-formedness를 체크하는 것 이외에 특이한 사항은 XSL 이름공간(namespace)지원과 PCDATA, 주석, 속성값을 엘리먼트에 매핑한 테이블 출력, 소스뷰 상에서의 직접접적인 텍스트 편집이 가능하다.

Synthesizer Generator[8]는 Gramma Tech에서 개발한 구문 지향 문서 작성기 자동 생성도구이다. 이 도구는 특정한 응용 프로그램에 대한 구문 지향 편집 환경을 개발하는데 많은 도움을 주고있다. 특히 편집되는 각 객체는 연관성있는 속성 트리 형태를 구축하여 표현된다. 이 도구에서 표현된 언어는 SSL(Synthesizer Generator Language)로 작성된다.

SVG(Scalable Vector Graphics)[9]는 세가지 그래픽 객체 즉, Shapes, images, text로 구성되었으며, 객체들은 그룹으로 묶을 수 있고 스타일을 지정할 수 있다는 특징을 가지고 있다. 또한 간단한 움직임을 표현하는 것뿐만 아니라 3차원 모핑(Morphing) 효과까지 그래픽 표현을 위한 많은 기능들을 정의함으로써 웹 개발자에게 동적인 웹 페이지를 개발할 수 있게 도와준다.

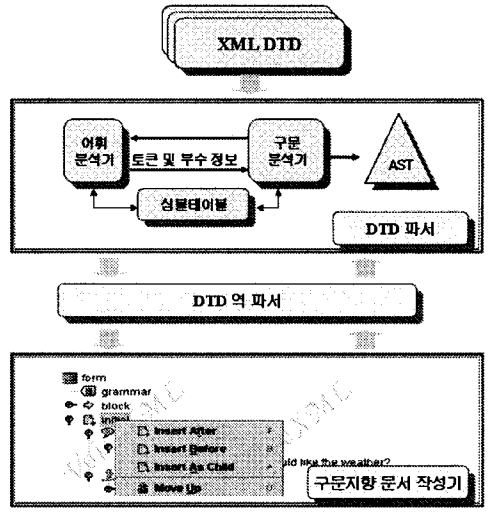
GML에 기반한 XML 응용프로그램인 XGMM (eXtensible Graph Markup and Modeling Language)[10]는 원소가 그래프의 노드(node)와 에지(edge)를 나타낸다. XGMML은 서로 다른 그래프 저작도구나 탐색 도구들 간의 그래프 교환을 목적으로 WWWPAL 시스템에서 웹 사이트를 표현하기 위해 개발되었다.

DiaGen(Daigram Editor Generator)[11]은 다이어그램 컴포넌트를 정의하는 방법, 리듀싱 규칙, 문법 및 의미를 기술하여 자동으로 다이어그램 편집기를 생성한다. DiaGen은 하나의 다이어그램 정의를 위해 여러 개의 복잡한 hypergraph 문법을 사용함으로써 구현이 어렵고, 확장성도 결여되어 있다는 단점을 가지고 있다.

Altova의 xmlspy® 2004[12]는 유효한 XML 문서를 쉽게 만들 수 있도록 템플릿과 상황에 대응하는 (context-sensitive) 편집, 필요한 엘리먼트와 속성을 자동으로 채우는 기능(auto-completion)을 지원한다. 또한 현재 위치에서 어떤 엘리먼트를 사용할 수 있는지 알려주는 엔트리 헬퍼(Entry Helper)를 지원을 통한 직관적인 사용자 인터페이스를 제공하고 DTD 편집기, XSL 편집 및 디버깅 기능 등을 지원한다.

3. 구문지향 XML 문서 작성기 모델

본 논문에서 개발한 XML 문서 작성기는 다음 그림 1과 같다. 그림에서 보는 바와같이 문서 작성기 모델은 XML DTD를 입력받아 내부적으로 DTD 파서에 의해 파싱되어 문서 작성기에 나타난다.



<그림 1> XML 구문 지향 문서 작성기 모델

AST는 DTD를 내부 문서 작성기나 역 파서에서 사용할 수 있도록 만들어놓은 문법 트리이다. 문서 작성기에서 만들어지는 모든 문서는 AST에 의해서 문법이 검증되어 올바른 문서만 만들어진다. 또한 DTD 파서(Parser)는 문서 유효성 검증을 하며, 문서의 토큰을 인식해 내는 어휘 분석기와 일련의 토큰의 조합이 적합한 구문 구조를 갖는가를 검사하는 구문 분석기로 구성된다. 역 파서(Unparser)는 AST의 각 노드를 그래픽 트리 구조로 보여주는 역할을 한다. 이것은 일반 사용자들이 XML을 직접 편집할 수가 없고, 대신에 그래픽 사용자 인터페이스 편집 환경을 통해서 편집해 나가기 때문에 필요하다. 역파서는 AST를 그래픽 트리 구조로 역 파싱하는 기

능 외에 일반 텍스트 문서로 저장도 한다.

3.1. DTD 파서

3.1.1. 어휘 분석과 구문 분석

어휘 분석기 (Lexical Analyzer)는 DI에서 일련의 문자를 받아서 토큰으로 인식하여 해당 토큰을 구문 분석기에 전달한다. 토큰은 각 엘리먼트의 태그와 속성 및 주석으로 구성되어 있다. 구문 분석기 (Syntax Analyzer)는 어휘 분석기로부터 토큰을 받아서 주어진 토큰의 조합이 문법에 올바른 것인지를 검증한다. 이 때 하향식 파싱(top-down parsing)의 대표적인 방식이며 다중 엔트리(multiple entry)를 지원하는 recursive descent parsing(RDP) 기법을 사용한다. RDP는 결정적 파서(deterministic parser)로서 비결정적 파서(non-deterministic parser) 보다 빠르며, 다중 엔트리를 통해서 문서 일부에 대한 유효성 검증이 가능하다.

3.1.2. 추상구문트리(AST)

AST는 XML 문서를 파싱하여 문서 작성기에서 편집 가능하도록 설계한 자료구조이다[7]. 사용자가 문서 작성기에서 행하는 편집 연산에 대한 처리는 모두 AST 인스턴스에 반영된다. AST의 한 노드는 문서 내에서 사용되는 태그 이름과 연관된 속성정보와 AST에서 다른 노드들과의 연결 정보들로 구성된다. 문자열 데이터는 작성자가 직접 작성한 문서 내용을 갖고 있는 부분으로, 이것은 PCDATA, CDATA나 주석과 같은 서브 엘리먼트로 이루어진 엘리먼트인 경우에만 유효한 정보이다. 실제 이 부분은 문자열만을 저장한다.

문서 작성기의 핵심은 DTD를 문법 규칙의 집합인 추상 문법(abstract syntax)으로 정의하는 것이다. 편집하는 개체(object)는 문법에 따라서 유도 트리(deprivation tree), 즉 AST로 표현된다. 문서 작성기 사용자 인터페이스의 조작에 따른 텍스트나 문서 구조의 변화는 주어진 구문 트리가 변화함을 의미한다. 본 논문에서는 XML의 DTD를 AST로 변환하였다. 추상 문법은 다음과 같은 형식의 프로덕션(production)의 집합으로 이루어진다.

$$x_0 : op\ (\ x_1, x_2 \cdots x_k\ );$$

여기서 □는 연산자명(operator name)이고 각 □는 문법의 비단말명(nonterminal name)이다. 프로덕션 인스턴스를 구별하기 위한 목적인 연산자를 제외하고 문법 규칙은 다음과 같은 문맥 자유 프로덕션(context-free production)과 같다.

$$x_0 \rightarrow x_1 x_2 \cdots x_k$$

비단말의 집합으로 이루어진 유도 트리를 term이라고 부른다. 다음은 "<xml><form/></xml>"에 대한 □항 연산을 표현한 term의 예이다. 이때 Nil연산의 term에서는 괄호를 생략했다.

MainXml(XmlList(XmlElementForm (FormNil), XmlNil))

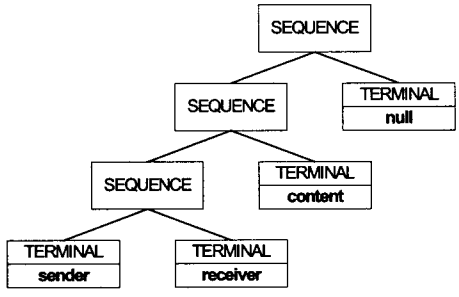
추상 구문트리의 예를 들기 위해 다음 표 1과 같은 memo 문서 구조를 나타내는 XML DTD이다.

<표 1> memo에 관한 DTD

```
<!DOCTYPE memo [
 <!ELEMENT memo (sender, receiver, content)>
 <!ELEMENT receiver (person)+>
 <!ELEMENT sender (#PCDATA)>
 <!ELEMENT person (#PCDATA)>
 <!ELEMENT content (#PCDATA)>
 <!ATTLIST sender sign CDATA #REQUIRED>
]>
```

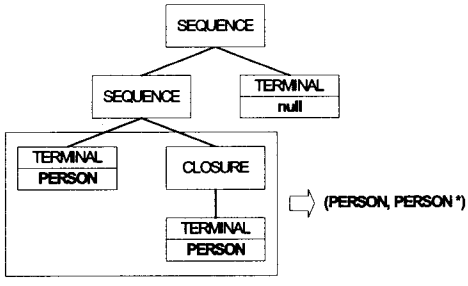
memo에 관한 AST는 다음 그림 2와 같다.

memo 엘리먼트는 속성 정의 리스트가 존재하지 않는다. 모델 상에서는 SENDER, RECEIVER, CONTENT 엘리먼트가 SEQ 연결자로 연결되어 있으므로 SEQUENCE 클래스를 사용하여 SENDER, RECEIVER, CONTENT를 TERMINAL 클래스의 이름으로 하는 트리를 구성한다. momo 엘리먼트의 구성 모델 구조는 momo 엘리먼트의 선언과 동시에 트리의 구성이 시작되어 선언 끝의 '>' 문자를 만날 때까지 트리 구성이 완료된다.



<그림 2> MEMO 엘리먼트 AST

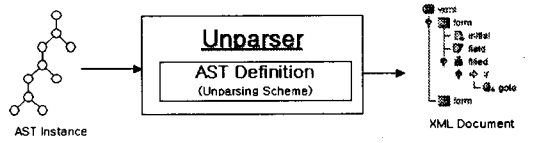
그림 2에서는 receiver 엘리먼트는 나타나져 있지 않다. 이는 receiver가 또 다른 엘리먼트를 형성하고 있기 때문이다. 따라서 receiver 엘리먼트는 다음 그림 3과 같다.



<그림 3> RECEIVER 엘리먼트의 AST

3.2. 역 파서(Unparser)

역 파서는 그림 4와 같이 AST 인스턴스를 입력으로 받아들이고 원래 문서 형태를 ASTD의 역 파싱 규칙(unparsing rule)을 참조하여 출력장치(화면 혹은 디스크)에 적당한 형태를 갖추어 보기 좋게 출력한다.



<그림 4> 역 파서

역 파싱 규칙은 다음과 같다.

Phylum: operator [selection symbols]

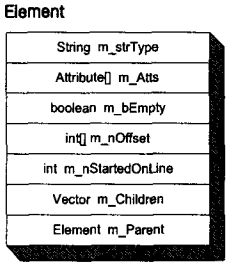
여기서 꺾쇠괄호(square bracket)는 해당 프로덕션의 이미지이다. 선택 심볼(selection symbol)은 @과 이름으로 나누어지는데 @은 자식 프로덕션을 역 파싱 함을 의미하고, 이름은 트리 상에서 출력할 아이콘이나 텍스트를 결정한다. 실제로 ASTD와 AST 인스턴스를 이용하여 역파싱을 하는 역파싱 알고리즘은 다음과 같다.

```
Algorithm UNPARSING( P, N)
Input : Node
Output : 역파싱된 출력물
Definition:
    P = 부모 노드
    N = 현재 노드
Method:
 if( N이 리스트 노드이면) {
    do {  AST정의에서 N의 프로덕션에 대한 역파싱
규칙을 읽는다.
      if( 읽은 역파싱 규칙이 첫 번째 @이면)
        UNPARSING( N, N의 왼쪽 노드 링크)
      else if( 읽은 역파싱 규칙이 두 번째 @이면)
        UNPARSING( N, N의 왼쪽 노드 링크)
      } while( 역파싱 규칙이 남아 있을 경우)
  }
else {
    do {  AST정의에서 N의 프로덕션에 대한 역파싱
규칙을 읽는다.
       if( AST정의에서 N의 프로덕션 메뉴값이 STRIN
G이면) P의 자식으로 STRING을 출력한다.
       else if( 읽은 역파싱 규칙이 @이면) {
              if( N의 Prod에 대한 메뉴값이 STRING이
면) P의 자식으로 N의 왼쪽 노드 링크를 출력한다.
              else
              UNPARSING( N, N의 왼쪽 링크 노드)
            }
       } while( 언파싱 규칙이 남아 있을 경우)
}
```

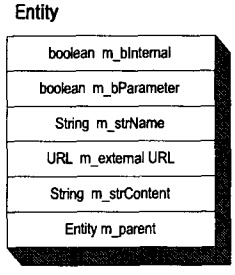
역파싱 알고리즘은 처음에 노드가 리스트 노드인가 아닌가를 검사한다. 노드가 리스트인 경우에는 화면에 출력이 없고 자식 노드에 대한 역파싱 루틴 호출만 있다. 역파싱 규칙이 첫 번째 @인 경우에는 왼쪽 노드 링크를, 두 번째 @인 경우에는 오른쪽 노드 링크를 현재 노드와 함께 UNPARSING인자로 전달한다. 이 과정을 역파싱 규칙이 없을 때까지 진행한다. 노드가 리스트가 아닌 경우에는 @에 대한 처리 외에 화면 출력을 위한 실제적인 부분이 있다. 해당 노드의 프로덕션에 해당하는 역파싱 규칙을 읽은 후에 AST 정의의 메뉴값이 STRING인 경우는 주석이나 PCDATA인 경우로서 화면에 해당 노드를 만든 후에 실제 주석값이나 PCDATA값을 출력한다. STRING이 아닌 경우는 @으로서 이때는 오직 왼쪽 노드 링크만이 존재하므로 현재 노드와 함께 UNPARSING인자로 전달한다.

3.3. XML DTD 내부 자료 구조

XML 문서 처리 부분에서는 XML 문서의 Element, Entity에 대한 내부 자료구조는 그림 5, 6과 같다.



<그림 5> Element의 자료구조



<그림 6> Entity 자료구조

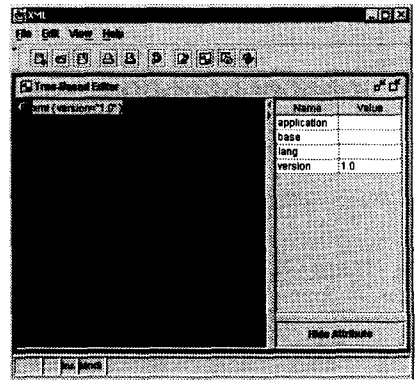
그림에서처럼 Element에 대한 자료구조는 트리구조로 링크되도록 하기 위해서 부모와 자식노드에 대한 링크가 있고, Element에 해당하는 Attribute의 자료구조가 배열로 정의 되어있다. 따라서 하나의 Element에 Attribute가 List형태로 존재하게 된다. Entity는 XML 문서 내에서 매크로(Macro)처럼 사용되기 때문에 XML 문서 내에서 Entity의 참조에 대한 확장이 이루어져야 한다. Entity의 자료구조에서는 Entity에 대한 정보를 리스트 형태로 표현하기 위해서 링크가 있고, Entity의 종류(즉, General Entity인지, Parameter 인지)에 대한 정보가 있다. 또한, Entity의 이름과 내용을 위한 문자열 형태의 정보가 있다.

4. 구현

본 시스템에서는 XML 문서의 구조적 시각화를 위해서 자바 스윙(JDK 1.2.2에서 구현)에서 제공하는 트리와 테이블 인터페이스를 제공한다.

4.1. 초기 화면

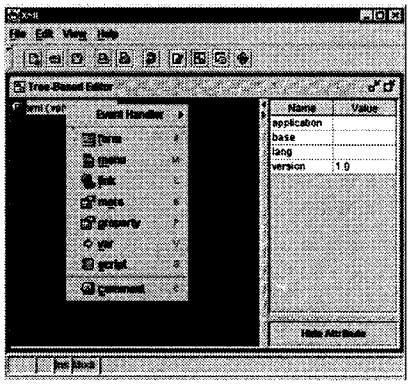
그림 7은 트리 기반 XML 어플리케이션 문서 작성기의 초기 새문서로 시작한 그림이다. 윈도우의 좌측은 엘리먼트 트리 윈도우로서(element tree window) XML 문서를 트리 구조로 보여주는 윈도우이다. 문서 작업의 대부분은 엘리먼트 트리 윈도우의 엘리먼트 편집으로 이루어진다. 우측은 속성 윈도우(attribute window)로서 엘리먼트 트리 윈도우에서 엘리먼트를 선택했을 때 해당 엘리먼트의 속성을 편집하는 윈도우이다. 또한 속성 윈도우 밑에는 속성 보이기/숨기기 버튼이 있다. 이 버튼은 좌측의 엘리먼트 트리 윈도우의 각 엘리먼트 노드에 속성 정보를 보이거나 숨기는 기능을 토글 시킨다. 문서의 구조적인 정보는 숨기기 설정하고 세부적인 내용은 보이기로 하는 설정하는 것이 좋다.



<그림 7> XML 구문지향 문서 작성기 초기화면

4.2. 팝업 메뉴

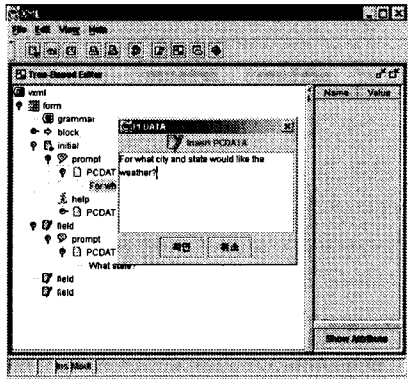
다음 그림 8은 구문 지향 문석 작성기의 팝업 메뉴를 보여준다. 엘리먼트 생성을 위해서는 노드에 마우스 왼쪽 버튼을 누르면 그림 8과 같이 확장 가능한 엘리먼트가 팝업 메뉴로 나온다.



<그림 8> XML 구문지향 문서 작성기의 팝업 메뉴

4.3. PCDATA 입력

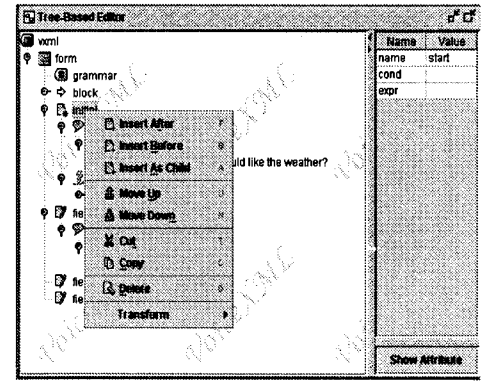
다음 그림 9는 PCDATA 입력 화면이다. 모든 엘리먼트는 문서 작성 윈도우에서 입력 대화상자를 가지고 편집을 한다. 이때 사용자가 입력을 마치고 확인 버튼을 눌렀을 때, 입력한 내용이 유효한지를 검사하기 위해서 내부 다중 엔트리 파서를 사용한다. 입력에 오류가 있을 경우에는 다시 입력을 받게 된다.



<그림 9> XML 구문지향 문서 작성기의 PCDATA 입력 화면

4.4. 엘리먼트 편집을 위한 인터페이스

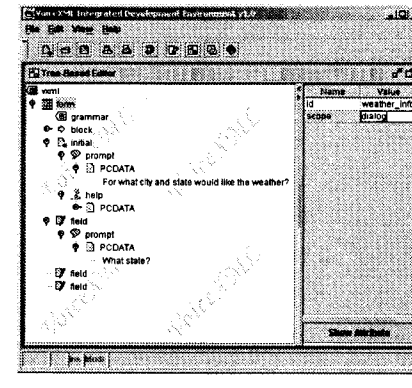
다음 그림 10은 본 시스템의 엘리먼트 편집을 도와주는 팝업 메뉴를 보여준다. 이 메뉴는 문서 작성자가 쉽게 엘리먼트를 수정/삭제/복사/변환 가능하도록 지원한다.



<그림 10> XML 구문지향 문서 작성기의 엘리먼트 편집을 위한 팝업 메뉴

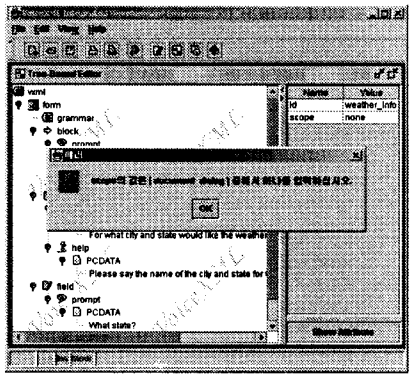
4.5. 속성 편집을 위한 인터페이스

엘리먼트 트리 윈도우의 각 엘리먼트를 클릭하면 이에 오른쪽 속성 윈도우에 해당 엘리먼트에 대한 속성명과 속성 값이 나온다. 이 때 각 값에 대한 편집이 가능하다.



<그림 11> 속성 편집

그림 11과 같이 각각의 속성 값에 대한 입력을 마친 후에는 AST 정의의 속성 정보를 이용하여 속성 값 유효성 검증을 한다. 만일 값에 오류가 발생한 경우에는 [그림 12]와 같이 오류 메시지가 나온다.



<그림 12> 속성 편집 오류

4.6. 평가

본 논문에서는 잘 구성되고, 유효한 xml 문서를 자동으로 만들 수 있는 시스템을 설계하는데 목적이 있다. 따라서 xml DTD를 이용하여 구문 지향 편집기를 구현하였다.

본 논문에서 제시한 XML 구문지향 문서 작성기는 XML DTD에 따라 문서를 작성할 수 있기 때문에 다른 XML 문서 작성기와는 달리 많은 장점을 가지고 있다. 첫째, 본 시스템은 내부 DTD를 파싱하여 AST를 생성한다. 생성된 AST는 내부적으로 구문분석이 끝난 상태이므로 문서가 DTD에 유효하다는 점이다. 둘째, 타 시스템과는 달리 문서 작성기의 지시자에 의해서 문서가 작성되므로 만들어질 문서는 잘 설계된 문서(well-formed) 문서이며, 유효한(valid) 문서임을 알 수 있다. 셋째, 기존의 DTD나 생성된 문서를 트리 형태로 표현해주고, 문서를 작성할 수 있도록 지원한다. 예를 들면, 그림 8에서 처럼 특정한 엘리먼트를 삽입하거나 설명문을 추가하는데 쉽게 문서 작성자에게 제시된다. 또한 속성 편집을 쉽게 할 수 있는 속성 창을 지원한다. 넷째 타 시스템과는 달리 트리 조작을 쉽게 할 수 있도록 팝업 메뉴를 지원한다. 예를 들면, 그림 10과 같이 특정한 엘리먼트를 다른 곳으로 쉽게 옮길 수 있도록 지원한다.

이 밖에도 AST를 이용하면, 타 시스템과는 달리 다음과 같은 많은 장점을 가지고 있다. 첫째, AST를 생성하기 위해서는 DTD 파서가 수행된다. 따라서 잘못된 DTD는 구문 분석 과정에서 오류를 자동 검사할 수 있는 특징을 가지고 있다. 둘째, AST를 이용하면, 구조가 똑같은 DTD에 대해서도 XML 문서의 유효성을 검사할 수 있다. 즉, DTD의 구조는 같지만, 이름 및 스타일이 달라도 XML 문서의 유효성을 검사할 수 있다는 특징을 가지고 있다. 왜냐하면, AST를 생성하는 과정에서 DTD의 단말노드들은 문서의 유효성에 영향을 미치지 않기 때문이다. 셋째, 구성된 AST는 SVG[9], XGMML[10], DiaGen[11]과 같은 그래픽 툴에 적합한 구조에서 쉽게 활용할 수 있다. 즉, AST는 DAG(Direction Acyclic Graph) 형태로 이용되는 응용프로그램들에서 쉽게 활용할 수 있다는 특징을 가지고 있다.

5. 결론

XML 문서를 작성하는 초보자는 XML 구문에 맞는 유효한 문서를 작성하기 어렵다. 따라서 XML DTD 구문에 맞도록 XML 문서를 쉽게 작성할 수 있는 문서 작성기가 필요하다. 본 논문에서는 XML 문서의 유효성(Validation)과 잘 설계된(Well-formed) 문서를 자동적으로 설계하기 위한 구문 지향 문서 작성기를 설계하고 구현하였다. 따라서 본 시스템의 특징은 XML 어플리케이션에 대한 편집 도구로서 기존의 수동으로 유효성 검사를 해야 하는 방식의 불편함을 해소하고 일반 사용자가 쉽게 XML 문서를 편집 할 수 있도록 구문 지향 방식을 적용한 것이 특징이다. 본 시스템 모델은 AST의 역파싱 규칙을 통해서 AST 인스턴스를 화면이나 파일로 출력을 했으며, 문서 편집의 효율성과 문서 구조의 파악이 용이하게 하기 위하여 내부 문법이 익숙하지 않더라도 쉽게 사용이 가능하도록 문서 편집을 할 때 확장/치환 가능한 엘리먼트를 AST 참조를 통해 제시하는 문서 기반의 비주얼 프로그래밍 환

경을 제공하였다. 또한 사용자들이 문서 편집 시 문법에 맞는 문서를 작성할 수 있도록 함으로써 문서의 오류를 줄였다. 따라서 사용자는 문서 작성 중 항상 유효한 문서를 작성할 수 있다는 장점을 가지고 있다.

향후 연구과제는 DTD의 AST 변환을 자동화하고 이를 이용하여 실시간 유효성 검증이 가능한 범용 XML 편집기의 설계 및 구현이 이루어져야 할 것이다.

참 고 문 헌

[1] Extensible Markup Language(XML) 1.0(1998). Available http://www.w3.org/TR/REC-xml.

[2] VoiceXML Forum(2000). Voice eXtensible Markup Language 1.0. March 7.

[3] Chemical Mrakup Language, Available http://www.xml-cml.org.

[4] Simple Object Access Protocol(SOAP), XML Query, and Mathmaticl Markup Language, Available http://www.w3.org.

[5] CLIP! XML Editor. Available http://xml.t2000.co.kr.

[6] XML Pro v2.0. Available http://www.vervet.com/products.php.

[7] EXml v1.2. Available http://www.cuesoft.com/products/exml.asp.

[8] Thosmas W. Reps(1988). Tim Teitelbaum, The Synthesizer Generator: A System for Constructing Language-Based Editors. Springer-Verlag.

[9] Scalable Vector Graphics, Available http://www.w3.org/Graphics/SVG

[10] Extensible Graph Markup and Modeling Language, Available http://www.cs.rpi.edu/~puninj/XGMML

[11] Mark Minas and Oliver Koth, Generating Diagram Editors with Diagen(1999), in Proc. of the Int'l Workshop with Industrial Relevance, pp. 433-440.

[12] xmlspy® 2004, Available http://www.xmlspy.com/products_ide.html

김영철

1990년 : 한남대학교 전자계산학과 졸업

2003년 : 숭실대학교 컴퓨터학과 공학박사

현재 : (주)뉴스텍 시스템즈 이사, 명지전문대학 겸임교수

※주관심분야: 컴파일러, 망관리, 프로그래밍 언어, XML 어플리케이션

김성근

1993년 : 숭실대학교 전자계산학과 졸업

1995년~현재: 숭실대학교 컴퓨터학과 박사과정

현재 : 가톨릭상지대학 컴퓨터정보계열 조교수

※주관심분야: 프로그래밍 환경, 에이전트 프로그래밍 언어

최종명

1992년 숭실대학교 전자계산학과 학사

2003년 숭실대학교 전자계산학과 공학박사

현재 : 목포대학교 정보공학부 컴퓨터공학전공

※주관심분야 : 시각 프로그래밍, 멀티패러다임 시스템, XML