

다중뷰를 지원하는 구조적 XML 에디터 생성 (A Multiple-View Structured XML Editor Generation)

신 경 희, 유 재 우

숭실대학교 컴퓨터학과

{khshin, cwyoo}@computing.ssu.ac.kr

요 약

XML은 구조를 포함하는 웹 문서 작성언어로서, 올바른 XML 문서를 작성하기 위하여 DTD를 이용한 구문기반의 에디터가 필요하다. 본 연구에서는 DTD에 독립적이고 다양한 편집화면이 지원되는 에디터를 동적으로 생성하는 XML 에디터 생성기를 제시한다. 이 생성기는 구조적 개발환경과 다중 뷰 특징을 갖는 문서작성환경으로, 문서의 구문은 추상문법으로, 편집 문서의 화면 출력 정보는 유형문법으로 각각 정형화 시켜 문서구조와 표현에 있어 유연성과 효율성이 있는 XML 문서작성 환경을 제시한다.

1. 서론

XML(eXtensible Markup Language)은 웹 문서 작성언어로서, XML 문서는 그 문서 자체에 계층적 구조를 갖고 있으며 여러 속성들을 가질 수 있다. 이 구조는 태그에 의해 표현되는데 태그의 이름이나 순서 또는 태그가 갖는 속성 등은 문서에서 지정한 문서형 정의부 (Document Type Definition: DTD)에서 결정된다. 따라서 올바른 XML

문서를 작성하기 위해서는 정의된 DTD를 알고 있어야 하며, 이러한 규칙에 따라 작성자에게 구조나 속성정보를 가르쳐 주면서 문서를 구문 지향적 편집을 도와주기 위한 에디터나, 이를 포함하는 종합적 문서개발환경 도구가 필요하다.

또한 구조적 기반의 에디팅 중심으로, 사용자가 대화식으로 작업하면서 프로그램 혹은 문서를 올바로 작성하기 위해서는 관련 정보를 효율적으로 보여주고 사용자와 인터페이스하는 방법의 중요성이 강조되고 있다.

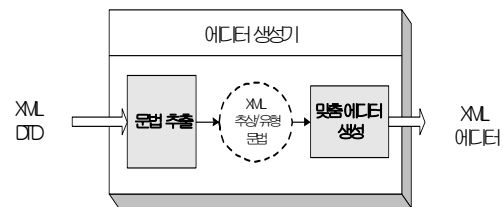
XML 문서를 편집하기 위한 에디터는 DTD를 이용하여 문서의 유효성(validation)을 검사하는 것과 태그가 잘 정의되어 있는(well-formedness)지를 점검하는 것으로 구분한다. XML Pro v2.0[9]은 DTD를 이용하여 문서 편집이 가능한 것으로 문서의 유효성을 검사하는 트리 뷰를 제공하는 구문 에디터이고, Exml[10]은 DTD 없이 태그가 잘 정의되어 있는지를 점검하는 구조 에디터로서, 트리와 텍스트 뷰가 제공된다. 대부분의 XML 에디터들은 계층적인 태그 구조를 중심으로 편집되는 구조적 에디터로서 트리 뷰를 제공한다. 에디터의 제한적인 뷰 제공은 사용자 입장에서 문서의 내용 확인에 불편한 단점들을 갖고 있다.

다중 뷰 시스템[2]은 프로그램이나 문서와 같은 단일 객체에 대한 처리를 위하여 두 개 이상의 뷰를 제공하는 환경이다. 다중 뷰는 여러 데이터를 서로 다른 뷰에 나눠 나타내는 것과 하나의 데이터를 여러 가지 시각적인 표현을 이용하여 서로 다른 뷰에 나타내는 것으로 구분할 수 있다. 따라서 사용자 요구에 따라 원하는 형태의 데이터를 확인할 수 있도록 한다.

소프트웨어 개발을 위한 여러 가지 환경이 존재 하지만, 본 연구에서는 유효성 확인이 필요한 XML 문서 편집을 위한 다중 뷰 지원의 구조기반 개발환경에 대하여 기술한다. XML 에디터는 DTD에 명시된 구조에 의해 문서를 편집한다. 그러나 DTD는 사용자 자신이 직접 작성할 수 있는 것이므로, XML 에디터는 모든 형태의 DTD에 대하여 처리가 가능한 유연성을 갖고 있어야 한다. 또한 특정된 타입의 단일 뷰를 갖는 에디터 보다는 다양한 시각적인 표현으로 문서내용

을 여러 가지 형태로 보여주는 다중 뷰를 지원하므로 사용자의 편집환경을 다양하게 제공한다. 본 연구는 이를 위한 효율적인 XML 에디터를 자동 생성하기 위한 방법과 그 모델을 제안하고자 한다.

본 XML 에디터 생성기는 여러 개 DTD에 적합한 에디터를 생성하는 것으로, 별도의 입력 포맷을 사용하지 않고 단지 W3C 규칙에 의해 기술한 XML DTD 파일만을 입력으로 취한다. 따라서 이 생성기는 DTD 파일의 적합성을 확인하기 위한 처리가 내부적으로 필요하고, 구문에디터를 생성하기 위한 문맥자유문법 형태가 요구된다. 그림 1은 본 연구에서 XML 에디터를 생성하는 전체적인 과정을 보여주고 있다.



(그림 1) XML 에디터 생성기 구조

이 생성기는 문법 추출과 맞춤 에디터 생성 처리에 의해 DTD 독립적인 XML 에디터를 만들어준다. 또한 문서의 에디팅 뷰는 문자, 트리 그리고, 아이콘등과 같이 다양한 형태가 지원되는 다중 뷰 특성을 갖게 하여 XML 문서 편집을 위한 개발환경을 좀 더 향상시킬 수 있도록 설계되었다.

논문은 서론부분에서 XML 문서편집을 위한 개발환경과 에디터 생성기에 대한 설명했고, 다음으로 에디터 생성기를 구성하는 문법 추출프로세서와 맞춤 에디터 생성 프로세서의 기능에 대하여 기술하고, 에디터를

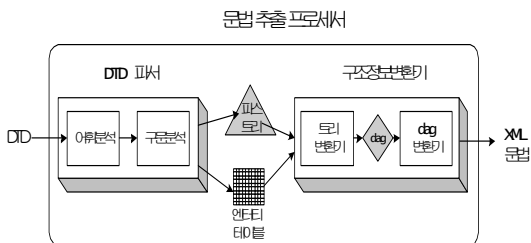
생성하기 위해 필요한 XML 문법에 대한 것으로 에디터 생성과 관련하여 용도에 따른 구분과 그 표기법에 대하여 언급한다. 마지막으로 제시한 에디터 생성기의 구현결과를 설명한다.

2. XML 에디터 생성기 구성

본 연구에서 제시한 에디터 생성기는 각 XML DTD에 적합한 구조 에디터를 동적으로 생성하는 것으로, 이는 DTD에서 에디터를 생성하기 위한 문법을 추출하는 프로세서와 사용자가 원하는 타입의 에디터를 생성하는 프로세서로 이루어진다.

2.1. 문법 추출 프로세서

문법 추출 프로세서는 입력된 DTD가 올바르게 작성된 것인지를 검사한 후 적절한 XML 문법을 생성하는 것이 목적으로, 그림 2와 같이 DTD 파서와 구조정보변환기로 구성된다.



(그림 2) 문법추출 프로세서 구조

DTD 파서는 입력된 DTD를 어휘분석과 구문분석을 한다. 어휘분석에서의 토큰 분류나 구문분석을 위한 문법은 W3C의 XML 표준에서 정의한 규칙을 이용하는데, 본 연구에

서 설계한 DTD 파서는 XML 표준 규칙을 DTD의 메타문법으로서 간주하여 입력된 DTD의 파스트리를 생성한다. 이때 DTD의 엔티티 정보는 별도의 테이블에 저장, 관리한다. 따라서 이 파스트리는 엔티티 선언을 제외한 입력된 모든 DTD의 모든 선언 정보를 포함한다.

구조정보변환기는 내부적으로 트리 변환기와 dag 변환기를 포함한다.

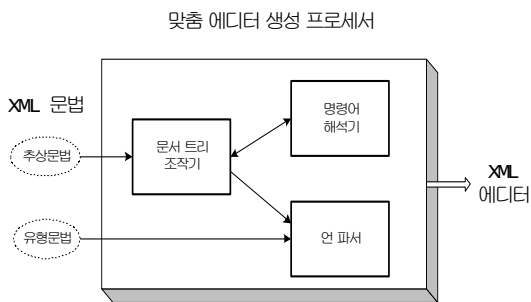
트리 변환기는 DTD 파서의 처리 결과로 생성된 파스트리에서 구문을 나타내는 엘리먼트 노드 정보만을 추출하여 그래프 구조로 재구성한다. 이 그래프는 엘리먼트 노드, 엘리먼트들간의 연관성을 나타내는 지시자 노드와 발생자 정보 노드 그리고 노드들간의 연결을 위한 단 방향 에지로 구성되고 차수는 2로 제한한다.

dag 변환기는 트리 변환기에 만든 그래프 구조를 에디터 구현에 적합한 문맥자유문법(CFG)으로 재 표기하는 것으로, 4단계 처리 과정을 거쳐 정리한다. 먼저, 그래프는 루트노드를 시작점으로 순환하면서 2 레벨 별로 부모노드와 자식노드간의 관계를 CFG의 생성규칙을 만들면서 초기문법을 생성한다. 두 번째 단계는, 생성된 초기문법 중에서 생성규칙의 오른쪽에 한 개의 비단말기호만을 갖는 규칙을 대상으로 생성규칙들을 결합시키므로써 불필요한 생성규칙을 제거한다. 그 다음은 발생자(+ , *)에 따른 생성규칙 처리 단계로 우선, + 발생자 처리한 다음 * 발생자에 따른 생성규칙을 재정리한다. dag변환기에서 만들어지는 문맥자유문법은 XML 에디터 생성에 필요한 것으로, 본 연구에서는 이 CFG를 XML 문법으로 하여 문서구조를 정의한 추상문법과 디스플레이 형태를 정의

한 유형문법으로 구분한다.

2.2. 맞춤 에디터 프로세서

XML 문법에 따라 그리고 사용자가 원하는 인터페이스가 제공되는 맞춤형 에디터를 생성하는 프로세서는 그림 3과 같이 문서트리 조작기, 명령어 해석기, 그리고 언파서 모듈에 의하여 에디터 출력 및 작성 문서 관리를 하게 된다.



(그림 3) 맞춤 에디터 생성 프로세서 구조

맞춤형 에디터 프로세서가 생성하는 에디터는 구문 에디터로서, 편집되는 객체는 계층적인 구조로 표현된다. 그러므로 이 프로세서가 필요한 문법은 XML 문서구조를 정의한 문법과 어떻게 화면에 보여줄 것인가를 정의한 문법으로, 본 연구에서 이를 추상문법[7]과 유형문법[7]으로 구분한다. 에디터에서 문서 편집은 정해진 규칙에 따라 top-down 방식으로 태그를 추가하면서 해당 문서를 작성한다. 이 작성된 문서를 처리하기 위한 내부적인 자료구조는 추상문법트리 구조로 하고, 이 트리 조작은 문서 트리조작기에서 이루어진다.

문서 트리 조작기는 XML 추상문법에 정의되어 있는 XML 구문에 따라 문서속성구문 트리(Document AST:DAST)를 조작 및 관

리하는 모듈이다. 이 모듈은 DAST 순환, 추가, 삭제, 그리고 어트리뷰트를 처리한다. 특히 DAST에서의 추가 및 삭제는 트리의 노드 위치에 따른 부분적인 처리를 위해 점진적인 트리 조작 방법을 사용한다.

명령어 해석기는 DAST와 사용자와의 관계에서 사용자에게 적당한 태그 정보를 제공하고 에디터에서 입력된 명령을 문서 트리 조작기에 전달하는 기능이 있다. 이 해석기는 문맥민감편집(context-sensitive editing) 기능에 따라 DAST의 노드 위치에 따라 사용할 수 있는 올바른 명령어 목록을 자동 제공한다.

언파서는 DAST의 화면 출력 형식을 처리하는 모듈로서, 생성된 DAST와 XML 유형문법을 이용한다. 동일한 DAST도 여러 가지 XML 유형문법에 의해 다양한 편집 화면을 구성한다. 특히 SDE와 같은 텍스트 편집기는 XML 문서의 시작태그와 끝태그를 자동적으로 출력하는 자동 태깅방식을 사용하여 문서 편집 시 사용자의 불필요한 입력을 최소화한다.

3. DTD 와 XML 문법

XML 문서에 포함되는 태그는 문서의 적절한 위치에 존재하므로써 구조를 나타내고 시작태그와 끝태그가 한 쌍을 이룬다. 이러한 태그의 위치는 DTD에서 정의된다. DTD는 엘리먼트 선언을 기본으로 어트리뷰트, 엔터티, 프로세싱선언등으로 이루어진다. 엘리먼트 선언은 태그 순서를 나타내는 문법선언이고 어트리뷰트 선언은 태그에 추가되는 속성정의 선언이다.

XML 문법은 DTD의 구문적 특성을 이

용한 형식 문법으로, 본 연구에서는 다중 뷰를 지원하는 구문 기반 문서 개발 환경을 지원하기 위하여 XML 문법을 추상문법과 유형문법으로 구분하였다. 추상문법[7]은 프로그램 구조를 정의한 것으로 추상구문트리를 어떻게 만들고 조작할 것인가를 나타내고 유형문법[7]은 프로그램의 출력형식을 정의한 것이다.

3.1. XML 추상문법

DTD의 엘리먼트 선언을 이용하여 구문 구조를 정의한 XML 추상문법은 문맥자유문법의 한 형태가 된다. 문맥자유문법처럼 XML 추상문법은 단말기호, 비단말기호, 시작기호 그리고 생성규칙으로 구성된다. 단말기호는 PCDATA 이고 비 단말기호는 엘리먼트명이 될 수 있으며 시작기호는 !DOCTYPE 뒤에 기술되거나 첫 번째 선언되는 엘리먼트 명이 된다. 그리고 생성규칙의 왼쪽에는 선언된 엘리먼트 명이 기술되고 오른쪽에는 엘리먼트 선언의 내용모형을 기술하면 된다. 그러나 XML 추상문법의 단말기호는 문서 내용 입력을 위한 문자열의 PCDATA 뿐 아니라 시작태그와 끝 태그도 단말기호가 되지만, 태그는 비 단말기호인 엘리먼트명에 의해 결정되는 특성이 있다.

[표 1]는 예제 DTD(a)를 문맥자유문법으로 표현한 추상문법(b)을 나타낸다. 본 연구에서 제시하는 XML 추상문법은 에디터를 생성하고 문서 편집을 유도하기 위한 구문정보로 사용하기 위한 것이므로, 엘리먼트명에 의해 결정되는 시작태그와 끝 태그는 단말기호로 분류하기보다는 비 단말기호가 내포하는 의미정보로 간주하여 (c)와 같은 형태로 재 정의한다.

어트리뷰트 선언은 엘리먼트의 특성을 나타내는 추가 정보를 정의한 것으로, 본 연구에서는 이를 일반 프로그래밍언어의 속성문법처럼 XML 추상문법의 속성값으로 처리한다. 프로그래밍언어의 속성문법에서는 파스 트리의 상하 노드간의 의존성에 따라 속성값이 계산 또는 상속되는 특성을 갖는 반면에, XML 추상문법의 어트리뷰트에 의한 속성값은 노드 자체의 특성을 정의하는 특징을 갖는다. 따라서 추상문법에서의 속성값들은 [표 2]와 같이 생성규칙 왼쪽 중괄호 { } 안에 열거한다.

$A \{ n1:(t1,dv1); \dots \}_{opt} \rightarrow a$
 $A \in V, a \in (V \cup N)^*$
 $n: name, t: type,$
 $dv: default-value$

[표 2] XML 추상문법과 속성값 표기 형식

(a)	(b)	(c)
<pre><!DOCTYPE a [<!ELEMENT c (#PCDATA)> <!ELEMENT b (#PCDATA)> <!ELEMENT a (b,c)>]></pre>	<pre>a → <a> b c b → PCDATA c → <c> PCDATA </c></pre>	<pre>a → b c b → PCDATA c → PCDATA</pre>

[표 1] DTD와 CFG로 표현되는 추상문법

예제 DTD	예제 DTD의 추상문법
<pre> <!DOCTYPE a [<!ELEMENT c (#PCDATA)> <!ELEMENT b (#PCDATA)> <!ELEMENT a (b,c)> <!ATTLIST a type CDATA "english" id ID #REQUIRED> <!ATTLIST b idref IDREF #REQUIRED>]> </pre>	<pre> a {type:(CDATA, english); id:(ID, #REQUIRED);} → b c b {idref:(IDREF, #REQUIRED);} → PCDATA c → PCDATA </pre>

[표 3] DTD와 변환된 r-attribute 속성을 갖는 XML 추상문법

엘리먼트에 대한 어트리뷰트는 없거나 그렇지 않고 존재한다면 (이름, 타입, 기본값)의 쌍이 하나이상 존재할 수 있으므로 $n1:(t1;dv1); n2:(t2;dv2);...$ 식으로 표시한다.

특히 어트리뷰트 타입이 ID, IDREF 또는 IDREFS은 이미 정의되어 어트리뷰트값을 확인하고 그 값을 재 사용하는 참조(reference)특성을 갖는다.

따라서 본 연구에서는 엘리먼트 선언을 이용하여 작성된 XML 문법을 r-attribute 속성값의 XML 추상문법이라 한다. [표 3]은 XML DTD를 r-attribute 속성을 갖는 XML 추상문법으로 변환한 것이다.

3.2. XML 유형문법

XML 유형문법은 추상문법을 기반으로 구조 에디터의 출력 형식을 정의한 것으로, 여러 가지 사용자 인터페이스를 지원한다. XML 구조 에디터는 문서 구조를 계층적인 형태로 나타내는 도구로 문서를 작성하기 템플릿을 이용한다. 구조 에디터에서 예약어의 포맷된 패턴과 플레이스홀더를 템플릿 [3]라 한다.

플레이스홀더는 실행문을 추가 삽입하기 위한 적절한 에디팅 위치를 나타낸다.

엘리먼트 명으로 구성되는 XML 추상문법과 다르게 XML 유형문법은 문서구문을 기초로 템플릿으로 구성된다.

$$A \rightarrow \chi A @_{sub-element} \chi' A$$

위의 유형문법은 2가지 종류 템플릿으로 이루어진다. 첫째, χn 와 $\chi' n$ 는 화면에 출력되는 엘리먼트 명의 디스플레이 포맷 정보를 나타내는 시각 템플릿로서, 문자열, 그래픽 또는 아이콘등을 나타내고, 둘째로 $@_{sub-element}$ 는 플레이스홀더 기능을 하는 내포 템플릿로서 문서 구조상 다음에 발생 가능한 부 엘리먼트를 나타낸다. [표 4]는 XML 추상문법과 그에 유형문법을 보여준다.

DTD 추상문법	DTD 유형문법
$a \rightarrow b c$	$a \rightarrow \chi_a @_b @_c \chi'_a$
$b \rightarrow PCDATA$	$b \rightarrow \chi_b @_{PCDATA} \chi'_b$
$c \rightarrow PCDATA$	$c \rightarrow \chi_c @_{PCDATA} \chi'_c$

[표 4] DTD 추상문법과 유형문법

1) 텍스트 에디터의 유형문법

SDE와 같은 텍스트 에디터인 경우는 시작태그와 끝 태그의 자동 태깅은 필수이고, 사용자에게 태그간의 연관성을 명확하게 표시해야 할 필요가 있으므로 계층적인 시각 표현이 요구된다. 그리고 입력커서 위치 또한 명확하게 표시할 필요가 있다. 따라서 χ_n 의 시각 템플릿은 시작태그명과 끝 태그명 그리고 제어문자로 이루어진 디스플레이 그룹정보를 구성하므로 다음과 같이 표현한다.

$$A \rightarrow \{ \chi_A \} @_{sub-element} \{ \chi_{A'} \}$$

그룹을 형성하는 시각 템플릿의 각 요소는 텍스트태그(" ")와 어트리뷰트필드(#), 탭(\t), 줄바꾸기(\n) 그리고 백탭(\b)와 같이 5가지이다. [표 5]는 임의의 텍스트 에디터를 생성하기 위한 XML 추상문법과 유형문법을 표현하였고, 그에 따른 에디터의 예상 화면

추상문법	유형문법	에디터 화면
$a \rightarrow b$	$a \rightarrow \{ "<a" ">" \n \t \} @_b \{ \n \ b " " \}$	<code><a></code> <code></code> [Celement]
$b \rightarrow c$	$b \rightarrow \{ "<b" ">" \n \ t \} @_c \{ \n \ b " " \}$	<code></code> <code></code>

[표 5] 텍스트기반 에디터에서의 XML 추상문법과 유형문법 그리고 에디터 출력

추상문법	유형문법	에디터 화면
$a \rightarrow b$	$a \rightarrow \text{Img}_a @_b$	
$b \rightarrow c$	$b \rightarrow \text{Img}_b @_c$	

[표 6] 트리 에디터에서의 추상문법과 유형문법 그리고 에디터 출력 예

의 일부분을 보인 것이다.

2) 트리 에디터의 유형문법

트리 편집기는 그래픽 형태의 한 예로서 문자태그 대신에 아이콘을 이용한다. 이 아이콘은 문서의 태그를 대신하는 것으로 유형문법에서 먼저 기술되는 한 개의 시각 템플릿만으로도 충분하다. 따라서 다음과 같은 유형문법 형식을 따르게 된다.

$$A \rightarrow \chi_A @_{sub-element}$$

트리 편집기를 위한 유형문법의 시각 템플릿은 적절하게 설계된 아이콘의 이미지 (img_A)를 사용하게 되므로 유형문법의 표현은 다음과 같다.

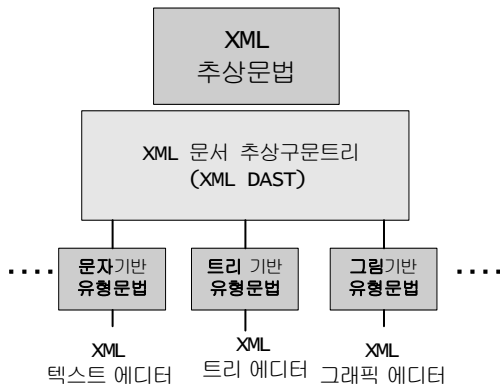
$$A \rightarrow \text{img}_A @_{sub-element}$$

이 시각 템플릿에 표현되는 아이콘은 마우스 이벤트와 팝업 메뉴처리가 필요하다.

[표 6]은 임의의 트리 에디터를 생성하기 위한 XML 추상문법과 그에 따른 유형문법을 나타낸 것이고 그리고 그에 따른 트리 에디터의 예상 화면의 일부분을 보인 것이다.

4. 구현 모델

본 연구에서 제시한 에디터 생성기는 XML 문법에 따라 동적으로 에디터를 자동 생성 하는 것으로, 여러 가지 에디팅 화면을 제공하는 다중 뷰 환경이 가능하다. 다중 뷰 환경은 에디터 화면의 출력 형식만 다를 뿐 편집되는 실제 문서 내용은 모두 같은 것으로, 에디터에서 원활한 문서편집과 에디터들 간의 일관성있는 문서 관리를 위하여 트리 자료구조를 선택하였다. 이 트리 구조는 편집된 XML 문서의 내부표현으로, XML문서 추상구문트리(DAST)라 한다.



(그림 4) DAST를 이용한 다중 뷰 지원 XML 구문에디터 구현 모델

그림 4는 여러 가지 유형문법을 이용한 다중 뷰 환경의 XML 에디터의 구현 모델이다. XML 에디터는 구조 중심의 에디터이므로 각 문서의 DTD에 의한 XML 문법을 사

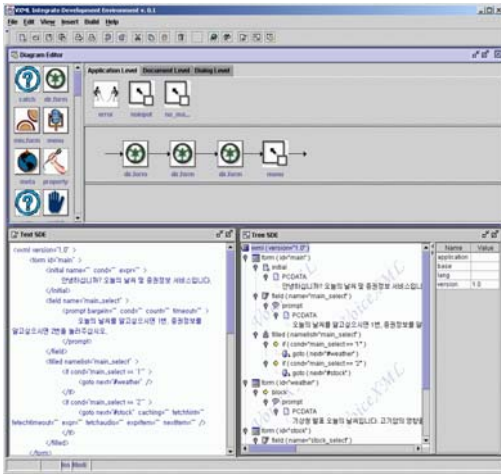
용한다. XML 문법의 추상문법은 문서의 구조를 정의한 것으로 에디터 환경의 기초가 된다. 이 추상문법은 XML 문서의 구조를 알려주는 안내 정보로서 에디터의 문맥민감 편집이 가능하다.

문서추상구문트리(DAST)는 XML 태그를 트리 노드 정보로 대응하여 만들어진 문서의 내부구조로서, 이는 본 연구에서 다루는 XML 문서의 표준 자료구조가 된다. DAST를 어떻게 화면 출력하느냐에 따라 구분되는 XML 에디터의 종류는 유형문법에 의해 좌우된다. 유형문법은 추상문법을 기반으로 문서 구조에 화면 디스플레이 정보를 포함하는 문법으로 첨가한다. 이 문법의 사용으로 인터페이스 사용 제한이 넓어지므로서 다양한 에디터의 생성은 물론 다중 뷰 환경까지도 가능하다.

다중 뷰 환경에 적합한 XML 에디터 생성기 구현하기 위하여 본 연구에서는 voiceXML[8]를 선택하였다. voiceXML은 전화를 이용한 음성-사용자 인터페이스를 생성하기 위한 XML응용언어로서, W3C의 XML을 기반으로 한다.

그림 5는 3개의 voiceXML 편집기로 구성되어 있는 것으로, 동일한 voiceXML 문서 내용이 서로 다른 편집기에서 비동기적으로 작동한다. 위쪽은 다이어그램편집기를 보여주고 아래 왼쪽은 구문편집기를 그리고 오른쪽은 트리편집기를 보여준다. 이 3개의 편집기는 모두 동일한 문서내용을 서로 다른 인터페이스로 출력하고 있다. 따라서 3개 편집기 중 하나를 이용하여 문서편집을 하게 되면 내부표현인 DAST가 변경되고 DAST 변경에 따라 각 에디터의 제어기들은 변경된 트리정보와 유형문법을 이용하여 변경된 내

용을 출력하게 된다. 그러므로 에디터들간의 문서 정보는 모두 동일하게 일관성이 유지된다.



(그림 5) 다중 VoiceXML 에디터

5. 결론

인터넷의 대중화로 대량의 문서 관리가 중요한 문제가 되면서 구조 중심의 XML을 이용한 문서처리가 증가하고 있다. XML은 구조에 의해 문서를 작성하는 웹 문서 작성 언어로서, 문서를 구성하는 태그는 DTD에 정의된 순서를 따른다. 따라서 DTD는 XML 문법이 된다. 올바른 XML문서를 좀 더 쉽게 작성하기 위하여, DTD에 정의되어 있는 구조 정보를 제공하는 XML 구문 에디터가 필요하다.

본 연구에서는 DTD에 독립적인 에디터를 동적으로 생성하는 에디터 생성기를 제시하였다. 이 생성기는 문법추출 프로세서와 맞춤형 에디터 생성 프로세서로 구성되는 문서개발환경으로, 에디터 생성에 필요한 구문과 화면 디스플레이 정보를 각각 추상문법과 유형문법으로 정형화 시켜 문서구조와 표현

에 있어 유연성과 효율성이 있는 XML 문서 환경을 제시하였다. 구문기반 개발 환경 특성을 갖는 XML 에디터 생성기는 단지 W3C 규칙에 따라 작성된 DTD 파일에서 추출된 추상문법을 기초로 처리되는 에디터를 생성하고, 유형문법을 이용하여 다중 뷰 시스템의 특성을 갖는 에디터 생성으로 다양한 편집 화면을 지원하였다. 따라서 본 XML 에디터 생성기는 구조에 대한 유연성과 다중 뷰 지원으로 다양한 편집환경을 제공하므로써 편집기에 대한 사용자의 접근이 용이해진다는 장점이 있다.

에디터 생성에 기본이 되는 XML의 추상문법과 유형문법에 대한 형식적인 정의가 필요하고 향상된 XML 문서개발환경에 대한 연구가 필요하다.

참고문헌

- [1] Donald J. Bagert Jr., Donald K. Friesen, "A Multi-Language Syntax-Directed Editor", *Proceedings of the 15th annual conference on Computer Science*, 1987.
- [2] Michelle Q. Wang, Allison Woodruff and Allan Kuchinsky, "Guidelines for Using Multiple Views in Information Visualization", *Proceedings of the Working Conference on Advanced Visual Interfaces 2000*, ACM Press.
- [3] Thomas W. Reps, Tim Teitelbaum, *The Synthesizer Generator*, Springer-Verlag New York Inc, 1989.
- [4] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, *Compiler:*

Principles, Techniques, and Tools, Addison-Wesley, 1986.

- [5] "Context free schemas for data-XML", <http://www.w3.org/People/Bos/Schemas/schemas>.
- [6] XML, <http://www.w3.org/XML>.
- [7] J.Linddkov Knudsen, M.Lofgren, O. Medsen, B.Magnusson, *Obect-Oriented Environments*, Prentice-Hall, 1993.
- [8] VoiceXML, <http://www.w3.org/TR/2001/WD-voicexml20-20011023>.
- [9] XML Pro v2.0, <http://www.vervet.com/>.
- [10] EXml Editor, <http://www.turbopower.com/products/xmlpartner/>.

유재우

숭실대학교 전자계산학과 학사(1976)
한국과학기술원 전산학과 석사, 박사(1978, 1985)
Cornell University 와 Univ. of Pittsburgh
Visiting Professor(1986, 1996)
KAIST인공지능센터 연구 교수(1987-94)
한국정보과학회, 논문지 편집위원(1997)
한국정보과학회, SIGPL 위원장(1999-2001)
한국표준협회 정보기술(IEC/ISO JTC1)
SC35(user interface) 국내 위원장(2001)
현재 숭실대학교 정보과학대학 컴퓨터학부
교수
<관심분야> 프로그래밍 언어, 컴파일러, 인
간과 컴퓨터 상호작용, 프로그래밍 환경, 시
각 프로그래밍 언어, SGML/XML 파서, 멀
티미디어 문서처리 객체지향 기술.

신경희

인하대학교 전자계산학과 학사(1985)
숭실대학교 컴퓨터학과 석사(1995)
숭실대학교 컴퓨터학과 박사 수료(2001)
<관심분야> 프로그래밍 언어, 프로그래밍
환경, 시각 프로그래밍 언어, SGML/XML
파서.