XML Toolkit 설계

노대식°, 김현기, 윤보현, 강현규
한국전자통신연구원 컴퓨터소프트웨어기술연구소 언어공학연구부
e-mail : {rohsik,hkk,ybh,hkkang@etri.re.kr

Design Specification of an XML Toolkit

Dae-Sik Roh°, Hyun-Ki Kim, Bo-Hyun Yun, Hyun-Kyu Kang
Dept. of Linguistic Engineering, ETRI-CSTLab.

요 약

XML 은 기존 HTML 의 한계를 극복할 수 있는 새로운 기술로 다양한 응용분야에 활용되고 있으며 많은 응용 제품들이 개발되고 있다. XML 편집기, XSL 편집기, XML 브라우저, XML 저장 관리기, XML 문서 저장 관리기, XML 문서 검색기, XML Conversion Tool 등의 다양한 XML 응용 프로그램에서 사용할 수 있는 표준 라이브러리 API 인 DOM(Document Object Model)과 SAX(Simple API for XML)를 지원하며 XML 문서의 모든 구성요소에 대한 처리를 할 수 있는 파서가 요구되고 있다. 이에 본 논문에서는 다양한 응용 프로그램의 요구사항을 분석하고 이를 반영하여 처리할 수 있는 XML Toolkit 모델을 제시한다. 본 XML Toolkit 은 W3C XML 1.0 스펙과 W3C Namespaces in XML 스펙과 W3C DOM Level 1 스펙을 지원하며 XML 사용자 그룹에서 정의한 SAX 를 지원한다. 또한 표준 API 로 접근할 수 없거나 그 기능이 표준에서 정의되지 않은 추가 기능을 제공하기 위한 XML 문서의 내부자료구조를 정의하고 이의 처리를 위한 API 를 제공한다.

1. 서론

XML(Extensible Markup Language)은 웹상에서 SGML(Standard Generalized Markup Language - ISO8879:1986)[1]의 사용을 보다 쉽고 간단하게 하기 위해 고안되었다. 즉, 문서타입들의 정의를 쉽게 하고, SGML 로 정의된 문서들의 저작과 관리를 용이하게 하고, 그것들을 웹상에서 쉽게 전달하고 공유하기 위해 고안된 것이다. XML 스펙문서에는 XML 은 아주 쉽고 간단한 SGML 의 방언(dialect)이며, XML 의 목적은 포괄적인 SGML 을 현재의 HTML(Hypertext Markup Language)처럼 웹상에서 서비스하고 수용하고 처리하는 것이다라고 정의되어 있다.[2]

XML Toolkit 은 XML 파서를 바탕으로 이루어지며 XML 문서를 처리하는 응용프로그램들에서 사용하는 공통의 API 를 제공하여야 한다. Toolkit 을 사용하여 작성할 수 있는 XML 응용프로그램들로는 XML 편집기, XSL 편집기, XML 브라우저, XML 저장 관리기, XML 문서 저장 관리기, XML 문서 검색기, XML Conversion Tool 등이 있다. 이와 같은 다양한 응용프로그램들에서 사용할 수 있는 표준 라이브러리 API 에 대한 요구가 증대됨에 따라 DOM(Document Object Model)[4], SAX(Simple API for XML)[5]와 같은 표준 API 가 정의되고 있다. 이에 본 XML Toolkit 은 응용프로그램들 간의 공용 API 로 정의된 DOM 1.0 과 SAX 1.0 을 지원하며 XML 문서의 모든 구성요소에 대한 처리를 할 수 있도록 하기위해 내부자료구조를 따로 정의하여 사용하였다.

2. 관련연구

2.1 관련 표준

DOM(Document Object Model)은 W3C(World Wide Web Consortium)에서 표준 문서로 제공하고 있으며 현재 버전 1.0 이 권고안으로 나와 있다. DOM1.0 에서는 HTML, XML 문서에 대한 표준 인터페이스를 정의하고 있으며 이 인터페이스에는 문서의 논리적 구조와 접근방법에 대한 API 를 정의하고 있고 HTML 문서를 위한 HTML DOM 과 XML 문서를 위한 XML DOM 을 각각 정의하고 있다.

SAX(The Simple API for XML)는 XML-Dev mailing list 회원들이 파서의 상호간의 표준 API 에 대한 필요성이 제기됨에 따라 만든 것으로 event-based XML 파싱

에 대한 공통 인터페이스를 제공하고 있다. 현재 많은 파서들이 SAX 1.0 스펙을 지원하거나 지원 계획 중에 있어 XML 파서의 표준 인터페이스로 여겨지고 있다.

2.2 파서의 종류.

XML 파서에서 지원하는 API 는 Tree-based API 와 Event-based API 의 두 종류가 있다. Tree-based API 는 메모리상에 문서 전체의 트리 구조를 만든 후 응용프로그램에서 사용할 수 있게 해 주는 방식으로 DOM 표준에서 채택되어 있는 방법으로 일반적인 XML 응용프로그램에서 사용될 수 있는 방법이다. Event-based API 는 메모리상에 문서에 대한 전체 트리를 만들지 않고 미리 등록된 이벤트에 대해 callback 함수를 만들어 사용하는 방법으로 대용량 대규모 문서에 대해 빠른 처리가 가능한 특징이 있다.

주요 XML 파서의 특징을 살펴보면 아래와 같다.

XP 파서 - James Clark : Well-formedness checking 기능을 지니고 있으며 Java 로 구현되어 있다. SAX1.0 을 지원하며 주요 기능으로는 외부 DTD 서브파일과 외부 매개변수 엔티티 및 외부 일반 엔티티에 대한 파싱 기능을 제공한다. XML 관련 표준에 대한 지원이 미흡하다.

XML4C – IBM : Validation checking 기능이 있으며 C++로 구현되어 있고 XML1.0, SAX, DOM 을 지원한다. 예제 및 소스가 공개되어 있으며 빠른 파싱 속도가 특징이다. 엔티티 및 엘리먼트 선언부에 대한 정보 추출기능이 없다.

JAXP – SUN: Valid 문서와 Well-formed 문서에 대한 checking 기능이 있으며 순수 Java 로 구현되어 있으며 SAX, Namespace 를 지원한다. Windows 환경에서는 한글폴더 처리에 문제가 있으며 비 Java 파서에 비해 속도가 느리다.

MSXML – Microsoft : Valid 문서 검증기능이 있으며 Java 와 C++구현되어 있다. XML1.0, DOM, XSLT, Namespace 를 지원하며 오류 메세지 출력기능이 있고 속도가 빠른 장점을 지닌다. 브라우져를 위한 파서로는 탁월한 성능을 보이지만, 편집기 등 다양한 XML 제품에 사용하기엔 API 가 부족하다.

3. Toolkit 요구사항 분석

본 XML Toolkit 은 W3C 표준으로 확정 발표된 안들 중 Extensible Markup Language (XML) 1.0[2], Namespaces in XML[3], DOM Level 1[4]을 지원하며 SAX Interface[5]를 지원하도록 한다. 활용분야로는 XML 전용편집기°, XML 브라우져, XML Repository Manager°, DTD 편집기, XML Information Retrieval System°, 전자도서관, 전자출판, XML Document Management System 등이 있다.

각각의 응용프로그램들에서 필요로 하는 기능들은 크게 다음과 같이 분류할 수 있다. 첫째, 모든 XML 구성요소에 대한 접근기능이 요구된다. 특정 엔티티의 선언부 정보를 요구할 수 있으며 DTD 에 선언된 엘리먼트들의 상하계층정보 등 다양한 XML 문서에 대한 접근기능을 요구한다. 둘째, 응용 프로그램별 특화된 API 가 요구된다. 편집기, 저장관리기, 검색기등 응용 프로그램에 따라 다양한 별도의 API 를 요구한다. 셋째, 사용하기 용이한 툴킷에 대한 요구가 증대되고 있다. 이것은 표준 API 에 대한 요구와 밀접한 관계가 있으며 널리 보편적으로 사용될 수 있는 API 를 제공하여 한다. 넷째, 윈도우 한글코드(KSC5601)지원을 요구한다. 일부 파서의 경우 한글폴더를 제대로 인식하지 못하는 경우가 있어 사용에 제한을 받는 경우가 발생된다. 다섯째, 지역적 특성을 반영한 부가 기능을 요구한다. 고부가가치의 제품을 생성하기 위한 기본 도구로 활용될 Toolkit 에 한국어 퇴고 기능과 같은 고급 기능을 부여함으로써 한차원 높은 제품을 개발할 수 있다. 여섯째, 표준 API 를 지원하여야 한다. 기본적인 XML 1.0 스펙뿐만 아니라 그와 관련된 각종 표준 API 를 지원하여야 한다. 일곱째, 빠른 파싱 속도를 요구한다. 이것은 모든 XML 구성요소에 대한 접근요구와 상충되는 면이 있지만, 대량의 XML 문서에 대한 처리를 위해서는 꼭 필요한 사항이다. 여덟째, Valid 한 XML 문서뿐만 아니라 Well-formed XML 문서를 지원하여야 한다.

4. XML Toolkit 설계

4.1 XML 모델

XML 문서는 크게 Prolog 와 Element 로 구분된다. Prolog 는 다시 XML 선언부와 XML 문서타입 정의부로 구성되며 XML 선언부는 버전정보, Encoding 정보, Standalone 정보로 이루어져 있고 XML 문서타입 정의부는 Element 선언부, Attribute 선언부, Entity 선언부, Notation 선언부. Processing Instruction, Comment, Parameter Entity Reference 로 구성된다. Element 정보는 Empty Element, Element, character data, Entity Reference/Character Reference, CData Section, PI, Comment 등이 나타난다.

Prolog 정보 중 XML 선언부는 파싱시 그 정보를 저장하였다가 API 를 통해 제공하고 XML 문서타입 정의부의 Element 선언부, Attribute 선언부, PI, Comment 정보는 Element 들간의 상호 계층관계를 표현한 트리로 나타낼 수 있으며 그 밖의 Entity 선언부, Notation 선언부는 테이블로 표현할 수 있다.

XML Element 정보는 루트 엘리먼트 이하의 트리로 모든 구성요소들을 표현한다.

4.2 XML 문서타입 정의부 모델

Element/Attribute/PI/Commnet 선언부 정보는 element 들의 트리로 표현된다. 각각의 노드는 Element, PI, Comment 및 오퍼레이터를 나타내고 각 노드는 Attribute 선언부에 대한 포인터 정보를 가진다. 트리상에서 중복을 없애기 위해 가상노드를 만들어 실제 노드와 연결한다.

4.2.1 노드정보

노드의 타입정보는 EMPTY_ELEMENT, ANY_ELEMENT, TAG_ELEMENT, COMMA_OPERA - TOR, OR_OPERATOR, QUESTION_OPERATOR, STAR_

OPERATOR, PLUS_OPERATOR, PCDATA_NODE, VIRTUAL_NODE, PI_ NODE, COMMENT_NODE, ENTITY_REF_NODE, CHAR_REF_NODE, CDATA_SEC_NODE 등이 있으며 이 타입정보는 XML 문서의 해당 구성요소와 대응된다.

노드의 링크정보로는 parent, child, previous, next 링크가 있으며 각 노드에는 타입정보, 노드 id, attribute 정보, text 정보 등을 저장하게 된다.

4.2.2 Attribute 정보

각 노드에 저장되는 Attribute 선언부 정보는 정의된 attribute 의 갯수에 따라 링크된 리스트로 저장한다. 각각의 Attribute 선언부 노드에는 attribute 이름, attribute Value Type 정보, Default Value Type 정보, Default Value 가 저장되며 Enumerated attribute 값이 별도로 저장된다.

4.2.3 Entity/Notation 정보

Entity 선언부와 Notation 선언부 정보는 테이블로 저장되며 Entity 테이블의 각 필드에는 엔티티이름, Data Type 정보, 텍스트, public identifier, system identifier, Declaration Type 정보, Notation 정보, internal 인지 external 인지의 정보를 저장한다. Notation 테이블은 Notation 이름, public identifier, system identifier 정보로 이루어져 있다.

4.3 XML 문서 Element 모델

XML element 정보는 루트 element 이하의 완벽한 트리로 표현할 수 있다. 4.2 절의 문서타입 정의부 모델의 노드와 대응되는 노드들을 생성하여 트리로 구성한다. 각 노드는 대응되는 4.2 절의 노드가 있으며 parent, child, prev,next 링크를 가지며 attribute 의 값을 저장하기 위한 Attribute Value 노드가 있으며 실제 #PCDATA 에 해당하는 데이터를 저장한다.

4.3.1 ID/ID REF 모델

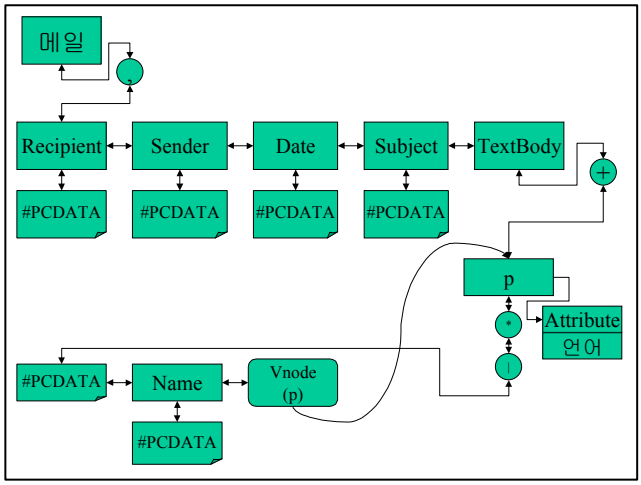
ID/IDREF 는 어트리뷰트의 값에 의해 참조되는 Element 노드들의 관계를 표현 저장하기 위한 것으로 별도의 테이블에 정보가 저장된다. 각 필드는 ID, Reference Count, definition 노드에 해당하는 Element 노드와 실제로 참조하는 노드들의 리스트를 저장하는 필드로 구성된다.

4.4 XML 문서에 대한 선언부 트리 구성 예

예제.xml

```
<?xml version="1.0"    encoding="KSC5601"?>
<!ELEMENT  메일  (Recipient, Sender, Date, Subject, TextBody) >
<!ELEMENT Sender       (#PCDATA)          >
<!ELEMENT Recipient        (#PCDATA)          >
<!ELEMENT Date    (#PCDATA)          >
<!ELEMENT Subject          (#PCDATA)          >
<!ELEMENT TextBody         (p)+               >
<!ELEMENT p                (#PCDATA|Name|p)* >
<!ELEMENT Name    (#PCDATA)          >
<!ATTLIST p     언어          (de|en|한국) "en"      >
```

결과 선언부 트리



5. 결론

본 논문에서는 XML Toolkit 에 대한 요구사항을 분석하였으며 모든 XML 구성요소에 대한 접근기능을 부여하기 위해 XML Element 정보 및 XML Prolog 정보를 트리와 테이블을 이용하여 정보를 저장하고 응용프로그램에 따라 특화된 API 를 제공하도록 하는 모델을 제시 하였다. 본 Toolkit 은 표준 API 인 DOM 과 SAX 를 지원하며 한국적 특성을 반영한 한국어 퇴고 기능과 같은 부가 기능을 구현할 예정이다.

6. 참고 문헌

[1] Charles F. Goldfarb, *The SGML Handbook*, Oxford Univ. Press, 1990.

[2] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, “W3C Extensible Markup Language(XML) 1.0”, W3C REC-xml- 19980210, See http://www.w3.org/TR/1998/REC-xml-19980210.html

[3] Tim Bray, Dave Hollnder, Andrew Layman, “W3C Namespaces in XML”, W3C REC-xml-names-19990114, See http://www.w3.org/TR/REC-xml-names/

[4] Vidur Apparao, Steve Byrne, etc., “Document Object Model(DOM) Level 1 Specification”, W3C REC-DOM-Level-1-19981001, See http://www.w3.org/TR/REC-DOM-Level-1/

[5] The members of the XML-DEV mailing list, “SAX 1.0 : The Simple API for XML”, Megginson Technologies, See http://www.megginson.com/SAX/

[6] 노대식, 김현기, 강현규. “RMOX : SGML/XML 문서 부분편집기”, ‘99 춘계 정보처리 학술대회, pp80~83, 1999