제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

메타데이타 관리를 위한 RDF 기반의 저작 툴 개발

조성훈[0], 이무훈, 조현규*, 송병렬*, 김동혁, 이찬섭, 최의인
한남대학교 컴퓨터공학과
*한국전자통신연구원
e-mail:{shcho[0], mhl, dhkim, cslee, eichoi}@dblab.ac.kr
{hkcho, sby}@etri.re.kr*

Development of Authoring Tool Based on RDF for Metadata Management

Sung-Hoon Cho[0], Moo-Hoon Lee, Hyun-Kyu Cho, Byoung-Youl Song, Dong-Hyuk Kim, Chan-Seob Lee, Eui-In Choi
Dept. of Computer Engineering, Hannam University
Electronics and Telecommunications Research Institute*

요 약

정보 인프라의 확장과 더불어 웹 이용의 보편화로 인해 오프라인(off-line)과 온라인(on-line)상에는 많은 데이터가 산출되고 있는 상황이다. 그러나 데이터 관리를 위한 메타데이타 표준이 특정 도메인(domain)에 제한적이거나 너무 광범위하게 정의되어 있기 때문에 효율적인 데이터 관리가 되지 못하고 있다. 또한 생성한 메타데이타에 대한 유효성 검증이 되지 않으므로 정확한 메타데이타인지를 보장할 수 없는 실정이다. 따라서 본 논문에서는 메타데이타에 대한 유효성 검증을 수행하고, RDF(Resource Description Framework)를 기반으로 메타데이타를 효율적으로 저작할 수 있는 저작 툴과 웹 자원(resource)를 N-triple로 표현하여 데이터를 관리할 수 있는 N-triple 생성기를 개발하였다.

1. 서론

웹(Web) 사용이 보편화되고 웹과 직·간접적으로 관련된 데이터가 급속히 증가하면서 효율적으로 데이터를 관리할 수 있는 기술의 필요성이 크게 요구되었다. 기존에 데이터 관리를 위해 사용되던 데이터베이스의 경우 특정 도메인(domain)에 의존적으로 스키마를 미리 정의하여 사용해야한다는 점과 각각의 데이터에 대한 의미(semantic)를 정의하거나 이해할 수 없다는 한계를 가지고 있어 이로 인한 동일한 의미의 데이터 중복 처리 문제가 발생된다.

이와 같은 문제를 해결하기 위해 W3C(World Wide Web Consortium)에서는 웹 상의 자원을 효율적으로 관리할 수 있는 메타데이타 표준으로서 RDF(Resource Description Framework)를 제안하였다[1,2,3]. RDF는 최근 웹 분야에서는 대두되고 있는 Semantic Web의 기반으로, 기존의 Dublin Core와 Warwick Framework가 가졌던 한계를 해결하고, 기계가 자원의 의미를 이해하고 관리할 수 있는 메타데이타 표준이다[4,5,6,7].

그러나 RDF를 이용하여 데이터를 관리한다고 하여도 RDF 기반의 문서가 유효한지에 대한 유효성은 보장할 수 없다. 이는 RDF 문서 자체에 대한 유효성 검사가 저작 시점에서 수행되지 않기 때문이며, 따라서 웹 자원이 더 이상 유효하지 않게 된다.

본 논문에서는 RDF와 RDFS(Resource Description Framework Schema)를 기반으로 RDF 문서를 저작할 수 있는 편집기, 유효성 검증이 가능한 검증기(validator), 자원의 관계를 쉽게 이해할 수 있는 N-triple 생성기를 설계 및 구현하였다[4,5].

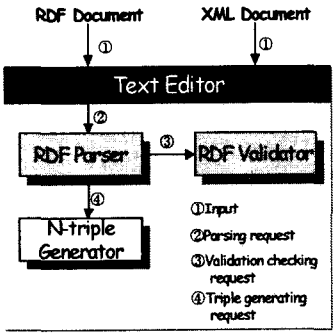
논문의 구성은 2장에서 저작 툴(Tool)의 구성과 기능을 알아보고, 3장에서 RDF 문서의 유효성 검증을 위한 파싱(parsing) 및 유효성 검증 모듈(validation checking module)을 설명한 뒤, 마지막

* 본 논문은 한국전자통신연구원의 "ebXML 메타정보 저작 도구 개발"의 연구 결과임

으로 4장에서 결론과 향후 연구 내용을 기술하도록 한다.

2. RDF 기반 저작 툴의 구성 및 기능

저작 툴의 구성은 RDF 및 XML 문서의 처리 과정에 기초하여 구성하였으며, 텍스트 편집기, 파서(parser), 유효성 검증기(validator), n-triple 생성기로 나누어 볼 수 있다. 이에 대한 구성도는 [그림 1]과 같다.



[그림 1] 저작 툴의 구성도

[그림 1]에 있는 각각의 모듈(module)들은 다음과 같은 기능을 갖는다.

■ Text Editor

일반적인 라인 편집기와 같은 기능을 제공하면서 XML과 RDF 구문에 대해 하이라이팅(highlighting)을 제공하여 Element, Attribute, Character등에 대해 가시적으로 명확하게 구분할 수 있는 모듈이다.

■ RDF Parser

사용자의 파싱 요청에 의해 RDF 문서에 대한 파싱을 수행하는 모듈이다. 이 모듈의 특징은 RDF 문서에 제한적이지 않다는 것이다. 이는 파싱 과정에서는 문서의 컨텐츠(contents)에 대한 의미 해석을 수행하지 않고 파싱되는 문서 안이나 외부에 정의되어 참조되는 DTD(Document Type Definition)나 XML Schema에 대한 유효성 검증을 수행하지 않기 때문이다. 이와 같은 특징으로 인해 XML 기반의 모든 문서들에 대해 파싱을 수행할 수 있다. 파싱 처리의 결과는 문서의 well-formedness이다. 즉, 문서가 XML 구문에 맞게 저작되었는지에 대한 결과로서 구문 오류(syntax error)를 출력하게 된다.

■ RDF Validator

사용자의 유효성 검증 요청을 통해 파싱을 수행한 뒤에 RDF 문서에 대한 유효성 검증을 수행한다. 파싱 과정에서 간과되었던 참조된 DTD나 XML Schema에 선언된 Elelment나 Attribute들과 구문 구조에 대한 매칭(matching)을 시도하여 문서가 DTD나 XML Schema에 정의된 대로 작성되었는지를 검사하게 된다. 유효성 검증 역시 RDF 문서뿐만이 아니라 XML 기반의 문서 전반에 대해 수행될 수 있는 공통 모듈이라 말할 수 있다. 이는 문서에 있는 XML 컨텐츠에 대한 구문 구조 및 선언에 대한 검사만을 수행하지 컨텐츠에 대한 의미 해석을 시도하지 않기 때문이다. 유효성 검증 결과는 well-formedness와 구문 오류이다. 구문 오류는 DTD나 XML Schema에 매칭되지 않는 데이터 타입(data type), 순서(sequence), 구문 구조 등에 대한 정보이다.

■ N-triple 생성기

사용자의 N-triple 생성 요청을 통해 파싱을 수행한 뒤 N-triple 생성 요청을 함으로서 동작된다. 이 모듈은 RDF 문서에 제한적인 특징을 가지고 있으므로 RDF 문서가 아닌 문서에 대해 N-triple 생성 요청을 할 경우에 오류를 발생시킬 수 있다. 이러한 이유는 N-triple Generator가 RDF에 정의된 구문을 토대로 N-triple을 생성하기 때문이다. 수행한 결과는 Subject-Predicate-Object(또는 Literal) 형태를 가진 테이블이다[4].

3. RDF 문서의 파싱 및 유효성 검증 모듈

3.1 파싱 모듈

파싱은 2장에서 언급한 바와 같이 문서의 well-formedness를 검사한다. 파싱을 수행하기 위한 필요조건은 파싱될 XML 기반의 문서와 문서의 URL(Uniform Resource Locator)이다. 문서가 로컬 시스템 상에 존재한다고 해도 시스템 파일 경로 표기법과는 별도로 반드시 URL이어야만 한다. 파싱은 SAX(Simple API for XML)를 기반으로 한다. SAX를 이용하는 이유는 처리 성능이 좋고 부하(load)가 DOM(Document Object Model)에 비해 작기 때문이

다. Java를 이용하여 SAX 파서(parser)를 구현할 경우에, 기본적으로 ContentHandler, ErrorHandler라는 2개의 핵심 인터페이스(interface)가 필요하다.

■ ContentHandler 인터페이스

XML 문서를 파싱 과정에서 Document, Element, Character, Processing Instruction을 만났을 때 각각에 대한 파싱 처리 메소드를 정의하는 인터페이스로서 다른 처리 과정을 연동하여 처리할 수 있다.

■ ErrorHandler 인터페이스

ContentHandler 인터페이스 분석 과정에서 발생할지 모르는 다양한 예외(Exception)를 처리하는 역할을 수행한다. 에러는 warning, error, fatal error로 나뉘며 정의된 예외 처리 메소드에 의해 처리된다. [그림 2]는 SAX를 이용한 파싱 알고리즘이다.

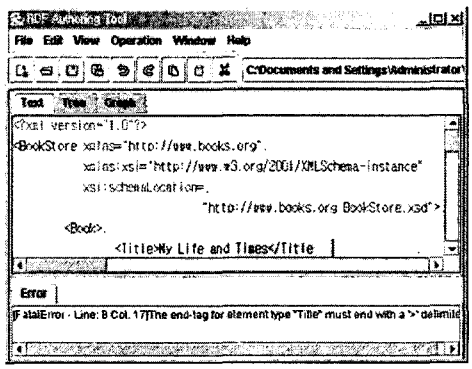
```
void Parsing() {
InputSource S;
XMLReader P;
ErrorHandler errs = new ErrorHandler();
ContentHandler cont = new ContentHandler();
Reader r = new StringReader(rawdata);
try {
S = new InputSource(r);
P = XMLReaderFactory.createXMLReader
        ("org.apache.xerces.parsers.SAXParser");
P.setErrorHandler(errs);
P.setContentHandler(cont);
①P.setFeature("http://xml.org/sax/features/validation", false);
②P.setFeature("http://apache.org/xml/features/continue-after-f
atal-error",true);
③P.setFeature("http://apache.org/xml/features/nonvalidating/lo
ad-external-dtd",false);
S.setSystemId(new File(sysid).toURI().toASCIIString());
P.parse(S);
} catch (SAXException e) {
  ④System.out.println(e.getMessage());
} catch (IOException e) {
  ⑤System.out.println(e.getMessage());
}
```

[그림 2] SAX를 이용한 파싱 알고리즘

[그림 2]의 알고리즘에서 setFeature() 메소드는 파서에 대한 설정 부분으로서 ①은 유효성 검사를 수행하지 않도록 하며, ②는 예외가 발생하더라도 파싱을 끝까지 수행하도록 하고, ③은 외부 DTD를 로드(load)하지 않도록 설정한 것이다. ④는 ContentHandler에서 문서에 대한 파싱을 수행하는 동안 발생한 예외들을 출력한다. ⑤는 RDF 또는 문서의 입출력에 관련된 예외로서 파싱과는 크게 연관은 없다. [그림 3]은 RDF 문서를 파싱한 결과이다.

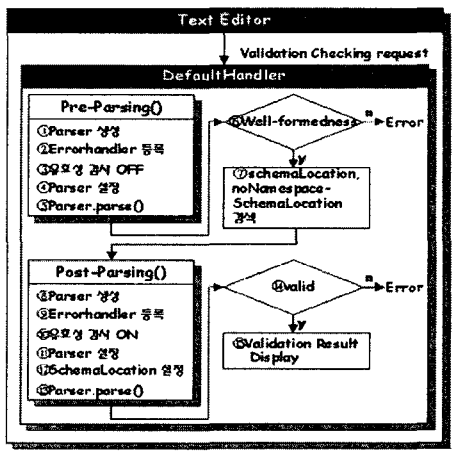
3.2 유효성 검증 모듈

RDF 문서의 유효성을 보장하기 위해 반드시 필요한 것이 유효성 검증이다. 이 모듈은 내부적으로 파싱 모듈을 포함한 형태를 취하고 있으며, 전체적인 구성은 파싱 모듈과 유효성 검증 모듈 각각 하나



[그림 3] RDF 문서의 파싱 예

로 되어 있다. 이와 같이 구성되는 이유는 참조된 DTD나 XML Schema가 있을 경우, 유효성 검증시 유효성 검증 모듈을 실행하기 전에 DTD의 문서 타입 선언과 XML Schema 참조를 위한 xsi:schemaLocation이나 xsi:noNamespaceSchemaLocation을 미리 설정하여야 하기 때문이다.

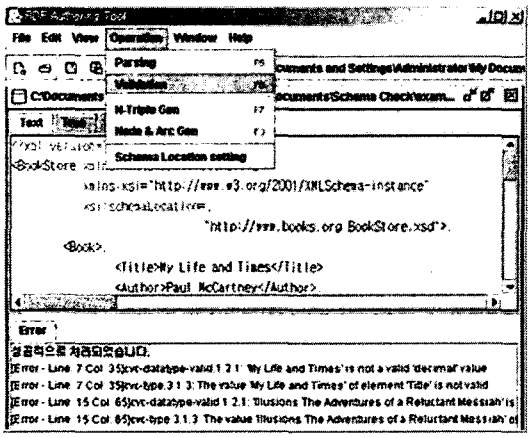


[그림 4] 유효성 검증 모듈의 처리도

유효성 검증 모듈은 [그림 4]와 같이 pre-parsing

과 post-parsing 메소드를 통해 처리된다. pre-parsing은 유효성 검증은 수행하지 않고 well-formedness만을 검사한다. 또한 문서가 XML Schema에 대한 참조를 정의하고 있을 경우에 xsi:schemaLocation이나 xsi:noNamespaceSchema-Location 애트리뷰트(Attribute)의 유무를 검사한 뒤 해당 애트리뷰트가 존재할 경우에 애트리뷰트값(Attribute value)을 post-parsing 메소드의 인자로 전달하게 된다. 이 때 전달되는 인자는 namespace URL과 XML Schema의 URL로서 상대(relative) 및 절대(absolute) 경로로 표현된다.

post-parsing 메소드는 인자로 받은 데이터를 유효성 검사에서 참조할 XML Schema의 SchemaLocation으로 설정한 뒤에 parse() 메소드를 호출하여 유효성 검사를 수행한다. [그림 5]는 schemaLocation을 포함한 XML 문서에 대해 유효성 검사를 수행한 결과이다.



[그림 5] schemaLocation을 포함한 XML 문서의 유효성 검사 수행 예

3.3 N-triple 생성 모듈

N-triple Generator는 RDF 문서에 기술되어 있는 자원의 관계(relation)를 명확히 구분할 수 있도록 자원간의 관계를 표현하는 모듈이다. 자원간의 관계는 파싱 모듈에서 사용되었던 SAX 기반의 parser를 이용하며 RDF에서 정의한 요소(Element 또는 Attribute)들을 만나게 되면 이를 ContentHandler에서 Subject, Predicate, Object따라 각각의 배열에 저장함으로서 N-triple을 생성한다. N-triple 생성 결과는 3개의 배열을 순회하여 출력한 것이다.

4. 결론 및 향후 과제

본 연구는 RDF 기반의 메타데이타 문서를 효율적으로 생성할 수 있는 저작 환경과 XML 기반의 문서에 대한 유효성 검증을 제공하여 웹 상의 자원 관리에 필요한 메타데이타 문서의 유효성을 보장할 수 있다. 이를 이용할 경우에 전자상거래 분야의 전자카탈로그를 기술하는 메타데이타를 효율적으로 관리할 수 있을 것이다.

RDF 문서의 저작 방법은 텍스트(text) 방식 이외에도 트리(tree), 노드 앤 아크 다이어그램(Node and Arc Diagram)등이 있다. 특히 노드 앤 아크 다이어그램의 경우 다이어그램으로 각 자원과 자원간의 관계를 표현하기 때문에 가시성이 높고 이해하기 쉽다는 장점을 가지고 있다. 따라서 본 연구의 추후 연구과제로는 노드 앤 아크 다이어그램을 이용하여 RDF 문서의 저작 환경을 개발하는 것이다.

참고문헌

[1] W3C, Resource Description Framework (RDF), February, 1999, http://www.w3.org/RDF/

[2] RDF Vocabulary Description Language 1.0, April, 2002, http://www.w3.org/TR/rdf-schema/

[3] RDF: Understanding the Striped RDF/XML Syntax, Dan Brickley, October, 2001, http://www. w3.org/2001/10/stripes/

[4] Jose Kahan, Marja-Ritta Koivunen: Annotea: an open RDF infrastructure for shared Web annotations. WWW 2001: 623-632

[5] Decker, S.; Melnik, S.; van Harmelen, F.; Fensel, D.; Klein, M.; Broekstra, J.; Erdmann, M.; and Horrocks, I. 2000. The semantic web: The roles of XML and RDF. IEEE Internet Computing Sept-Oct:63-74.

[6] Jeen Broekstra, Michel Klein, and Stefan Decker, Enabling knowledge representation on the Web by extending RDF Schema, In Proceedings of the 10th World Wide Web conference, pg. 467-478, Hong Kong, China, May 1-5, 2001.

[7] Olivier Corby, Rose Dieng, and Cedric Hebert, A Conceptual Graph Model for W3C Resource Description Framework, ICCS2000, Darmstadt, Germany, August 14-18, 2000.