RDF 기반 메타데이타 문서를 위한 유효성 검증 가능한 툴 개발

조성훈, 김동혁, 조현규*, 송병렬*, 이무훈, 최의인
한남대학교 컴퓨터공학과
*한국전자통신연구원

Verifying Validation for Metadata Document In RDF

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

요 약

인터넷의 보급으로 인해 수많은 데이터가 생성되고 있고 이러한 데이터를 효율적으로 관리하고자 메타데이타 표준이 수차례에 걸쳐 발표되었다. 그러나 기존 발표된 표준들은 데이터 관리를 위해 특정 도메인에 제한적이거나 너무 광범위하여 효율적이지 못하다는 문제를 가지고 있다. 또한 표준에 따라 생성된 메타데이타라 할지라도 유효성을 검증하지 못하기 때문에 정확한 메타데이타로 정의내릴 수 없는 실정이다. 따라서 본 논문에서는 RDF(Resource Description Framework)에 기초하여 메타데이타를 효율적으로 저작할 수 있는 저작 기능, 생성된 메타데이타에 대한 유효성 검증이 가능한 유효성 검증 기능, 메타데이타의 N-Triple 표현을 지원하고 생성할 수 있는 N-Triple Generator를 지원할 수 있는 RDF 기반 저작 툴을 개발하였다.

1. 서론

웹의 출현 이전에도 많은 데이터들이 메타데이타에 의해 분류되고 관리되어 서지학이나 도서관 등과 같은 특정 분야에서 사용되었다. 그러나 인터넷의 확대와 정보통신의 비약적인 발전으로 더 이상 적절하지 못하게 되었다. 이는 네트워크 환경이 구축됨에 따라 웹에서 데이터 교환이 쉽고 빠르게 할 수 있게 되었고 상호 교환된 정보에 기반하여 새롭게 생성되는 데이터의 양이 기하급수적으로 증가하기 때문이다. 또한 새로운 데이터는 기존 데이터에 근거하거나 완전히 새로운 분류로 나누어지거나 인터넷 환경에서의 교환과 공개를 위한 형태를 가지게 되므로 다양한 종의 문서와 데이터가 생성되었기 때문이다. 이러한 다양한 분류를 가지는 특성과 인터넷에서의 정보 공유를 위한 데이터는 기존의 특정 도메인(domain)에 한정되었을 때와는 달리 관리 분야에 대해 현격한 차이가 있어서 범도메인적인 관리 기준이 필요하게 되었다.

이를 위해 제안된 개념이 바로 메타데이타(Meta-Data)이다. 메타데이타란 "데이터의 데이터"를 의미하는데, 데이터를 관리하기 위한 데이터를 의미하며, 더블린 코어(Dublin Core)와 같은 메타데이타 표준이 제안되어 사용되고 있다. 그러나 더블린 코어는 그 목적을 단순한 구문 구조를 이용한 데이터의 표현과 관리라는 점과 웹 자원의 관리에 적절하지 못하다는 문제를 가진다. 이러한 한계를 극복하고자 새롭게 제안된 것이 워릭 프레임워크인데, 이 표준은 상호운용성을 위한 컨테이너 구조를 제안하였다. 그러나 실제적인 상호운용성을 위한 메카니즘을 정의하고 있지 않고 있다[1,2,3].

이와 같은 문제를 해결하기 위해 W3C(World Wide Web Consortium)에서 웹 자원을 관리하고자 RDF(Resource Description Framework)를 제안하였다[1,2,3]. RDF는 시멘틱 웹(Semantic Web) 기반으로,

* 본 연구는 한국전자통신연구원의 "시맨틱 비즈니스 문서편집기 개발"의 연구 결과임

기존의 더블린 코어와 워릭 프레임워크의 문제를 해결하고, 기계가 자원의 의미를 이해하고 관리할 수 있는 메타데이타 표준이다[4,5,6,]. 그러나 RDF는 데이터 관리를 목적으로 하지 데이터의 유효성에 대한 보장은 하지 않는다. 따라서 유효하지 않은 데이터로 웹 자원을 관리하게 되는 문제가 발생할 수 있게 된다.

이와 같은 문제를 해결하고 웹 자원의 유효성 검증을 위해 본 논문에서는 RDF 메타데이타 표준에 기반하여 웹 자원을 효율적으로 저작할 수 있는 편집기, 자원 기술의 유효성 검증을 위한 검증기(validator), 자원간 관계를 쉽게 정의할 수 있는 N-triple(이하 n3) 생성기를 설계 및 구현하였다.

논문의 구성은 2장에서 RDF 저작 툴의 구현 환경과 구성에 대해 알아보고, 3장에서 저작 툴의 모듈별 기능을 설명하고, 4장에서 저작 툴의 처리 루틴에 대해 논한 뒤 마지막 5장에서 결론과 향후 연구 내용을 기술하도록 한다.

2. RDF 저작 툴의 구현 환경과 구성

2.1 구현 환경

본 논문에서는 RDF 저작 툴의 구현을 위한 프로그래밍 언어로 Sun Microsystems 사에서 개발한 J2SE v1.4를 채택하였다. 또한 RDF 모델을 처리를 위해 RDF 및 OWL에 기반한 시맨틱 웹(semantic web) 응용프로그램 개발을 위한 툴킷인 Jena v2.1을 이용하였고, Xerces 2 Java Parser 2.0.2 Release와 JAXP(Java API for XML Processing) v1.2를 API로 이용하여 파서를 구현하였다.[6] 이와 같이 Java에 기반한 구현 환경을 구축함으로서 운영체제에 독립적으로 실행 가능하게 되었다.

2.2 RDF 저작 툴의 구성

RDF 기반 저작 툴은 그림 1과 같이 크게 다섯 부분으로 나눌 수 있다. 각각은 RDF를 이용한 웹 자원 기술을 위한 편집기, RDF 문서를 파싱하는 파서, n3를 생성하는 n3 생성기, RDF 문서에 기술된 웹 자원에 대한 유효성 검증을 수행하는 검증기, 파싱 및 유효성 검증 결과를 출력하는 결과 출력기로 구성되어 있다.

편집기의 경우에는 웹 자원 기술의 효율적인 저작을 위해 텍스트, 그래프, 트리에 기반한 저작 환경을 지원한다. 각 편집 환경은 상이하지만 처리하는 데이터는 동일하게 유지하며, 편집 환경의 변화에 따라 자원 표시 형태를 자동으로 변경할 수 있도록 구성하였다.

파서는 SAX, DOM, ARP 파서로 구성되어 있는데, 이는 각기 사용되는 경우에 따라 선택적으로 실행된다. ARP 파서는 RDF 문서의 파싱에만 사용되는데, 이는 파싱 성능을 높이기 위한 구성이다. SAX 파서는 XML 문서의 파싱을 위한 것이고, DOM 파서의 경우는 n3의 생성, 트리 편집기에서 RDF 문서의 구조적 정보를 요구하기 때문에 이를 지원하기 위한 파서이다. n3 생성기는 RDF 문서의 구조적 정보와 웹 자원의 기술형태를 n3로 변환 생성한다. RDF 검증기는 RDF 문서를 RDFS에 기술된 스키마와 비교하여 유효성을 검증한다. 마지막으로 결과 출력기는 파싱, n3 생성, 유효성 검증 결과를 사용자에게 지원하기 위한 것이다.

이와 같은 구성을 통해 RDF 기반의 웹 자원 기술 효율성을 극대화하도록 하였다.

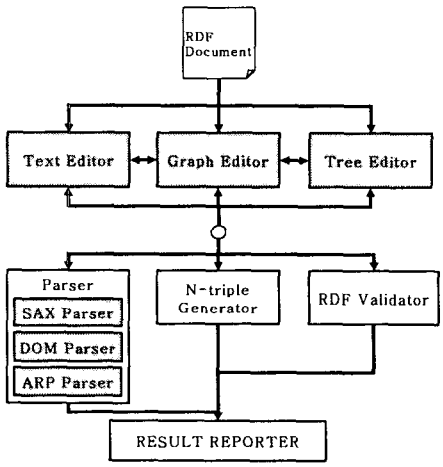

그림 1. RDF 저작 툴의 구성도

3. RDF 저작 툴의 모듈별 기능

RDF 저작 툴은 그림 1에 제시된 바와 같이 텍스트 편집 모듈, 그래프 편집 모듈, 트리 편집 모듈, 파서 모듈, n3 생성 모듈, RDF 유효성 검증 모듈, 결과 출력 모듈로 나눌 수 있다. 각 모듈 중 편집 모듈의 경우는 웹 자원의 효율적인 기술을 위해 상호 연관되어 있다. 또한 파서, n3 생성기, RDF 유효성 검증기도 마찬가지로 처리 루틴에 대해 상호 연관되어 있다. 각 모듈에 대한 세부적인 기능 설명은 아래와 같다.

3.1 텍스트 편집 모듈

라인 편집 인터페이스를 이용하여 웹 자원을 기술하며 XML 구문 정의에 따라 구문 강조를 통해 표현하여 가시적인 자원 구분이 가능하게 한다.

3.2 그래프 편집 모듈

노드와 아크 다이어그램(Node & Arc Diagram)을 이용하여 웹 자원을 기술한다. 이 때의 각 자원 형태는 Subject, Predicate, Object(Literal)로 구분하며, 상호간의 관계를 표현하기 위해 아크를 이용한다. 또한 하위 노드에 대해서 상위 노드들의 경로를 추적하는 Trace 기능을 지원하여 웹 자원의 참조 경로를 알 수 있다. 그림 2는 그래프 편집 모듈의 실행화면이다.

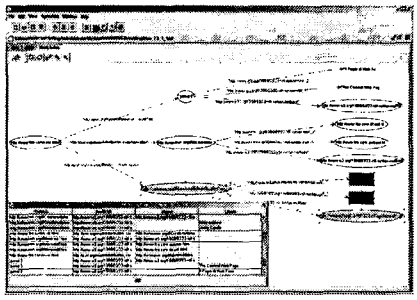

그림 2. 그래프 편집 모듈의 실행화면

3.3 트리 편집 모듈

트리 편집 모듈은 RDF 문서를 DOM 파서로 실행하여 DOM document를 생성한 후 트리 형태로 변환하고, 트리 노드에 웹 자원을 기술하는 편집 모듈이다. 노드 편집에 대해 삽입, 삭제, 변경에 대해 구문을 검사하여 유효한 형태가 아니라면 입력되지 않도록 구현되어 있다.

3.4 파서 모듈

파서 모듈은 RDF 문서에 대한 파싱을 수행하는 모듈로서 문서의 컨텐츠에 대한 의미 해석을 수행하지 않고 파싱될 문서의 내외부에 정의된 DTD(Document Type Definition)나 XML Schema에 대한 유효성 검증은 수행하지 않는다. 단순히 문서의 well-formedness만을 검사하고 그 결과를 결과 출력 모듈에 전달한다.

3.5 n3 생성 모듈

RDF를 기반으로 기술된 웹 자원을 Subject-Predicate-Object(Literal)의 형태의 테이블로 변환하여, 자원을 기술한다[4]. n3 생성시의 오류 결과는 결과 출력 모듈에 전달되어 오류 형태를 사용자에게 보고한다.

3.6 RDF 유효성 검증 모듈

RDF 문서에 대한 유효성 검증을 수행한다. 이 모듈에서는 파싱 과정에서 검사되지 않은 DTD나 XML Schema에 대한 구문 정의와 웹 자원 기술에 사용된 구문들을 매칭을 수행하여 DTD나 Schema에 정의된 대로 작성되었는지를 검사한다. 처리 결과는 문서의 well-formedness와 구문 정의와 맞지 않은 데이터 타입, 순서, 구문 구조에 대한 구문 오류이다.

3.7 결과 출력 모듈

결과 출력 모듈은 파싱, 유효성 검증, n3 생성에 관련된 처리의 성공와 실패에 대한 정보를 각 모듈에서 전달받아 출력한다. 단순히 오류 보고의 기능을 수행한다.

4. RDF 저작 툴의 처리 루틴

본 절에서는 RDF 저작 툴의 처리 과정에 대해 상세히 설명한다. 먼저 RDF 저작 툴은 문서 편집, 파싱, 유효성 검증, n3 생성 과정으로 분류할 수 있다. 이 중 편집 과정은 일반적인 데이터 입력이므로 세부적 처리 사항은 언급하지 않고, 파싱, 유효성 검증, n3 생성 처리 루틴에 대해서만 논하도록 한다.

4.1 파싱 처리 루틴

파싱 처리 루틴은 RDF/XML 문서의 구문 검사, 즉 well-formedness 검사를 수행한다. 이 처리 루틴에는 DTD나 XML Schema에 대한 유효성 검사는 수행하지 않는데, 세부 처리 루틴은 그림 3과 같다.

먼저 편집 모듈로부터 RDF/XML 문서에 대한 파싱 요청을 받아서 SAX 파서를 활성화시킨다. 이 때 SAX 파서를 사용하는 것은 파싱 성능이 가장 좋기 때문이다. 그런 뒤에 유효성 검사 옵션을 "비활성" 상태로 설정하고 문서의 컨텐츠를 처리하기 위한 ContentHandler와 오류 처리를 위한 ErrorHandler를 설정한다. 이와 같이 각각의 처리기를 정하고 파싱을 수행한 뒤에 그 결과를 결과 출력 모듈에 전달을 함으로서 모든 처리를 마치게 된다.

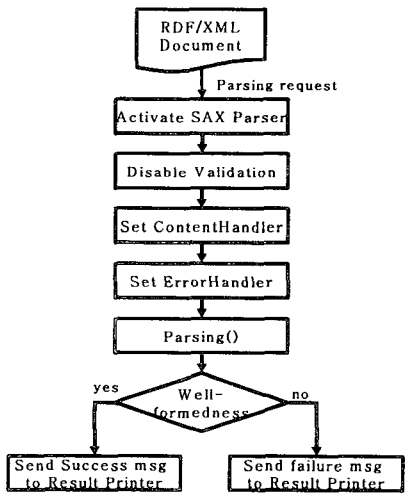

그림 3. 파싱 처리 루틴

4.2 유효성 검증 루틴

이 루틴은 파싱 루틴을 거쳐 DTD와 XML Schema에 근거하여 RDF/XML 문서의 유효성을 검증하는데, 특징적으로 두 번의 유효성 검사를 수행한다. 그림 4는 두 번의 파싱 단계와 그 처리 과정에 대한 것이다.

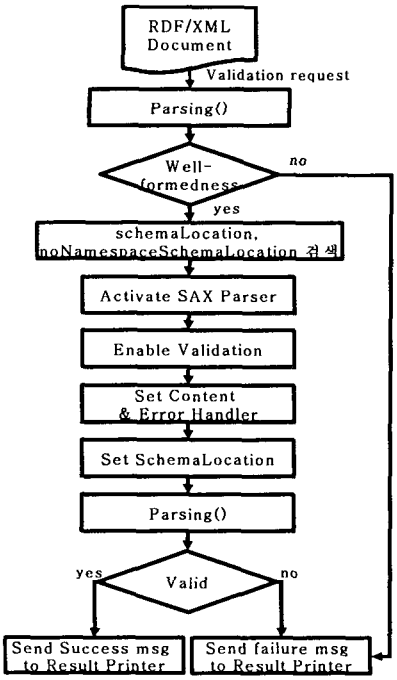

그림 4. 유효성 검증 루틴

처리 루틴에서 첫 번째 파싱 단계는 위의 파싱과 같으나 두 번째 유효성 검증 단계에서는 schemaLocation과 noNamespaceSchemaLocation을 검색한다. 그리고 파싱 과정 중 유효성 검사를 위해 옵션을 활성화하고 파싱에서와 마찬가지로 Content와 Error를 처리하기 위한 각각의 Handler를 설정한다. 그런 뒤에 이전에 검색된 DTD 또는 XML Schema의 위치를 설정한 뒤에 파싱을 수행하고 그 과정 중에 RDF/XML 문서가 유효한지를 검사한다. 이런 루틴을 통해 RDF 기반으로 기술된 웹 자원에 대한 유효성을 보장할 수 있게 된다.

4.3 n3 생성 루틴

n3 생성 루틴은 RDF 문서에 기술되어 있는 자원의 관계(relation)를 명확히 구분할 수 있도록 자원간의 관계를 표현하는데 필요한 모듈이다. 이러한 관계를 분석하기 위해서 ARP 파서를 이용하여 파싱하는데, 문서에서 Element나 Attribute를 찾아 ContentHandler에서 Subject, Predicate, Object따라 각각의 배열에 저장함으로서 n3를 생성한다.

5. 결론 및 향후 연구 과제

본 연구는 RDF 메타데이타 표준을 이용하여 효율적으로 웹 자원을 기술할 수 있는 저작 환경과 기술된 문서의 유효성 검증이 가능하다. 이에 따라 메타데이타 문서의 유효성이 보장되고 이를 이용한 웹 자원의 효율적 관리가 가능하다. 또한 다양한 편집 인터페이스를 지원하여 자원간 관계를 표현할 수 있다는 장점을 지니고 있다. 그러나 저작, 파싱, n3의 생성에서 각각 다른 파서를 사용함에 따라 처리의 지연이 생긴다는 문제를 가지고 있다. 따라서 본 연구의 추후 연구과제로는 처리시간의 지연을 해결하기 위한 통합된 파서를 구현하는 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.