

시맨틱 웹 서비스를 위한 DAML-S 문서 관리 시스템 개발

김학수^o, 손진현
한양대학교 컴퓨터 공학과
{hagsoo^o, jhson}@cse.hanyang.ac.kr

Development of the DAML-S Document Management System enabling Semantic Web Services

Hag Soo Kim^o, Jin Hyun Son

Department of Computer Science and Engineering, Hanyang University

요 약

기존의 WSDL이 갖고 있던 문제점을 개선하고 좀더 의미있는 서비스 기술이 가능한 시맨틱 웹 서비스는, 현재의 웹 서비스와 유사하지만 웹 서비스를 기술하던 WSDL 대신 시맨틱 웹 기술언어를 기반으로 한 새로운 기술 언어(DAML-S)를 사용한다. 이런 상황 속에서 만일 수많은 회사에서 자신들의 서비스를 제공하고, 그 사실을 알리기 위해서 시맨틱 웹 서비스 문서를 발간했을 때, 당연히 사용자는 검색 엔진을 사용하여 그런 비즈니스 정보를 검색하게 되고 사용자가 원하는 서비스, 사용자가 만족할 수 있는 서비스를 찾고 결합시켜주는 검색 엔진이 필요하게 될 것이다. 따라서, 시맨틱 웹 정보 검색은 기존의 정보 검색과는 다른 형태의 검색이 된다. 기존의 문서 내 단어의 존재 유무나 빈도수와 같은 휴리스틱한 방법을 적용하기는 매우 어려우며, 시맨틱 웹 기술 언어에 알맞은 새로운 검색 방법과 저장 시스템이 필요하다.

이 논문에서는 시맨틱 웹 정보 중 특별히 시맨틱 웹 비즈니스 정보에 대한 효율적인 저장 시스템의 모델을 제시하고자 한다. 이 모델을 통해서 시맨틱 웹 비즈니스 정보를 구조화하고 효율적으로 저장할 수 있다.

1. 서 론

HTML을 이용한 웹은 HTML이 가지고 있는 한계점으로 인해 더 이상의 기능적인 성장을 기대하기 어려운 상황이다. 이에 지금의 웹보다 더 많은 기능성과 상호 운용성을 지닌 새로운 웹인 시맨틱 웹이 등장하게 되었다. 시맨틱 웹은 전혀 다른 웹이 아닌 지금의 웹의 확장이다. 그리고 전체를 하나의 통합된 형태로 바라보고 있는 시맨틱 웹은 당연히 지금의 웹이 갖고 있는 많은 요소들을 하나로 통합시키고 있다. 그래서 지금의 웹 상에서 이루어지던 웹 서비스를 시맨틱 웹으로 통합시킨 것이 시맨틱 웹 서비스이다. 이 것이 나오기까지는 많은 기반지식을 요구했다.

시맨틱 웹 서비스가 나오기 까지는 많은 기반지식을 요구하였다. 그러한 기반 지식으로 시맨틱 웹 언어 (RDF, RDFS, DAML+OIL)이 있으며 시맨틱 웹 서비스를 기술하기 위해 DAML-S가 있다. 이러한 것을 기반으로 해서 시맨틱 웹 서비스를 제공할 수 있다. 또한, 시맨틱 웹언어, 시맨틱 웹 서비스를 저장하기 위한 효율적인 저장 시스템도 필요하다. 왜냐하면 문서가 가지고 있는 의미적인 부분들을 손실 없이 저장해야 하며 대용량의 서비스들을 관리해야 되기 때문이다.

이에 이 논문은 시맨틱 웹 비즈니스 정보에 대한 효율적인

저장 시스템의 모델 제시를 통해서 시맨틱 웹 비즈니스 정보를 구조화하고 효율적으로 저장할 수 있게 하는 것이 목적이다.

먼저, 논문의 2장에서는 Jena, Sesame를 보완한 효율적인 저장 시스템의 모델을 제안한다. 3장에서는 2장을 기반으로 해서 실제로 구현한 시스템을 소개한다. 마지막으로 4장에서 이 논문에 대한 결론을 맺고자 한다.

2. DAML-S 문서 저장을 위한 데이터베이스 스키마 설계

2장에서는 DAML-S문서 저장 시스템인 Sesame+BOR, Jena를 비교를 통해서 새로운 데이터베이스 모델을 제시하고 한다. 먼저, 그림 1은 Sesame+BOR와 Jena의 성능 비교 분석 표이다. 표에서는 Sesame 0.92 + BOR를 대상으로 했는데 Sesame가 RDF, RDFS만 저장할 수 있기 때문에 Jena와 비교를 위해 BOR를 사용하였다.

	Jena 1.6.1	Sesame 0.92 + BOR
table	6	20
scalability (statement)	small	3,000,000 이상
performance	small	$O(10^3)$ classes and $O(10^5)$ triples 이상

<그림 1 Sesame와 Jena의 성능 비교>

위의 그림에서 볼 수 있는 것처럼 Jena는 6개의 테이블을 통해서 RDF, RDFS, DAML+OIL를 저장할 수 있다. Scalability

* 본 연구는 과학기술부 21세기 프론티어 사업 인간기능 생활지원 지능로봇 기술개발 사업단의 지원으로 수행되었음

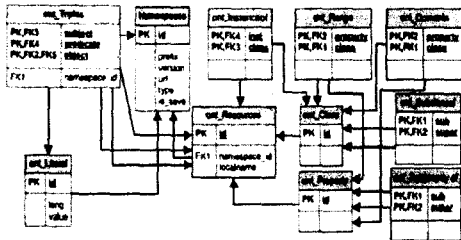
* 본 연구는 한국 과학재단 목적 기초 연구 (R08-2003-000-10464-0)지원으로 수행되었음.

y, performance를 보면 Jena는 대용량의 데이터를 저장하기에는 부적절하다. 왜냐하면, Jena의 질의 시스템 구조 때문이다. Sesame처럼 데이터베이스 시스템을 대상으로 직접 질의하는 것이 아니다. Jena는 A라는 문서를 저장할 때 문서 A가 저장되었다는 것을 그대로 데이터베이스에 유지한다. 그리고 질의할 때는 반드시 하나의 문서만을 대상으로 질의하게 된다. 즉, 질의시 A라는 문서를 저장하고 데이터베이스로부터 A의 문서를 메모리상에 모델화 시킨 다음에 그 모델을 대상으로 질의하는 구조로 되어 있다. 결과적으로 많은 문서가 데이터베이스에 저장될 경우 질의시 성능상의 문제가 발생하는 것은 당연한 결과이다.

이에 반해 Sesame+BOR는 데이터베이스를 대상으로 질의한다. 그래서 많은 문서를 저장할 수 있다. 위의 표에서처럼 Sesame+BOR는 데이터베이스 시스템의 성능에 좌우되는 경향이 있는 것을 볼 수 있다. Sesame+BOR가 목표로 하는 것은 RDF, RDFS, DAML+OIL를 저장하는 데 있어서, 관계형 데이터베이스에 저장할 때 의미 손실을 최대한으로 줄이는 것이다.

다음은 데이터베이스 스키마에 대한 비교이다. 표에서 볼 수 있는 것처럼 Jena가 6개, Sesame+BOR가 20개로 상당히 많이 차이 나는 것을 볼 수 있다. Jena는 데이터베이스에 저장된 레코드를 관리하기가 쉽다. 이에 반해 Sesame+BOR는 상당히 많은 테이블들을 유지관리하기 위해 많은 비용을 들여야 한다.

이에 이 논문에서는 Jena와 Sesame의 장점을 모아 그림 2와 같은 데이터베이스 스키마를 제안하고 자한다. 아래의 데이터베이스 스키마는 Sesame의 데이터베이스 스키마를 수정하고 Jena의 Database API를 사용하여 생성되는 RDF triple을 저장할 수 있도록 테이블 수를 적게 구성하도록 설계하였다.



<그림 2 제안한 데이터베이스 스키마>

위의 그림에서처럼 데이터베이스 스키마는 총 11개의 테이블로 구성이 된다. 본 테이블 중 실제 데이터가 저장되는 것은 ont_Triples, ont_Resource와, ont_Literal, 그리고 Namespaces이다. 나머지 테이블들은 이 세가지 테이블에 저장된 정보의 ID를 저장한다. 각각의 테이블에 대한 설명은 아래와 같다.

- ① Namespace : 이름 공간에 대한 정보를 저장하는 테이블
- ② ont_Resource : 온톨로지에 포함된 모든 자원들을 저장하는 테이블
- ③ ont_Literal : 온톨로지에 포함된 모든 Literal들을 저장하는 테이블
- ④ ont_Triple : 온톨로지에 포함된 모든 Triple들을 저장하는 테이블
- ⑤ ont_Range : 자원들의 Range 정보를 저장

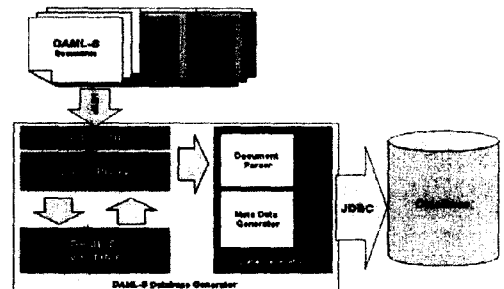
- ⑥ ont_Domain : 자원들의 Domain 정보를 저장
- ⑦ ont_Subclassof : 클래스들의 상호 상속 관계 정보를 저장
- ⑧ ont_Subproperty : 속성들의 상호 상속 관계 정보를 저장
- ⑨ ont_Class : 자원들 중 클래스로 정의된 것들을 추출하여 ID를 저장
- ⑩ ont_Property : 자원들 중 속성으로 정의된 것들을 추출하여 ID를 저장
- ⑪ ont_Instanceof : 각 자원들의 클래스-인스턴스 관계 정보를 저장

이처럼 제안한 데이터베이스 스키마는 11개의 최적화된 테이블을 가지고 있다. 이 11개의 테이블을 통해서 RDF(S), DAML+OIL, DAML-S 문서를 관계 데이터베이스에 저장할 때 의미의 손실을 최대한으로 줄이도록 설계되었다.

데이터베이스 스키마에서 핵심을 이루는 테이블은 ont_Triples, ont_Resource, ont_Literal, Namespaces부분이다. RDF(S), DAML+OIL, DAML-S 문서들은 RDF의 Triple형식을 가지고 있기 때문에, 이 것을 반영한 설계이다. 그 중에서 핵심을 이루는 테이블은 ont_Resource이다. 시맨틱 웹 문서 자체가 웹에 있는 자원의 기술이기 때문이며, ont_Resource는 그러한 모든 자원을 저장하는 가장 중요한 테이블이다. 그리고 이 리소스 테이블을 기반으로 해서 Triple 형태로 ont_Triples에 저장하고, 리소스와 관련된 namespace 부분을 Namespaces 테이블에 저장한다. 또한 데이터베이스에 저장할 때 약간의 추론을 통해서 저장하는데 그것은 의미의 손실을 최대한으로 줄이기 위한 목적이다. 거기에 해당 되는 것이 ont_instanceOf, ont_Class, ont_Property, ont_Range, ont_Domain, ont_Subclassof, ont_Subpropertyof 이다.

실제로 시맨틱 웹 검색에서의 질의시에 대부분 추론엔진이 문서의 의미적인 부분을 추론하게 된다. 그러나 저장된 문서가 가지고 있는 의미의 손실을 줄여야지만 추론엔진에서 효율적으로 추론할 수 있다. 제안한 데이터베이스 스키마는 RDF triple, Resource를 중심으로 하여, 문서가 내포하고 있는 의미를 subClassOf, oneOf, subPropertyOf 등을 손실 없이 저장할 수 있게 한다. 그래서 4장에서는 이 데이터베이스 스키마를 가지고 실제 시스템 구현에 대해서 설명한다.

3. 시맨틱 웹서비스를 위한 시맨틱 웹 서비스 문서 저장 시스템 설계 및 개발



<그림 3 Database Generator 아키텍처>

그림 3은 데이터베이스 Generator에 대한 아키텍처이다.

데이터베이스 Generator는 DAML-S 문서를 분석하여 다른 모듈(Query 모듈, 사용자 인터페이스 등)에서 사용할 수 있는 적절한 데이터를 생성하여 데이터베이스에 저장하는 작업을 수행하는 모듈이다. 전체적으로 3개의 모듈이 데이터베이스 생성 작업을 수행한다. 기본적으로 문서 Validator와 문서 파서는 Jena 1.6.1을 사용한다. 왜냐하면, 개발자 입장에서 Jena 1.6.1은 효율적인 문서 Validator와 문서 파서를 제공하기 때문이다. 그리고 Jena를 사용해서 RDF triple를 추출하게 된다. 각각의 모듈에 대한 설명은 아래와 같다.

① Client Proxy

관리자 인터페이스로부터 데이터베이스에 추가 하고자 하는 문서 정보를 입력받아 문서 객체를 생성하고 전체 시스템을 가동시킨다. 한 문서에 대한 모든 처리가 종료되면 관리자 인터페이스로 처리 결과를 출력하여 관리자로 하여금 결과를 확인할 수 있도록 한다. 문서의 주소는 인터넷에 존재하는 DAML-S 문서의 URL을 입력 받는다.

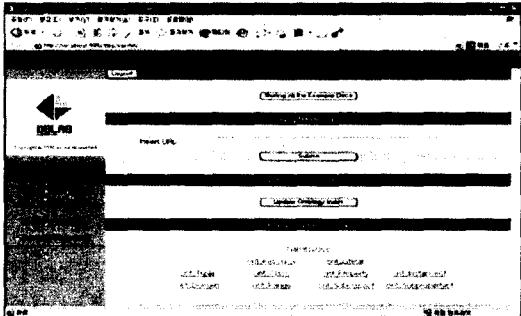
② DAML-S Validator

Client Proxy에서 받아들이는 문서의 유효성을 검사한다. 기존의 DAML Validator를 기반으로 DAML-S를 처리 할 수 있도록 한다. 실제로는 Jena 내부의 Validator를 사용하였다. Jena의 Validator는 DAML 문서에 대한 간단한 수준의 유효성 검사를 수행한다.

③ Data Generator

아키텍처의 대략적인 설명은 다음과 같다. DAML-S Validator로부터 정상적인 문서임을 검증 받으면, 이 문서로부터 적절한 정보를 추출하고, 그 정보를 데이터베이스에 저장한다. 실제 데이터베이스와의 통신은 이 모듈 내부의 DB Connector가 수행한다.

위와 같이 설계된 Database Generator 모듈은 질의 프로세서나 사용자 인터페이스와 쉽게 결합하여 시맨틱 비즈니스 정보 검색 엔진구현에 이용할 수 있다. 실제로 구현된 시스템 환경은 Jakarta 'Tomcat' HTTP Server기반이며 데이터베이스 시스템으로 Microsoft SQL Server 2000을 사용하였다. Database Generator에 추가적으로 웹 기반의 유저 인터페이스를 구현하였고 질의를 하기 위해 Semantic Query Processor와 Semantic Query Supporter라는 모듈을 추가하여 구현하였다.



<그림 4 데이터베이스 Generator와 관리 모듈 구현>

그 중에서 데이터베이스 Generator와 관리 모듈을 그림 4에서 보여주고 있다.

데이터베이스 관리 모듈은 DAML-S 문서를 데이터베이스에 입력하고, 입력한 데이터에 이상이 없는지 확인할 수 있는 관리자 페이지이다. 데이터 입력과 같이 중요한 부분에 있어서는 여느 웹사이트들과 마찬가지로 계정 로그인을 통해 접근하도록 하였으며, 데이터 확인 페이지는 일반 사용자들도 로그인을 거치지 않고도 접근할 수 있게 구성하였다.

관리자 페이지를 기능적으로 분류하면 데이터베이스 입력 부분과 데이터 확인 부분으로 나뉘어 지는데, 위 그림에서 'Storing DAML-S Docs' 메뉴가 입력부분에 해당하며, 'Browsing Ontologies'와 'Browsing Repository' 메뉴가 데이터 확인 부분에 해당한다.

4. 결론

시맨틱 웹에 대한 연구가 진행되면서 시맨틱웹의 활용 분야가 점차 확대되고 있다. 이에 따라 효율적으로 시맨틱 웹 언어의 문서를 저장하는 시스템의 중요성도 커지고 있는데, 이러한 문서는 기존의 텍스트 문서처럼 단순히 저장되는 것이 아니라 문서가 가지고 있는 의미적인 부분들 (subjectOf, range, domain 등의)을 손실없이 저장하는 것이 중요하다. 위에서 소개한 데이터베이스 스키마는 이러한 손실을 최대한 줄이기 위한 목적으로 설계되었다. 그러나 그러한 손실을 줄이는 것도 한계가 있으며 관계데이터베이스 기반이 아닌 객체기반 데이터베이스나, XML 데이터베이스 기반의 저장 시스템이 좀더 적합하다고 볼 수 있겠다. 앞으로 이러한 기반의 저장 시스템이 필요하게 될 것이다.

References

- [1] Alexander Maedche, Steffen Staab, *Ontology Learning for the Semantic Web*, 2001
- [2] BOR, <http://www.ontotext.com/bor/index.html>
- [3] DAML, <http://www.daml.org>
- [4] DAML+OIL, <http://www.daml.org/2001/03/daml+oil/index.html>
- [5] DAML-S 0.9 Draft Release. <http://www.daml.org/services/daml-s/0.9>
- [6] Extensible Markup Language, <http://www.w3.org/XML>
- [7] James Hendler, Sheila A. McIlraith, David L. Martin, *Bringing Semantics to Web Services*, 2003
- [8] Jeen Broekstra, Arjohm Kampman, *Sesame : A Generic Architecture for storing and Querying RDF and RDF Schema*, 2001
- [9] Resource Description Framework, <http://www.w3.org/RDF>
- [10] RDF Vocabulary Description Language 2.0 : RDF Schema, <http://www.w3.org/TR/rdf-schema>
- [11] Semantic Web, <http://www.w3.org/2001/sw>
- [12] Sheila, A. McIlraith et al, *Semantic Web Services, IEE Intelligent Systems*, 2001
- [13] T. Berners-Lee, J. Hendler, and O. Lassila, *The Semantic Web*, *Scientific American*, vol 284, no.5, p.34-43, 2001