

온톨로지 저장소 아키텍처

이동훈⁰, 양정진

가톨릭대학교

{rephaser⁰, jungjin}@catholic.ac.kr

An Architecture for Ontology Repository

DongHun Lee⁰, Jungjin Yang

School of Computer Science and Information Engineering

The Catholic University of Korea

요약

차세대 웹으로 기대되는 시맨틱웹에 대한 기술에 대한 관심이 증대되고 있어 시맨틱웹에서 가장 중요시 되는 온톨로지의 대한 관리에 대한 연구가 활발히 진행중이다. 온톨로지 관리에 대한 연구로는 KAON, OntoServer등이 진행되고 있으며 이러한 연구에서는 RDF, DAML등의 온톨로지 언어를 지원한다. 이에 본 연구에서는 W3C에서 표준 온톨로지 언어로 지정한 OWL을 지원하는 온톨로지 저장소에 대한 아키텍처를 제시하고자 한다.

1. 서 론

시맨틱 웹은 컴퓨터가 웹상에 존재하는 자원들을 인지하여 논리적인 추론을 통해 자원들 간의 관계를 파악하고, 사용자가 필요로 하는 정보에 대한 정확한 결과를 찾아주는 차세대 지능형 웹을 의미한다.

시맨틱 웹에서 가장 중요시 되는 기술이 온톨로지 기술로 온톨로지란 웹상에 존재하는 자원들을 정의하고 자원들 사이의 관계를 나타내는 기술이다. 이 온톨로지 정보들은 컴퓨터가 이해할 수 있는 형태로 표현되는데 이러한 표현을 위해 사용되는 것이 온톨로지 언어다.

대표적인 온톨로지 언어로는 RDF(Resource Description Framework), DAML(DARPA agent markup language), DAML+OIL, OWL(Web Ontology Language)[1]등이 있다.

RDF는 주어(Subject), 서술어(Property), 목적어(Object)로 온톨로지를 표현하며, RDFS(RDF Schema)는 RDF의 타입 및 관계를 나타내는 언어다. DAML은 미국 DARPA에서 개발된 마크업 언어로 객체 및 객체들 간의 관계를 묘사를 통해 온톨로지를 표현하고, 웹사이트들 간에 보다 높은 차원의 상호 운용성을 구축하도록 설계되었다. DAML+OIL은 DAML과 유럽에서 개발한 OIL을 결합한 온톨로지다. OWL은 RDF와 DAML+OIL에서 확장된 언어로서 보다 풍부한 Vocabulary를 가지고 있어, 자원들의 개념 및 관계를 다양하고 세밀하게 기술하는 것이 가능하며, 추론을 지원한다. 또한 W3C가 OWL을 표준 온톨로지 언어로 지정하여 앞으로의 온톨로지 데이터들은 OWL로 가술되어질 전망이다.

시맨틱 웹의 실현을 위해서는 OWL로 기술된 온톨로지 정보를 효율적으로 저장하고 관리하는 기술이 뒷받침되어야 한다. 기존에 연구된 온톨로지 저장소로는 KAON[2], OntoServer[3]등이 있으나, 이를 온톨로지

관리 시스템은 대용량의 온톨로지 데이터에 대한 저장, 질의등을 지원하지 못하거나 표준 온톨로지 언어인 OWL이 아닌 RDF, DAML+OIL등으로 기술된 온톨로지 데이터를 대상으로 설계 되었다.

따라서 온톨로지 표준언어인 OWL로 기술된 데이터를 효율적으로 관리하는 시스템의 필요성이 대두 되었고, 본 연구에서는 OWL을 지원하는 온톨로지 관리 시스템에 대한 모형을 제시하고자 한다.

2. 관련연구

2.1 KAON

KAON은 온톨로지 기반의 어플리케이션을 개발하기 위한 기본적인 프레임워크로 여러 모듈들을 제공한다.

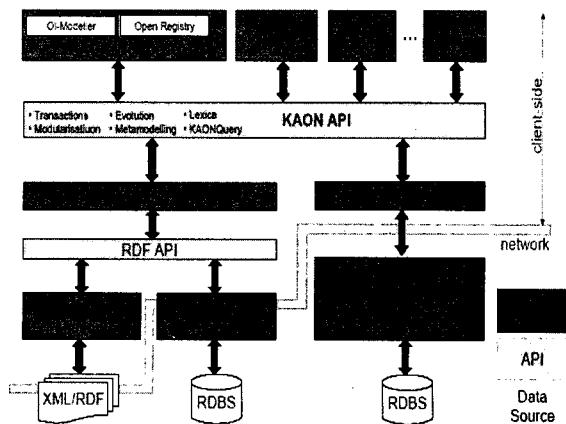


그림 1. KAON 아키텍처

그림 1은 KAON의 아키텍처를 보여주고 있다. KAON의 각각의 모듈에 대해서 보면 엔지니어링 서버는 관계형 데이터베이스(RDB)에 기반한 카온 온톨로지 를 위한 저장 메커니즘, 온톨로지 엔지니어링을 위한 것들, 응집성 탐지를 통한 클라이언트 사이드 캐시, 온톨로지 엘리먼트의 벌크로딩과 분산 변화 탐지 메커니즘 등을 가지고 있고 KAON API는 온톨로지와 온토롤러 저장 독립적인 인스턴스에 대한 프로그램을 가능하게 하는 문서. 온톨로지 전화, 온톨로지 모듈화, 동시적인 온톨로지 접근과 트랜잭션 프로세싱을 지원함. 저장 메커니즘은 RDF 모델과 관계형 데이터베이스를 포함한다. KAON Query는 질의 언어를 통하여 카온 온톨로지에 대한 질의를 가능하게 하는 역할을 한다.

KAON REVERSE는 관계형 데이터베이스를 온톨로지에 매핑하여 RDF 인스턴스를 생성하는 작업을 한다.

이 외에도 시맨틱 웹을 위한 어플리케이션 서버인 KAON SERVER, KAON API 구현에 직접 동작하는 OntoEdit를 위한 플러그인인 KAONtoEdit등등 온톨로지를 관리하기 위한 많은 모듈들을 제공한다.

2.2 DLDB[4]

DAML+OIL에 관한 지식기반 시스템으로 관계형 데이터베이스 관리 시스템을 기반으로 하여 추론을 가능하게 하는 시스템이다. DAML+OIL의 기반이 되어지는 RDF는 주어, 서술어, 목적어의 세 요소로 구성되며, DLDB는 이러한 세 요소를 기반으로 저장한다.

FACT라는 추론기에 의해 추론된 클래스 및 프로퍼티의 계층 정보도 함께 저장되어 진다.

표 1은 기존의 온톨로지 저장소들을 비교한 표이다.

	KAON	DLDB	OntoServer
Storage	XML repository / RDBS	XML repository / RDBS	Client/Server Based
Versioning	Versioning	no Versionning	no Versionning
Searching	Graphical Display	no Searching	no Searching
Editing	Graphical Editor	no editor	no editor
Reasoning	Description Logic Program	Description Logic	no reasoning
Language	RDF	DAML+OIL	RDF

표 1. 온톨로지 저장소 비교

기존에 연구된 온톨로지 저장소들은 대부분 초기의 온톨로지 언어인 RDF나 OWL 이전의 언어인 DAML+OIL을 사용하고 있으며, KAON을 제외한 온톨로지 저장소

들은 온톨로지를 관리하는 기능들이 부족한 것을 볼 수 있다. KAON은 온톨로지 관리 기능들도 충실히 구현되어 있으나 대규모 온톨로지 관리에는 부적합하다.

3. 온톨로지 저장소 아키텍처

온톨로지 저장소가 온톨로지를 관리하기 위해 기본적으로 요구되는 사항은 아래와 같다.

- Legacy Database의 접근을 위한 Interface
- 분석적인 질의 지원
- 임시적인 저장공간과 질의문
- 일관성 Checking
- 신뢰성 Management
- Legacy Data와 지식을 통합하기 위한 기술
- 다수의 Mediator와 Knowledge Server를 연결하기 위한 Bridge

그림 1은 위의 요구사항들을 고려하고, 기존의 온톨로지 저장소들이 가지고 있는 단점을 보완하여 온톨로지 저장, 통합, 편집, 검색, 버전 처리 등의 파트를 가진 온톨로지 저장소 아키텍처 모델을 보여주고 있다.

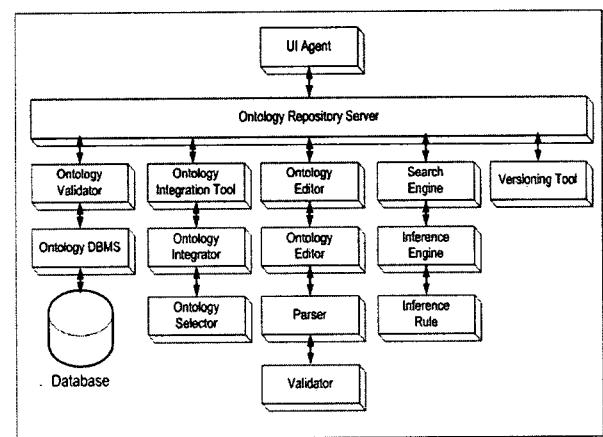


그림 2. 온톨로지 저장소 아키텍처

각 컴포넌트에서 수행하게 될 역할 및 특징은 아래와 같다.

- UI Agent : 사용자나 다른 시스템이 온톨로지 저장소에 접근하기 위한 인터페이스로서 시스템간의 협업에 필요한 작업 등을 처리한다.
- Ontology Repository Server : UI Agent로부터 들어온 작업을 각 컴포넌트에 연결하는 역할을 한다.
- Ontology Validator : 입력된 온톨로지가 유효한지 판단하는 역할을 한다.
- Ontology DBMS : 입력된 OWL 형태의 온톨로지에서 클래스, 프로퍼티의 계층 정보를 추출하여 XML 데이터베이스에 저장하고, 클래스 인스턴스에 대한

정보는 주어, 서술어, 목적어의 트리플로 재구성하여 클래스 프로퍼티들의 정의에 대한 정보와 함께 관계형 데이터베이스에 저장한다.

- Database : 관계형 데이터베이스와 XML 데이터베이스로 나누어지며 Ontology DBMS에서 전달된 정보를 저장한다.
- Ontology Integration Tool : 온톨로지 통합 방법으로는 자동 및 수동 통합 방법을 모두 제공하며, 자동적인 방법은 기계학습 등의 방법을 제공한다. 수동적인 방법은 온톨로지 편집기와 같은 편집기로 사용자가 직접 통합할 수 있는 방법을 제공한다.
- Ontology Integrator : 온톨로지를 기계학습 등의 방법을 통해 자동적으로 통합한다.
- Ontology Selector : 통합되어질 온톨로지를 선택하는 역할을 한다.
- Ontology Editor : 온톨로지의 각 구성 요소의 편집 등을 제공하며, 편집하고자 하는 온톨로지의 버전을 선택한다.
- Document Editor : OWL 문서 형태의 온톨로지를 편집한다.
- Parser : 작성된 OWL 문서의 문법을 검사하는 역할을 한다.
- Validator : 작성된 온톨로지가 유효한 온톨로지인지 검사하는 역할을 한다.
- Search Engine : 저장된 온톨로지를 사용자의 요구에 따라 검색한다.
- Inference Engine : 온톨로지로부터 추론 규칙을 적용하여 추론의 기능을 수행하며, 질의어의 연관 관계를 파악하여 관련 용어를 검색 엔진에 제공한다.
- Infernece Rule : 추론에 필요한 규칙들을 가지고 필요한 규칙을 검색엔진에 제공한다.
- Versioning Tool : 온톨로지의 버전에 따른 처리를 담당한다.

4. 온톨로지 저장소 수행과정

온톨로지 저장소에서 온톨로지가 저장되는 과정을 보면 다음과 같이 수행되어진다. UI Agent로부터 온톨로지의 생성 요청이 들어오면, Ontology Repository Server는 생성 요청이 들어온 온톨로지가 유효한 온톨로지인지 판별하기 위해 Ontology Validator에 입력된 온톨로지 문서를 전달한다. Validator은 전달 받은 온톨로지 문서의 유효성 체크를 하고 유효한 문서라면 저장을 위해 Ontology DBMS에 온톨로지 문서를 전달한다. 온톨로지 DBMS는 전달 받은 온톨로지 문서에서 클래스, 프로퍼티의 계층 정보를 추출하여 XML 데이터베이스에 저장하고, 클래스 인스턴스에 대한 정보는 주어, 서술어, 목적어의 트리플로 재구성하여 클래스 프로퍼티들의 정의에 대한 정보와 함께 관계형 데이터베이스에 저장한다.

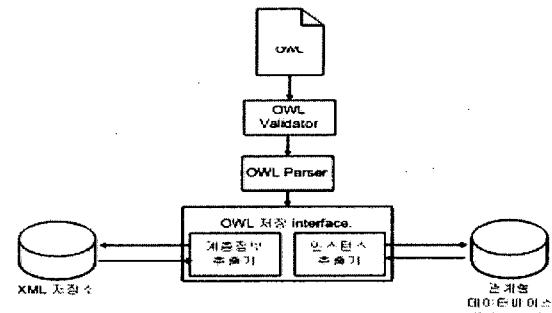


그림 3. 온톨로지 저장 수행 절차

5. 결과 및 고찰

본 연구에서는 기존에 연구된 온톨로지 저장소들의 단점을 보안하며, W3C가 지정한 표준 온톨로지 언어인 OWL에 기반 한 온톨로지 저장소 아키텍처 설계 방법을 연구하였다. 향후 연구로는 제안한 온톨로지 저장소를 효과적이고 효율적으로 구축하기 위한 전략적 방안과 OWL 형식의 온톨로지 뿐만 아니라 웹상에 존재하는 RDF, DAML, DAML+OIL등의 다른 형식의 온톨로지들을 지원하는 온톨로지 저장소에 대한 연구가 진행될 예정에 있다.

참조문헌

- [1] W3C, OWL Web Ontology Language Guide, <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>
- [2] KAON - An Overview Karlsruhe Ontology Management Infrastructure, <http://kaon.semanticweb.org/documentation>
- [3] Vassiliadis, V. Rdftp: RDF Server. <http://semanticweb.gr/rdftp>
- [4] Z. Pan and J. Heflin. "DLDB: Extending Relational Databases to Support Semantic Web Queries." In Workshop on Practical and Scalable Semantic Web Systems, pp. 109-113, 2003.
- [4] Wonder Web - Ontology Infrastructure for the Semantic Web, <http://wonderweb.semanticweb.org>.
- [5] Jin Pan, Stephen Cranefield and Daniel Carter, "A Lightweight Ontology Repository", ACM 1-58113-683-8/03/0007
- [6] R. Fikes, P. Hayes, and I. Horrocks. "OWL-QL - A Language for Deductive Query Answering on the Semantic Web." Knowledge Systems Laboratory Technical Report 03-14, Stanford University, 2003.
- [7] I. Horrocks. The FaCT System. In Proc. of Automated Reasoning with Analytic Tableaux and Related Methods: International Conference, pp. 307-312, 1998.