

설계 지능화를 위한 통합 시스템 구조

Integration system architecture for intelligence design

*한지웅¹, #양정진²,

*J.W.Han¹, #J.J.Yang(jungjin@catholic.ac.kr)²

¹가톨릭 대학교 컴퓨터 정보 공학부, ²가톨릭 대학교 컴퓨터 정보 공학부

Key words : Intelligence design, Ontology, Description Logic, Datalog, System Architecture

개요

설계 시스템들의 자동화와 더불어 설계 지식의 재사용성의 요구가 증대되면서 설계 자동화 시스템은 주어진 문제에 따라 독립적으로 발전하고 있기 때문에 설계 지식의 재사용을 위해선 설계 데이터에 대한 저장, 추론, 질의를 할 수 있는 통합 시스템이 필요하다. 본 논문에서는 선행 연구에서의 온톨로지 저장소에서 설계 지식의 저장, 회수 방법에 이어, 저장된 설계 지식을 사용하기 위하여 적재와 추론과 질의를 통합적으로 사용할 수 있는 시스템을 제시한다.

1. 서론

설계 시스템들의 자동화와 더불어 설계 지식의 재사용성의 요구가 증대되고 있다. 설계 자동화 시스템은 주어진 문제에 따라 독립적으로 발전하고 있기 때문에 설계 지식의 재사용을 위해선 설계 데이터에 대한 저장, 추론, 질의를 할 수 있는 통합 시스템이 필요하다. 선행 연구에서는[1] 온톨로지 저장소를 이용하여 설계 지식의 저장과 회수를 위한 방법을 제안하였으며, 본 논문에서는 선행 연구에서 연구된 설계 지식의 저장과 회수와 함께 이러한 설계 지식을 추론하고, 질의 할 수 있는 통합적인 구조를 제안한다.

2. 관련연구

선행 연구에서 온톨로지로 작성된 설계 지식을 온톨로지 저장소에 저장하기 위해 Minerva[2] 온톨로지 저장소와 제공된 API를 사용하여 저장한다. 온톨로지를 저장하기 위한 DBMS로 IBM DB2를 사용하고 표현된 설계 지식을 관계형 데이터베이스에 저장하기 위해서 Minerva 온톨로지 저장소에서 제공하는 DB Schema와 적재 모듈을 사용하였다.

또한 온톨로지 저장소로부터 설계 지식을 회수하기 위해 의미적 질의를 수행한다. 의미적 질의 언어는 질의문 자체에 필요로 하는 데이터의 의미를 부여해 질의 할 수 있는 언어이며 RDF 트리플 형태로 의미를 부여하는 SparQL[3]을 사용하였다.

온톨로지 저장소는 지역적으로 온톨로지를 구성하는 데이터의 생성, 저장, 회수, 편집, 검색, 통합, 버전관리, 일관성유지를 지원하여야 하고 추론을 위한 메타정보를 효율적으로 저장/회수할 수 있어야 하며 대외적으로 온톨로지들 간 병합, 합성을 지원할 수 있어야 한다. 대부분의 온톨로지 저장소는 온톨로지를 기술하는 표준 언어인 OWL을 OWL/RDF로 파싱하여 RDBMS에 저장한다. 파싱된 결과를 데이터베이스에 저장/회수하기 위한 다양한 메커니즘들이 시도 되고 있으며 Ontology Repository에 저장된 데이터를 핸들링 하기 위한 API, Query Engine 인터페이스들을 외부 개체에게 제공한다. 이러한 온톨로지 저장소로 HAWK [4], KAON[5] 등이 있고 본 논문의 통합 시스템에서 사용할 Protege[6], Minerva 가 존재한다.

Minerva는 관계형 데이터베이스 시스템을 기반으로 OWL을 저장/회수, 조작하며 SPARQL Protocol And RDF Query Language (SPARQL)로 의미 기반의 질의와 응답을 수행하는 온톨로지 저장소이다. Minerva는 TBox 추론을 위하여 DL 추론 엔진을 사용하고, ABox 추론을 위하여 Description Logic Programming (DLP)을 사용한다. Minerva는 온톨로지가 저장되는 Storage 모듈, 온톨로

지를 불러오는 Import 컴포넌트로 back-end를 구성하며 추론엔진 인터페이스와 질의 엔진 인터페이스로 Front-end를 구성한다.

3. 설계 지능화를 위한 통합 시스템 구조

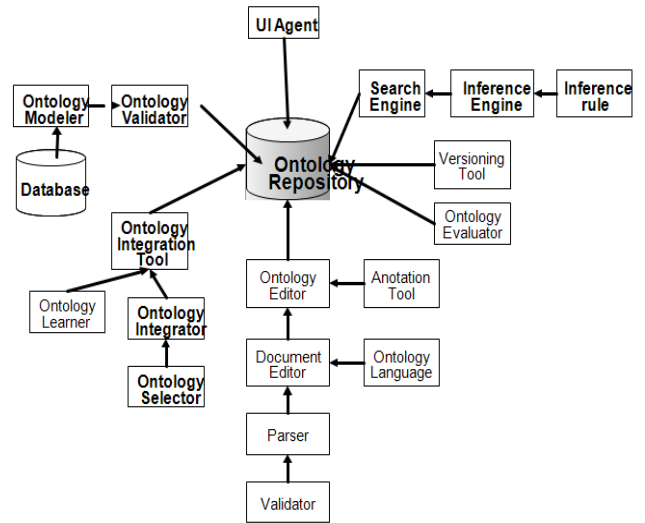


Figure 1. 통합 시스템의 개념도

Figure 1.의 온톨로지 저장소 아키텍처는 대용량 온톨로지를 효율적으로 관리하기 위해 설계 되었다. Ontology Repository는 크게 UI Agent, Ontology Database, Search Engine, Ontology Editor, Ontology Integration Tool로 나눌 수 있고, 각각의 역할로써 UI Agent는 사용자나 다른 시스템이 온톨로지 접근하기 위한 인터페이스의 역할을 담당하고, Ontology Database는 온톨로지 데이터를 저장,관리하며 Search engine은 추론엔진과 규칙을 이용하여 저장된 온톨로지로부터 필요로 하는 정보를 추출하는데 사용된다. Ontology Integration Tool은 온톨로지를 통합하는 도구로서 다른 온톨로지들을 통합하여 필요로 하는 새로운 온톨로지를 만드는데 사용되고 Ontology Editor는 온톨로지의 생성, 편집 등의 기능을 담당한다. Ontology Validator는 입력된 온톨로지가 유효한지 판단하며, 부가적으로 온톨로지의 버전에 따른 처리를 위해 Versioning Tool 등이 있다.

Ontology의 표현 및 모델링 기법으로 지식 저작 도구와 응용 계층 간 연결을 해야 하며, 이를 위하여 Figure 2.에서 보듯이 eclipse 기반의 응용 계층으로 Minerva를 이용하여 Ontology의 적재 및 업데이트를 하게 하며 Protege 3.1.1을 이용하여 Ontology를 Modeling하고 이 두 계층을 연계 할 수 있도록 한다.

Load Process 에서는 온톨로지화된 설계 지식을 파싱하고 파싱된 설계 지식을 지식 베이스에 적재 할 수 있도록 한다.

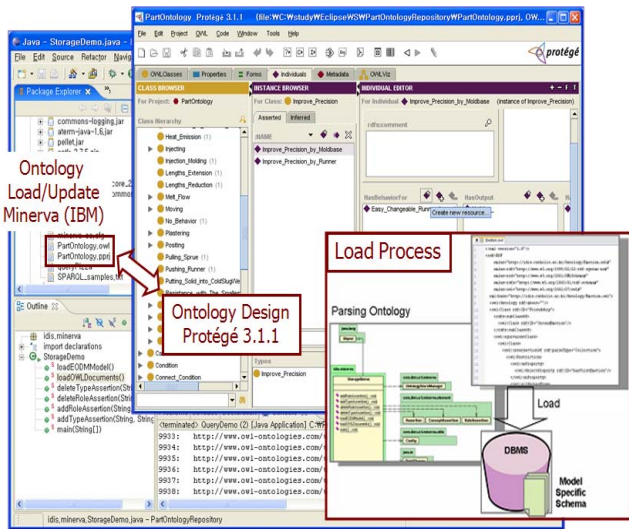


Figure 2. Ontology Design and Load/Update

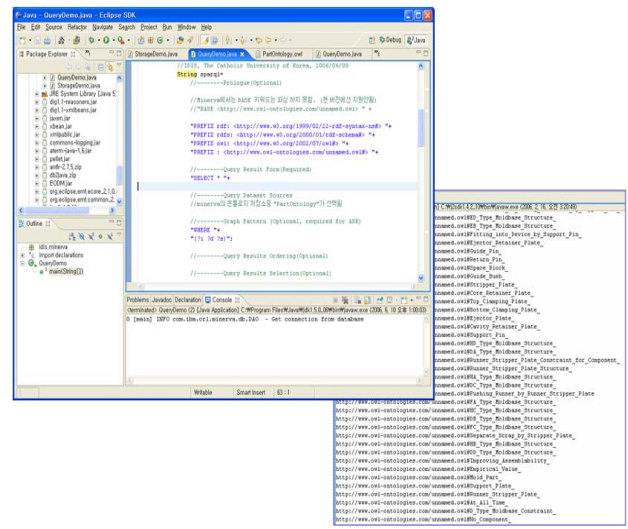


Figure 4. SPARQL을 이용한 Query & Result

Ontology를 RDF로 파싱하여 DB에 저장하고 기술된 Logic으로 Ontology를 확장하여야 한다. Completeness와 Soundness를 만족하는 Logic과 Reasonor를 바탕으로 추론된 지식들을 회수하여 저장소의 지식을 확장시킨다. Figure 3.는 Ontology Integration과 관련하여 Racer 엔진을 이용하여 RDF 데이터와 서술 논리 연산을 함께 Racer를 이용하여 추론하고 다시 DBMS에 추론된 RDF 데이터를 저장한다.

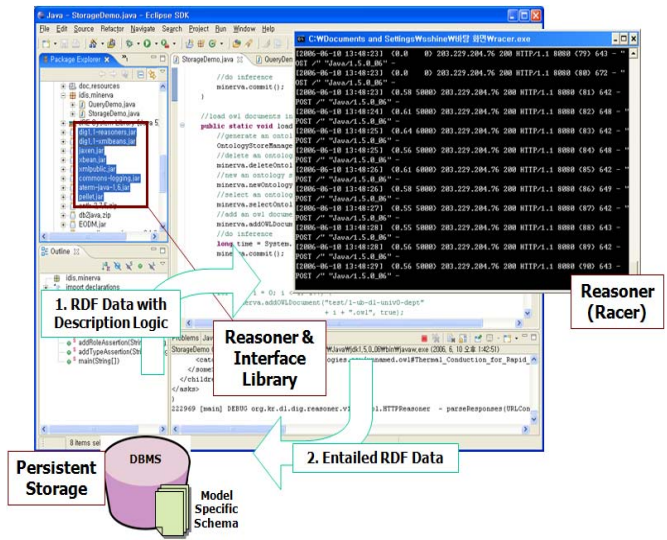


Figure 3. Racer를 이용한 Resoning

Figure 4. 는 저장된 설계 지식을 사용하기 위해 SparQL을 이용해 Query를 보내고 이에 대한 결과를 받는 부분이다. Minerva에 있는 SparQL의 질의 엔진을 이용하여 DB에 대한 질의를 수행한다.

위에서 보여진 Protege에서의 온톨로지의 디자인과 minerva에서의 Ontology 적재 및 Updateae, Racer를 이용한 추론 및 SparQL을 이용한 질의 과정은 Eclipse 플랫폼을 통해 통합된 시스템으로 구축할 수 있다.

4. 결론

본 논문에서는 선행 연구였던 설계 지식의 저장과 회수 방법이 이어, 저장된 설계 지식을 이용하기 위한 통합 시스템의 구축을 하였다. 여기서 추론기였던 Racer의 경우 Tableau 알고리즘 기반의 접근 방식으로 TBox 추론에 효과적이거나 ABox 추론에서는 실용성이 떨어지며, 이런 단점을 극복하기 위해 선언적 데이터 그와 같은 접근 방식이 존재하는데 대표적인 추론기로 KAON2[7]가 존재한다.

현재의 시스템을 통하여 온톨로지의 재사용성을 높이고 설계 지식의 Know-How를 생성할 수 있는 기반 구조를 가지게 되며, 추후 Racer의 단점을 해결할 수 있는 Kaon2를 이용한 시스템 설계가 가능 할 것으로 보인다.

후기

본 연구는 산업자이부 핵심연구개발사업 “미세 형상 설계 기능화 기술 개발” 과제 수행의 일환으로 이루어진 것임을 밝히며, 지원 기관에 감사드립니다.

참고문헌

1. 지경환, 양정진, “온톨로지 저장소를 이용한 설계 지식의 저장과 회수, 한국정밀공학회 2006 춘계 학술대회
2. F. Zambonelli, N. Jennings, and M. Wooldridge, Developing Multiagent Systems: The Gaia Methodology, ACM Transactions on Software Engineering Methodology, 12(3), pp.317-370, 2003.
3. <http://www.w3.org/TR/rdf-sparql-query/>
4. M. Kim et al., Agent-Oriented Software Modeling, In Proceeding of Sixth Asia Pacific Software Engineering Conference, pp.318-325, 1999.
5. M. Wooldridge, N. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design, International Journal of Autonomous Agents and Multi-agent System, 3(3), pp.285-312, 2000.
6. <http://protege.stanford.edu/overview/protege-owl.html>
7. <http://kaon2.semanticweb.org>