

온톨로지 저장소 관리에 관한 연구

이동훈*(가톨릭대 컴퓨터공학과), 양정진(가톨릭대 컴퓨터공학과)

The Research for Ontology Repository Management

D. H. Lee(Computer. Science. Dept. Catholic Univ), J. J. Yang(Computer. Science. Dept. Catholic Univ)

ABSTRACT

The increased use of ontologies for knowledge sharing emerges in many applications where knowledge applicability plays a critical role. The trend demands the need for an infrastructure that allows management tools to use ontology more easily such as ontology editors, storing, integration and inference engines towards comprehensive ontology-based solutions. We call such an infrastructure as ontology repository. This paper designed ontology repository for scalable ontology data

Key Words : Ontology, Repository, OWL

1. 서론

금형 설계의 지능화 및 자동화는 많은 연구가 있어왔지만 다양한 형태의 설계지식 활용이나 대용량 데이터를 이용한 설계 자동화는 아직 큰 성과를 거두지 못하고 있는 실정이다. 이에 본 논문에서는 온톨로지를 활용하여 금형 설계 자동화를 실현하고자 한다.

온톨로지란 단어와 관계들로 구성되고, 특정도메인에 관련된 단어들을 계층적 구조로 표현하고 이를 확장할 수 있는 추론 규칙을 포함 하고 있다.

온톨로지는 어휘 사이의 형식보다는 내용 중심으로 만들어지고 정보의 처리방식이 단순한 패턴 매칭이 아닌 내용 중심적이기 때문에 정보의 통합과 공유에 유리하며 사용자에게 맞게 특화되거나 전문 분야에 적합하게 세분화가 가능하며 경우에 따라서는 상충되는 온톨로지의 집합체도 허용한다. 또한 추론기능을 제공하고 있다.

온톨로지 언어로는 대표적으로 RDF(Resource Description Framework), DAML(DARPA Agent Markup Language), DAML+OWL, OWL(Ontology Web Language)[1]등이 있다.

RDF 는 주어(Subject), 서술어(Property), 목적어(Object)로 온톨로지를 표현하며, RDFS(RDF Schema)는 RDF 의 타입 및 관계를 나타내는 언어다. DAML 은 미국 DARPA 에서 개발된 마크업 언

어로 객체 및 객체들간의 관계를 묘사를 통해 온톨로지를 표현하고, 웹사이트 들 간에 보다 높은 차원의 상호 운용성을 구축하도록 설계되었다.

DAML+OIL 은 DAML 과 유럽에서 개발한 OIL 을 결합한 온톨로지다. OWL 은 RDF 와 DAML+OIL 에서 확장된 언어로서 보다 풍부한 Vocabulary 를 가지고 있어, 자원들의 개념 및 관계를 다양하고 세밀하게 기술하는 것이 가능하며, 추론을 지원한다. 또한 W3C 가 OWL 을 표준 온톨로지 언어로 지정하여 앞으로의 온톨로지 데이터들은 OWL 로 기술되어질 전망이다.

본 논문에서는 이러한 온톨로지를 이용한 금형 설계 자동화를 위해 필요한 OWL 을 지원하는 대용량 온톨로지 저장소 연구에 초점을 둔다.

2. 관련연구

2.1 온톨로지 저장소 요구사항

온톨로지 저장소가 온톨로지를 관리하기 위해 기본적으로 요구되는 사항은 다음과 같다.[2]

- Legacy Database 의 접근을 위한 Interface
- 분석적인 질의 지원
- 임시적인 저장공간과 질의문
- 일관성 Checking

- 신뢰성 Management
- Legacy Data 와 지식을 통합하기 위한 기술
- 다수의 Mediator 와 Knowledge Server 을 연결하기 위한 Bridge

2.2 기존 온톨로지 저장소 비교 분석

기존의 대표적인 온톨로지 저장소는 KAON 과 DLDB 가 있다. 이들은 RDF, DAML 등의 온톨로지 언어를 저장 관리하는 저장소로서 연구목적으로 연구가 진행되고 있다.

2.2.1 KAON

KAON[3]은 온톨로지 기반의 어플리케이션을 개발하기 위한 기본적인 프레임워크로 여러 모듈들을 제공한다.

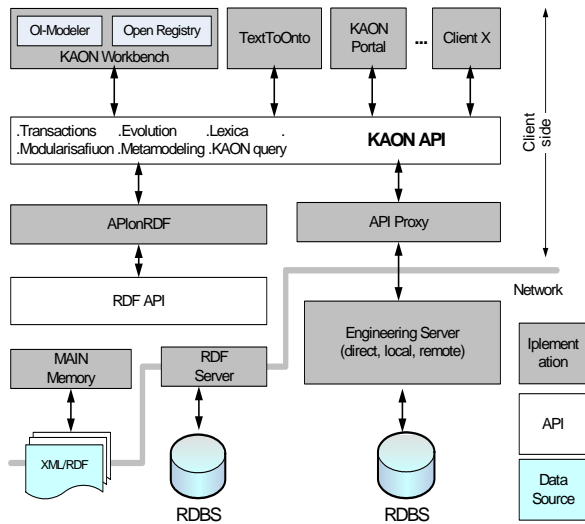


Fig. 1 KAON Framework

KAON 을 구성하는 각각의 모듈에 대해서 살펴 보면 엔지니어링 서버는 관계형 데이터베이스 (RDB)에 기반한 KAON 온톨로지를 위한 저장 메커니즘, 온톨로지 엔지니어링을 위한 것들, 응집성 탐지를 통한 클라이언트 사이드 캐시, 온톨로지 엘리먼트의 벌크 로딩과 분산 변화 탐지 메커니즘 등을 가지고 있고, KAON API 는 온톨로지 진화, 온톨로지 모듈화, 동시적인 온톨로지 접근과 트랜잭션 프로세싱을 지원한다. 저장 메커니즘은 RDF 모델과 관계형 데이터베이스를 포함한다. KAON Query 는 질의 언어를 통하여 KAON 온톨로지에 대한 질의를 가능하게 하는 역할을 한다.

KAON REVERSE 는 관계형 데이터베이스를 온톨로지에 매핑하여 RDF 인스턴스를 생성하는 작업

을 한다.

이 외에도 어플리케이션 서버인 KAON SERVER, KAON API 구현에 직접 동작하는 OntoEdit 를 위한 플러그인인 KAONtoEdit 등등 온톨로지를 관리하기 위한 많은 모듈들을 제공한다.

2.2.2 DLDB

DAML+OIL 에 관한 지식기반 시스템으로 관계형 데이터베이스 관리 시스템을 기반으로 하여 추론을 가능하게 하는 시스템이다. DAML+OIL 의 기반이 되는 RDF 는 주어, 서술어, 목적어의 세 요소로 구성되며, DLDB[4]는 이러한 세 요소를 기반으로 FaCT[5]라는 추론기에 의해 추론된 클래스 및 프로퍼티의 계층 정보를 이용하여 DAML+OIL 형태의 온톨로지를 저장한다.

기존에 연구된 온톨로지 저장소들은 대부분 OWL 이전의 온톨로지 언어인 RDF 나 DAML+OIL 을 사용하고 있으며, 연구목적으로 진행되어 대용량의 온톨로지 처리에는 부적합하다는 문제점을 가지고 있다.

Table 1.에서 기존의 온톨로지 저장소들을 비교하였다.

| | KAON | DLDB | OntoServer |
|------------|---------------------------|-----------------------|---------------------|
| Storage | XML repository / RDBS | XML repository / RDBS | Client/Server Based |
| Versioning | Versioning | no Versioning | no Versioning |
| 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 |

Table 1 Comparison Ontology Repository

3. 온톨로지 저장소 구성

본 연구에서 제안하는 온톨로지 저장소는 Fig2와 같은 모듈들을 가지고 있으며, 이를 통해 온톨로지 저장소가 요구하는 조건들을 만족하고자 한다.[6]

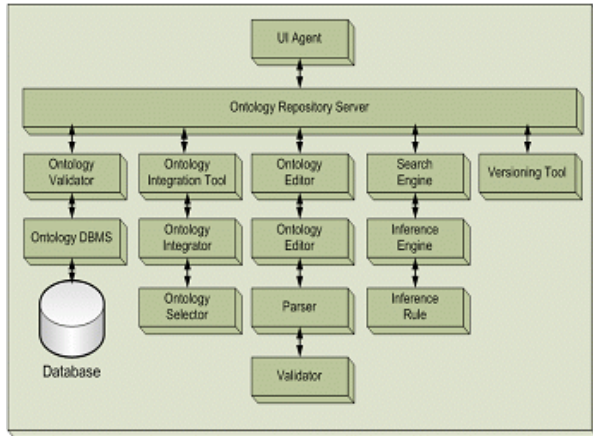


Fig. 2 Ontology Repository Framework.

3.1 온톨로지 저장에 대한 처리

OWL 형태의 온톨로지를 저장하기 위한 방법으로 OWL 문서 입력시 그 문서가 유효한 문서인지 판별하기 위해 Ontology Validator 에 입력된 문서를 전달한다. Validator 은 문서의 유효성을 판별한 뒤 유효한 문서라면 온톨로지 저장 Interface 에 문서를 전달한다.

저장 인터페이스는 입력 받은 OWL 형태의 문서에서 계층구조를 추출하여 각각의 클래스와 프로퍼티에 ID 를 부여하고, 이 정보를 XML 형태로 변환하고 XML 저장소에 저장한다. 온톨로지 데이터는 클래스와 프로퍼티에 대한 ID 와 함께 관계형 데이터 베이스에 저장된다.

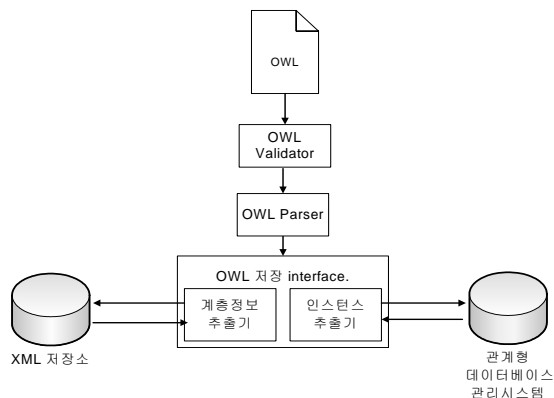


Fig. 3 Ontology Store Structure.

3.2 온톨로지 질의 처리

사용자로부터 주어지는 질의는 OWL-QL[7]을 사용하며 XML 저장소와 관계형 데이터베이스에 저장되어 있는 OWL 형태의 온톨로지 데이터에 대한 검색을 수행 한다.

질의 처리를 위해 필요한 클래스 및 프로퍼티에 대한 정보를 XML 저장소로부터 추출하며, 추출된 클래스 및 프로퍼티의 인스턴스는 계층 정보 이외의 데이터가 저장되어 있는 관계형 데이터로부터 SQL 질의를 이용하여 추출한다. 추출된 인스턴스는 OWL-QL 질의 패턴에 맞게 변환된 후 이를 사용자에게 전달한다.

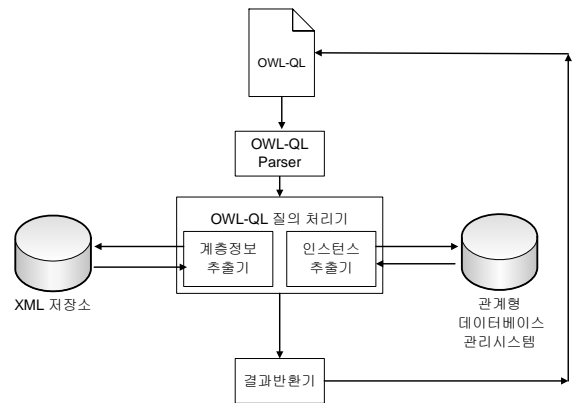


Fig. 4 Ontology Query Structure.

4. 구현

금형 설계 자동화를 위한 온톨로지인 Gate 크기에 대한 온톨로지를 Protégé 를 사용하여 OWL 형태로 작성하고 이를 저장하였다. Fig.5 는 실험을 위해 작성된 Gate 의 크기에 대한 OWL 형태로 작성된 온톨로지이며, Fig.6 은 그 실행과정을 나타내고 있다.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="gatesize"/>
  <owl:DatatypeProperty
    rdf:ID="KB_238728_DatatypeProperty_6">
    <rdfs:range>
      <owl:DataRange>
```

```

<owl:oneOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="limit_stress">
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Datatype
Property"/>
  <rdfs:domain rdf:resource="#gatesize"/>
  </owl:FunctionalProperty>
  <gatesize rdf:ID="gate">
    <limit_stress
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>20000.0</limit_stress>
  </gatesize>
</rdf:RDF>

```

Fig. 5 Gate Size Ontology

```

C:\Windows\System32\cmd.exe
C:\wis>Fullloader c:\data\gatesize.owl c:\db\wonto1.ndb
C:\wis>SET FPATH=file:WWWc:\data\gatesize.owl
C:\wis>java.exe -cp lib\jg13.1.0.jar;LIB\fact-client.jar;LIB\kerces.jar;lib\jena.jar;lib\getopt.jar;lib\jxap.jar;lib\jracer.jar;lib\rdf-api.jar;lib\shoe.jar;classes\edu.lehigh.cse.swat.FactDb.DamLoader file:WWWc:\data\gatesize.owl c:\db\wonto1.ndb
com.hp.hpl.jena.rdf.arp.ParseException: (E201) Syntax error when processing "20000.0".
Encountered "20000.0" Was expecting one of:
end element tag
property attributes
attribute rdf:type

time in database: .000 seconds.

time used: .953 seconds.
Exiting...
C:\wis>

```

Fig. 6 Ontology Store Execute

본 실험에서는 관계형 데이터베이스 MS Access를 사용하였고 fig.7에서 온톨로지 데이터가 저장된 것을 확인할 수 있다.

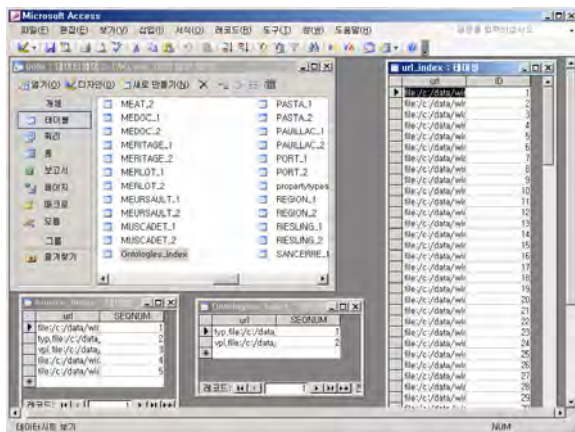


Fig. 7 MS Access Database

5. 결론

본 연구에서는 금형 설계 자동화를 위해 사용되는 온톨로지를 위해 기존에 연구된 온톨로지 저장소들의 단점을 보완하며, W3C가 지정한 표준 온톨로지 언어인 OWL에 기반한 온톨로지 저장소 아키텍처 설계 방법을 연구하였다. 향후 연구로는 제안한 온톨로지 저장소를 효과적이고 효율적으로 구축하기 위한 전략적 방안과 OWL 형식의 온톨로지 뿐만 아니라 웹 상에 존재하는 RDF, DAML, DAML+OIL 등의 다른 형식의 온톨로지들을 지원하는 온톨로지 저장소에 대한 연구가 진행될 예정이다.

참고문헌

1. W3C, OWL Web Ontology Language Guide, <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>
2. Jin Pan, Stephen Cranefield and Daniel Carter, "A Lightweight Ontology Repository", ACM 1-58113-683-8/03/0007
3. KAON – An Overview Karlsruhe Ontology Management Infrastructure, <http://kaon.semanticweb.org/documentation>
4. Z. Pan and J. Heflin. "DLDB: Extending Relational Databases to Support Semantic Web Queries." In Workshop on Practical and Scaleable Semantic Web Systems, pp. 109-113, 2003.
5. I. Horrocks. The FaCT System. In Proc. of Automated Reasoning with Analytic Tableaux and Related Methods: International Conference, pp. 307-312, 1998.
6. Wonder Web – Ontology Infrastructure for the Semantic Web, <http://wonderweb.semanticweb.org>.
7. 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.