

단계적인 온톨로지 정규화 기법의 제안

최윤호*, 김학수*, 김종진*, 이승미**, 손진현*

한양대학교 컴퓨터공학과*

한양대학교 컴퓨터공학과 BK21 AIS 사업팀**

e-mail : yhchoi@database.hanyang.ac.kr, hagsoo@cse.hanyang.ac.kr, jjkim@database.hanyang.ac.kr, smlee@database.hanyang.ac.kr, jhson@hanyang.ac.kr

A Step-by-Step Approach for Ontology Normalization

Yun Ho Choi*, Hak Soo Kim*, Jong Jin Kim*, Seung Mi Lee**, Jin Hyun Son*

Dept. of Computer Science & Engineering, Hanyang University*

BK21 AIS Team, Dept. of Computer Science & Engineering, Hanyang University**

요 약

인간의 지식을 컴퓨터에 저장하기 위한 연구 중 하나로 온톨로지가 등장하게 되었다. 온톨로지는 최근 주목 받고 있는 시맨틱 웹을 구현하기 위해 필수적으로 필요할 뿐만 아니라 다른 형식으로 저장된 정보 사이의 공유를 위해서도 많은 연구가 진행되고 있다. 특히 온톨로지의 초기 설계 후 변경을 하기 위해서는 매우 큰 비용이 소모된다는 점에서 온톨로지 설계에 대한 연구 역시 진행되고 있다. 본 논문에서는 온톨로지 설계를 위한 단계적인 온톨로지 정규화 기법을 제안한다. 제시된 온톨로지 정규화는 기존의 제시된 데이터베이스 정규화 기법을 활용하여, 단계적으로 더 엄격한 정규화가 가능하도록 설계되어 있다. 이러한 정규화 기법의 적용을 통해서 비효율적인 온톨로지 설계를 수정하고 짜임새 있는 설계가 가능하도록 온톨로지를 수정하는 기법을 제안한다.

1. 서론

1989년 Tim Berners-Lee에 의해 제안된 월드 와이드 웹은 정보통신 기술과 함께 발전함에 따라 많은 정보와 웹 문서가 인터넷상에 존재하게 되었으며 검색엔진을 통해서 이러한 정보들의 검색이 가능하게 되었다. 하지만 사용자가 임의로 제작한 문서들간의 연결로 이루어져 있으며 문서에 대한 의미정보를 담고 있지 않은, 즉 메타데이터를 가지지 않은 문서로 이루어진 현재의 인터넷 환경으로는 사용자의 요청을 의미적으로 분석하고 정보를 제공하는데 많은 어려움이 있다. 이러한 문제점을 극복하고 현재의 웹을 한 차원 더 발전시켜서 사용자가 원하는 정보를 의미적으로 분석해서 제공해 주는 시맨틱 웹[1]이 등장하게 되었다.

시맨틱 웹을 구현하기 위해서는 온톨로지의 존재가 꼭 필요하다. 온톨로지는 사람이 사용하는 용어에 대한 각각의 의미정보와, 그 용어와 용어사이의 관계를 프로퍼티로 연결하여 정의하게 된다. 이러한 온톨로지를 정의하여 각각의 문서에 연결하고, 운영하므로써 사용자의 요청의 의미를 온톨로지를 통해 기계가 이해하고, 단순히 키워드상으로 연결된 웹페이지를

연결하는 것이 아니라 그 페이지와 의미상으로 연결되는 페이지를 연결해 주는 것이다.

온톨로지의 시작은 인간이 가지고 있는 지식을 컴퓨터 상에 저장해서 자동화된 처리를 할 수 있게 하기 위하여 등장하게 되었다. 이러한 온톨로지의 특징은 데이터베이스의 특징과 유사한 점을 가지고 있다. 또한 데이터베이스의 설계에 관한 연구는 수년간 진행되어 어느 정도 체계화 되어 있고 검증되어 있다.

본 논문에서는 단계적인 온톨로지 정규화 기법을 제안한다. 논문에서 제시된 정규화 기법은 단계적인 적용을 통하여 더 엄격하고 제한적인 설계를 가능하게 해준다. 이러한 정규화 기법의 온톨로지 적용을 통하여 온톨로지의 설계와 구조를 더 짜임새 있고 잘 정의되도록 해준다.

본 논문의 구조는 다음과 같다. 제 2 장에서는 관련 연구에 대한 내용으로, 온톨로지에 대한 기본적인 연구와 온톨로지 설계에 대한 연구에 대하여 알아보며, 제 3 장에서 정규화 기법에 대하여 설명한다. 제 4 장에서는 정규화 기법의 단계적인 적용에 대하여 설명한다. 제 5 장에서는 결론을 맺는다.

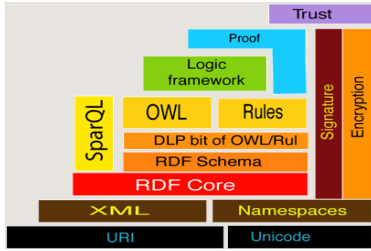
2. 관련연구

Tim Berners-Lee에 의하여 제안된 시맨틱 웹의 구성 요소는 그림 1과 같다. 시맨틱 웹의 구현을 위해서 정보 저장 및 처리를 위해 기존에 널리 사용되고 있는 XML 기술이 사용되었으며, 온톨로지의 구현을 위해서 RDF[2]와 OWL[3]이 W3C에 의해서 제안되었

본 연구(논문)는 산업자원부 지원으로 수행하는 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)의 일환으로 수행되었습니다.

이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2007-313-D00747).

으며, 이러한 온톨로지의 쿼리를 위한 언어인 SPARQL[4]이 제안되었다.



(그림 1) 시맨틱 웹의 구조

온톨로지의 구현을 위해 가장 기초가 되는 것은 RDF 로, 이는 특정 자원에 대한 메타 데이터를 기술하는 XML 기반의 프레임워크이다. RDF 는 레코드를 하나의 기술 단위로 취급해온 기존의 방식과는 달리 자원, 속성, 속성값을 하나의 단위로 취급하는 이른바 “Triple” 개념이 핵심이다. 자원 속성 표현의 세분화로 인해 자원에 대한 좀 더 정교한 기술이 가능해지고, 자원들 간의 관계 설정이 속성을 통해 무한으로 가능하게 되어진다. 각각의 자원들은 URI 를 통해 고유 식별자를 가지게 된다. 그리고 자원을 기술하는 속성명 또한 고유한 URI 를 통해 XML Namespace 에 정의된 속성을 사용함으로써 상호간 의미 충돌을 막는다. 속성의 값으로는 다른 URI 가 지정될 수 있으며, 속성 값으로 지정된 자원 역시 다시 기술의 대상이 되기 때문에 그 자원에 대한 속성과 속성 값을 다시 부과할 수 있다.

웹온톨로지 설계를 위하여 발표된 OWL 언어는 RDF 의 문제점인 주요 관계를 지원하는 표현력이 부족한 점을 보완한 온톨로지 언어이다. OWL 은 기존에 온톨로지를 위해 발표된 DAML+OIL 에 기반을 둔 언어로 기존의 온톨로지 구축 경험을 토대로 개념의 일관성을 확보하여 클래스와 프로퍼티의 개념 및 그들 사이의 관계가 더 명료하게 정의되도록 정리한 온톨로지 언어이다. 즉 OWL 은 시맨틱 웹 형성의 기반인 온톨로지 개발에 필요한 언어 구조를 재정립한 결과로 볼수 있으며 RDF 에서 지원하지 않는 프로퍼티간의 관계성에 대한 명확한 정의와 강한 제약을 설정해 줄 수 있다.

OWL 언어는 표현력이 서로 다른 세 가지의 하위언어로 구성된다. 첫번째로 OWL Lite 는 클래스 분류 계층과 간단한 제약 사항 표현을 필요로 하는 사용자들을 위한 언어이다. 두번째로 OWL DL 은 계산학적 완전성과 결정가능성을 유지하면서 최대의 표현력을 활용하고자 하는 사용자에게 적합하다. 마지막으로 OWL Full 의 경우는 어떠한 표현력의 제약사항 없이 RDF 와 OWL 의 풍부한 문법을 모두 활용하고자 하는 사용자에게 적합하다.

RDF 와 OWL 문법에 맞춰서 설계된 온톨로지는 XML 문서형태로 기계에 저장된다. 이러한 온톨로지에 사용자의 쿼리를 보내고 결과값을 받기 위해서

SPARQL 이 등장하게 되었다. SPARQL 은 관계형 DB 에서의 SQL 과 마찬가지로 온톨로지로부터 사용자가 원하는 데이터를 가져오기 위해서 사용하는 쿼리 언어이다. 기본적인 언어 형태를 SQL 언어와 매우 유사하며 온톨로지 특성에 맞춰 WHERE 절의 내용에 온톨로지 Triple 이 들어가는 것이 SQL 언어와 의 가장 큰 차이점이다. SPARQL 을 지원하는 온톨로지 추론틀을 사용해서, 사용자는 온톨로지로부터 원하는 정보를 가져올 수 있다.

온톨로지의 효율적인 설계를 위하여 기존 데이터베이스 설계에서 진행되었던 연구를 온톨로지의 특성에 맞춰서 적용하는 연구[5] 역시 진행되어 왔다. 이러한 연구는 기존의 설계에서 문제되어 왔던 부분을 변환하여 더 좋은 온톨로지 설계를 유도하는 연구이다. 또한 온톨로지의 논리적 모델에서의 정규화에 대한 연구[6] 역시 진행되어 왔다. 이러한 연구는 본 논문과 정규화 기준을 어디에 두느냐에 따른 차이를 가지고 있다. 본 논문에서는 온톨로지 설계 설계에 있어서 클래스와 그 클래스를 관계짓는 프로퍼티를 기준으로 정규화를 제안한다.

3. 정규화 기법

본 논문에서 제시하는 온톨로지 정화는 두개의 클래스와 그 클래스간에 연결되는 프로퍼티를 기준으로 정규화를 제시하며 4 단계의 정규화를 제시한다. 제 1 정규화는 두 클래스간의 역방향 프로퍼티가 존재하지 않도록 보장하는 것이며, 제 2 정규화는 프로퍼티가 토달 프로퍼티임을 보장하는것이다. 제 3 정규화는 프로퍼티가 M-N 관계를 가지는 프로퍼티가 아님을 보장하는 것이며, 마지막으로 제 4 정규화는 프로퍼티가 한 개의 domain 에 대하여 오직 한 개의 range 값만을 가지도록 보장하는 것이다.

정규화를 제안하기 위해 앞서 정규화 대상 검출을 위한 SPARQL 의 기본 쿼리를 제시하겠다. 각각의 정규화에서 제시되는 정규화 대상 검출 쿼리는, 그림 2 에서 제안되는 기본 쿼리에 기반하여 동작하는 쿼리이다.

```
{
  ?p_n      rdf:type      rdf:property .
  ?Ca      rdf:type      rdf:class .
  ?Cb      rdf:type      rdf:class .
  ?Ia_n    rdf:type      ?Ca .
  ?Ib_n    rdf:type      ?Cb .
}
```

(그림 2) Basis Query

3.1. 제 1 정규화

제 1 정규화는 두 클래스간의 역방향 프로퍼티가 존재하지 않음을 보장하는 것이다. 두 클래스가 프로퍼티에 의하여 관계되어져 있을 때, 한 프로퍼티에 의해서 domain 과 range 관계가 설정되어 있고, 다른 프로퍼티에서는 이 domain 과 range 관계가 반대로 형

성되어 있다면, 두개의 프로퍼티를 하나의 프로퍼티로 통합해서 역방향 프로퍼티를 제거해주는 것이 제 1 정규화의 핵심이며 이러한 변환작업을 통하여 두개의 클래스 사이에 프로퍼티 관계가 오직 하나의 프로퍼티로 관계되어지도록 보장하는 것이 제 1 정규화의 핵심이다.

```
SELECT ?p1
WHERE {
    ?fa_1 ?p1 ?fb_1 .
    ?fb_2 ?p2 ?fa_2 .
}
```

(그림 3) 제 1 정규화 검출 쿼리

그림 3 은 제 1 정규화 대상을 검출하기 위한 SPARQL 쿼리이다. 정규화 대상이 되는 프로퍼티를 검출하기 위하여, 임의의 두 클래스 간에 domain 과 range 과 정방향과 역방향으로 동시에 존재하는 프로퍼티의 조건을 WHERE 에 정의한 것을 확인할 수 있다.

3.2. 제 2 정규화

제 2 정규화는 프로퍼티가 토탈 프로퍼티임을 보장하는 것이다. 여기서 토탈 프로퍼티란 두 클래스 사이에 관계하는 하나의 프로퍼티가 있다고 가정할 때, 그 프로퍼티의 Domain 인스턴스는 모두 프로퍼티의 값을 가지고 있는 프로퍼티를 말하는 것이다. 이처럼 토탈 프로퍼티가 존재할 경우, Domain 으로 존재하는 클래스는 모두 프로퍼티에 대한 값을 가지는 비슷한 성질의 인스턴스를 모은 클래스 임이 보장되는 것이다.

```
SELECT ?fa_1
WHERE {
    ?fa_1 ?p ?fb_1 .
}
// is equal
SELECT ?fa_1
WHERE {
    ?fa_1 rdfs:type ?Ca .
}
```

(그림 4) 제 2 정규화 검출 쿼리

그림 4 는 제 2 정규화 대상을 검출하기 위한 SPARQL 쿼리이다. 제 2 정규화 대상은 단일 쿼리로써 대상을 검출하는 것이 어려우므로, 두번의 쿼리 실행을 통한 값의 차이를 통해서 정규화 대상을 검출한다. 첫번째 쿼리는 한 프로퍼티에 소속되는 Domain 인스턴스를 추출하는 쿼리이며, 두번째 쿼리는 인스턴스들 중 한 클래스에 소속된 공통 인스턴스를 추출하는 쿼리이다. 이 두가지 쿼리의 결과값을 비교하면 프로퍼티가 토탈 프로퍼티인지 검사할 수 있으며 이를 통하여 정규화 대상을 검출할 수 있다.

3.3. 제 3 정규화

제 3 정규화는 프로퍼티가 M-N 프로퍼티가 아님을 보장하는 것이다. M-N 관계의 예를 들면, 최소 두개의 Domain 인스턴스가 하나의 프로퍼티에 대하여 각각 두개의 동일한 Range 인스턴스를 가질 경우에 이를

M-N 관계가 성립한다고 한다. 이러한 M-N 관계를 가지는 프로퍼티가 온톨로지 내에 존재하게 될 경우에, 전체 온톨로지 복잡도와 온톨로지를 운영하는 데 있어서의 추론 비용이 크게 상승하게 된다. 이를 방지하기 위해서 제 3 정규화에서는 M-N 관계를 분리하여 임의의 클래스를 생성해서 1-N 관계와 N-1 관계의 프로퍼티로 분리하여 M-N 관계를 제거하게 된다.

```
SELECT ?p
WHERE {
    ?fa_1 ?p ?fb_1 .
    ?fa_1 ?p ?fb_2 .
    ?fa_2 ?p ?fb_1 .
    ?fa_2 ?p ?fb_2 .
}
```

(그림 5) 제 3 정규화 검출 쿼리

그림 5 는 제 3 정규화 대상을 검출하기 위한 SPARQL 쿼리이다. M-N 프로퍼티의 최소 조건을 만족하는 Triple 을 추출하기 위해서 두개의 Domain 과 두개의 Range 값을 모두 가지는 프로퍼티를 검출하는 쿼리가 그림 5 에 제시된 것을 확인할 수 있다. 이를 통하여 제 3 정규화 대상을 추출하여 정규화할 수 있다.

3.4. 제 4 정규화

제 4 정규화는 프로퍼티가 한 개의 Domain 에 대하여 오직 한 개의 Range 값만을 가지도록 보장하는 것이다. 이를 위하여 한 개의 Domain 에 대하여 두개 이상의 Range 인스턴스 값을 가지는 프로퍼티를 검출하고 이러한 프로퍼티가 존재할 경우에 프로퍼티의 방향성을 바꿔줌으로써 제 4 정규화의 규칙을 만족하도록 변환해준다. 이를 통해 온톨로지내의 프로퍼티 카디널리티값이 1 이 되도록 보장해 주며 이는 OWL Lite 언어의 최소한의 문법규칙을 만족하게 해준다.

```
SELECT ?p
WHERE {
    ?fa_1 ?p ?fb_1 .
    ?fa_1 ?p ?fb_2 .
}
```

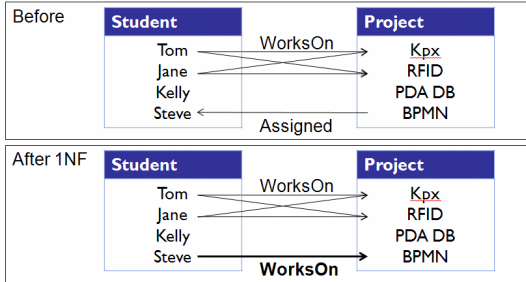
(그림 6) 제 4 정규화 검출 쿼리

그림 6 은 제 4 정규화 대상을 검출하기 위한 SPARQL 쿼리이다. 제 4 정규화 대상은 같은 Domain 과 같은 프로퍼티에 대하여 Range 값이 두개 이상인 경우에 정규화 대상이 되므로 이를 위한 최소조건인 두개를 가지는 프로퍼티를 검출하는 쿼리를 확인할 수 있다. 이를 통해서 제 4 정규화 대상을 검출하고 이를 변환하여 정규화 과정을 진행한다.

4. 단계적인 적용

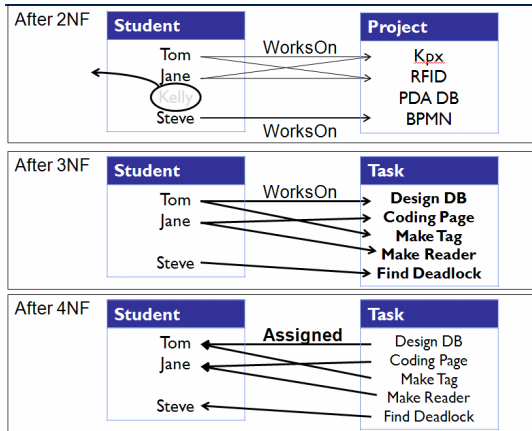
정규화 과정에 있어서 가장 중요하게 고려해야 할 사항 중 하나는 정규화 과정이 단계적으로 진행되어야 하며, 정규화 과정이 진행될수록 더욱 엄격하고 제한적인 온톨로지가 설계되어야 한다는 점이다. 이

러한 점을 고려하여 정규화 과정을 설계해야만 온톨로지 설계자는 정규화 기법의 단계를 온톨로지의 특성과 사용성을 고려하여 활용할 수 있기 때문이다.



(그림 7) 제 1 정규화의 적용

그림 7 은 최초 설계된 온톨로지의 일부분에서 제 1 정규화 과정을 적용시킨 온톨로지의 예이다. WorksOn 프로퍼티와 Assigned 프로퍼티 모두가 두 클래스간의 관계를 만들고 있으며 이를 제 1 정규화에 맞게 변환한 것을 확인할 수 있다. 이처럼 불필요한 프로퍼티를 적절히 변환하여 제 1 정규화를 만족하는 프로퍼티로 만들 수 있다.



(그림 8) 단계적인 정규화 적용

그림 8 은 제 4 정규화 과정까지 단계적으로 진행된 온톨로지의 예이다. 각각의 과정마다 정규화 기법에 제시된 SOARQL 을 활용하면, 정규화 대상을 검색할 수 있으며 이를 각각 변환하여 정규화 규칙에 맞는 온톨로지로 변환할 수 있다.

5. 결론

기존의 웹에 한계를 극복한 시맨틱 웹을 구축하기 위해서는 온톨로지의 존재가 필수적이다. 이러한 온톨로지의 활용도를 높이기 위하여 온톨로지 설계에 대한 연구가 계속 진행되고 있다. 이러한 연구를 통하여 온톨로지의 비효율적이고 복잡한 설계를 줄이고

활용도를 더욱 높일 수 있다.

향후 연구 과제로는 온톨로지의 의미정보를 토대로 분석하고 설계상의 문제점을 찾아내는 연구가 선행되어야 한다. 단순히 구조만을 고려한 온톨로지 설계에 대한 연구가 아닌 온톨로지가 설계되고 구현되는 이유를 고려하여 이러한 연구가 진행되어야 한다. 이러한 연구는 진정한 시맨틱 웹의 구현과 인공지능을 가진 인터넷 환경을 더욱 앞당겨줄 것이다.

참고문헌

- [1] T.Berners-Lee and M.Fischetti, "Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor"
- [2] RDF Primer "<http://www.w3.org/TR/rdf-primer/>"
- [3] Ian Horrocks, Peter F. and Frank van Harmelen "From SHIQ and RDF to OWL: The Making of a Web Ontology Language"
- [4] SPARQL Query Language for RDF "<http://www.w3.org/TR/rdf-sparql-query/>"
- [5] F.Mostowfi and F.Fotouhi, "Improving Quality of Ontology : An Ontology Transformation Approach"
- [6] Denny Vrandečić and York Sure "How to Design Better Ontology Metrics"