

## 인터넷 온톨로지 데이터 변화에 따른 질의 처리 모델 개발

오성균\*, 김병곤\*\*

### 요약

사용자들에게 더욱 효율적이고 정확한 검색 결과를 제공하기 위하여, 인터넷시스템은 시맨틱웹과 같은 차세대 기술에 더욱 의존하게 된다. 시맨틱웹을 구현하는데 가장 중심적인 요소가 온톨로지이다. 온톨로지의 데이터는 시간이 흐르면 변하는 경향이 있으며, 변화 제어 시스템을 통하여 변화의 내용들을 추적하고 반영할 수 있다. 특히, 온톨로지가 변하기 전의 버전에 대하여 작성된 질의는 새로운 버전의 온톨로지에 대하여 실행되지 못할 가능성을 지니고 있다. 따라서, 이러한 경우에는 새로운 버전의 온톨로지에 대하여 실행이 가능하도록 질의를 수정하여 가장 최신의 데이터를 검색하도록 해야 한다. 이와 관련한 많은 연구들이 진행되었지만 아직 해결하지 못한 문제들이 있으며, 본 연구는 이를 해결하기 위하여 온톨로지 클래스와 속성에 대하여 변화 그래프를 제안하였다. 그래프는 온톨로지 데이터가 변화하여 새로운 버전이 생성될 때 만들어지도록 하였으며, 이를 이용하여 질의를 변형하여 실행하도록 하였다.

키워드 : 시맨틱웹, 온톨로지, 질의처리, 온톨로지 버전

## Query processing model for Internet ontology data change

Sung-Kyun Oh\*, Byung-gon Kim\*\*

### Abstract

To provide more efficient and exact search result, internet systems will rely more and more on semantic web. Ontology is one of the important methods for implementation of semantic web. Ontology is used to implement an explicit formal vocabularies to share. However, important problems rise when dealing with ontology. Ontologies are typically subject to change because they are living. In order to handle ontology data change situation, a version handling system is needed to keep track of changes. For example, the queries subject to the previous ontology may become inconsistent and must be updated according to the newest version of ontology. Although many research was done in this area, there are still many problems to overcome. In this paper, we propose class and property transition graph for query transformation. The graph is created when ontology data is changed and applied to query transformation.

Keywords : Semantic Web, Ontology, Query processing, Ontology version

### 1. 서론

차세대 인터넷 서비스의 발전 방향은 인터넷 상의 콘텐츠를 사람이 직접 읽고 해석하고 판단하여 처리하는 방식이 아닌 컴퓨터가 콘텐츠의 의미를 해석하고 판단하여 결과를 도출할 수 있는 환경인 시맨틱웹을 구현하고자 하는 방향으로 진행되고 있다.[1] 이를 구현하고자 할 때 가장 핵심적인 방법론으로 대두되고 있는 것이 온톨로지이다. 온톨로지는 특정영역의 개념과 그들 간의 관계를 정의하는 명세이며, 추론규칙을 포함하여 새로운 사실을 자동으로 추출하거나 제약 조건에 어긋난 내용들을 찾아 낼 수 있다. 시

※ Corresponding Author : Byung-gon Kim

Received : December 04, 2015

Revised : January 30, 2016

Accepted : February 20, 2016

\* Seoil University, Dept. of Computer Software

Tel: +82-2-490-7398 , Fax: +82-2-490-7396

email: skoh@seoil.ac.kr

\*\* Bucheon University, Dept. of e-Business

Tel: +82-32-610-3460 , Fax: +82-32-610-3207

email: bgkim@bc.ac.kr

맨틱웹에서 온톨로지 데이터는 시간의 변화에 따라 여러 다른 버전으로 변형될 수 있다.[2] 시맨틱웹은 이 모든 변화들을 정확하게 인식하고 다룰 수 있어야 한다. 실제적으로 많은 응용 분야에서 온톨로지는 정적이지 않고 시간이 흐름에 따라 변화하는 것을 볼 수 있다. 온톨로지의 클래스나 속성의 생성이나 삭제, 이동, 변형 등이 이에 해당한다. 이러한 변화는 데이터베이스에서의 스키마 변화와 유사한 개념이다. 데이터베이스에서 스키마가 변화하면 이를 바탕으로 생성된 테이블과 테이블내의 인스턴스에 변화가 생긴다. 마찬가지로 온톨로지의 클래스나 속성에 변화가 생기면 기존의 온톨로지에 연결되어 있던 인터넷 자원과의 관계도 변화되며, 원래의 환경에서 생성된 질의의 처리가 불가능해진다. 이는, 과거에 생성된 질의는 과거 온톨로지에 호환되는 데이터에 대하여만 질의를 처리한다는 의미이다. 예를 들어, 컴퓨터 도서관 시스템을 통하여 전세계에 있는 도서를 찾고 도서관에 있는 자료들을 찾는데 이용되고 있다. 인터넷에는 점점 더 많은 문서와 자료들이 올라오고 있다. 현재의 온톨로지 버전을 가지고 사용하는 시스템이 시간이 지나 다른 버전의 온톨로지를 가지고 구축된 시스템들 간의 상호호환적인 소통에 있어서 문제가 발생 할 수 있다. 하지만 대부분의 검색 사용자들은 가장 최신 온톨로지에 대한 검색결과를 원하므로 이를 지원해줄 수 있는 방법이 필요하다. 만약, 여러 개의 다른 온톨로지에 대하여 데이터 변화에 따른 데이터 자원간의 관계를 정확히 분류하고 처리하지 못하면 시맨틱웹은 여러 온톨로지 버전들간에 비호환성, 데이터 충돌, 질의처리 불가능 등의 문제 등을 야기할 수 있다.

본 연구의 목표는 이러한 문제점들을 해결하고, 가장 최신의 온톨로지를 기반으로 처리된 질의 결과를 제공하는 방법을 제시하는 것이다. 이를 위하여 데이터 변화에 따른 온톨로지의 변화가 있을 때에는 온톨로지 데이터 변화 규칙을 제시하며, 이를 바탕으로 클래스와 속성에 대한 변화그래프를 생성하도록 하였다. 생성된 그래프를 통하여 항상 최신의 버전에 대하여 질의 처리가 가능하도록 질의를 수정하는 메커니즘을 구현할 수 있다. 2장에서는 온톨로지의 개념과 데이터변화에 대한 기존의 연구를 설명하였고, 3

장에서 변화그래프를 이용한 질의 처리 개념을 제시하였다. 4장에서 이에 대한 질의 처리 실험을 수행하여 결과를 설명하고 5장에서 결론으로 마무리 하였다.

## 2. 관련연구

### 2.1 온톨로지

온톨로지는 인터넷상의 문서에 포함된 개체들에 대한 개념과 개체들 간의 관계를 정의한 것이다. 온톨로지의 정의는 “공유된 개념에 대한 형식적이고 명시적인 설명 혹은 서술”이라 한다.[3] 공유된 개념이란 실세계에 존재하는 개체나 현상 등에 대하여 특정 그룹내의 구성원들이 동의하여 함께 사용되는 개념이다. 형식적이라는 의미는 개념에 대한 정의를 기계가 스스로 읽고 해석할 수 있는 형태이어야 한다는 것이며, 명시적이란 개념과 더불어 개념에 대한 제약조건 등이 정의되어 있어야 한다는 것이다.

온톨로지를 구성하는 구성요소로는 개념(concepts), 관계(relations), 인스턴스(instances), 공리(axioms)가 있다. 개념은 도메인에 존재하는 엔티티의 집합을 의미하며, 관계는 개념들 혹은 개념의 속성간의 상호 관계를 의미한다. 인스턴스는 도메인의 개념을 반영하는 실세계의 실제 예이며, 공리는 개념의 사용을 제약하는 명시적인 규칙을 의미한다. 온톨로지는 실제 웹상에서 사람이나 소프트웨어 에이전트로 하여금 개념에 대한 구조를 제공하여 공유할 수 있으며, 이는 온톨로지를 통하여 구축된 지식을 다른 사람이나 그룹에서 재사용이 가능하도록 할 수 있다. 온톨로지가 완성되면 에이전트는 이를 이용하여 사용자가 원하는 검색결과나 처리를 제공할 수 있다. 사람과 사람간의 의사소통을 위해서 온톨로지를 이용할 수 도 있지만, 더 넓은 범위에서는 시스템과 시스템간의 상호작용과 시스템 공학 측면에서의 이용이 가능하다. 즉, 시스템이 요구하는 도메인의 요구사항 명시, 일관성체크, 정보획득 등에 사용될 수 있다.

온톨로지를 표현하기 위하여 사용되는 언어는 XML, RDF, OWL 등이 있다. 특히, RDF는 문장을 주어(Subject), 서술어(Verb), 목적어(Object)와 같은 트리플 구조로 표현한다. 주어,

서술어, 목적어는 리소스(Resource), 속성(Property), 값(Value)으로도 표현 한다.[4] 이러한 사실들을 표현하고 이용하기 위하여 필요한 것이 스키마의 개념이고 이를 위하여 만들어진 것이 RDFS(RDF Schema)이다. RDFS는 RDF를 기술하기 위해 필요한 어휘들의 의미와 어휘들 간의 의미적 관계를 기술하기 위해 W3C에서 초기에 제안한 언어이며, RDF와 같은 문법적 구조를 가지는 것이 장점이다. 이외에 스키마 표현 언어로는 DAML+OIL과 OWL 등이 있다.

## 2.2 온톨로지의 변화

온톨로지를 이용한 시맨틱검색을 위하여 여러 연구가 진행되었다. [5]에서는 온톨로지와 KB를 이용하여 검색결과를 추출하고 랭킹기법을 이용하여 처리하는 연구를 진행하였다. 이처럼, 웹정보의 규모가 급격히 팽창할수록 온톨로지를 이용하여 지식정보를 구축하고 이를 통해 정보를 검색하는 경우가 증가할 것이다. 예를 들어, 온톨로지 정보를 이용하여 구축된 전자도서관을 통해 도서를 찾고 도서관에 있는 자료들을 찾는 데 이용할 수 있다. 그러나, 인터넷에는 점점 더 많은 문서와 자료들이 올라오고 있으며, 그것들을 검색하는 것이 점점 더 어려워지고 검색결과도 좋지 않다. 현재의 전자도서관들이 각각의 온톨로지 버전을 가지고 사용한다면, 시간이 지나 다른 버전의 온톨로지를 가지고 구축된 시스템들과의 상호호환적인 소통에 있어서 문제가 발생할 수 있다. 서로 상충되는 문장을 처리해야 한다면, 데이터의 중복, 시간차이에 따른 데이터의 불일치 등을 해결해야 한다. 이를 해결하기 위해서는 버전간의 매핑방법을 제공해야 하는데, 이를 위해서는 온톨로지 관리자가 두 버전간의 차이를 인지하고 적용 가능해야 한다. 이를 위하여 온톨로지를 개발하거나 수정하는 경우 개발자는 온톨로지의 여러 버전의 변화 이력을 참고할 수 있도록 변화 이력에 대한 저장과 접근이 용이하도록 시스템을 만들어야 하며, 수정 정책에 있어서도 추후에 발생할 문제의 소지들을 해결할 수 있도록 설계하여야 한다.

올바른 설계는 온톨로지 관리 시스템이 온톨로지의 일관성을 보존하기 위하여 온톨로지에 대한 수정작업을 하도록 설계되어 지는 것이다. 이러한 과정을 온톨로지 버전 관리라 하며 이는

온톨로지 관리 시스템이 온톨로지 버전간의 차이를 인지하고 관리하여 온톨로지의 변화를 관장할 수 있는 개념이다. 온톨로지가 변화하는 이유에는 도메인의 변화, 데이터 개념의 변화, 자원간의 관계 변화 등이 있다. 더 나아가, 새로운 요소나 속성이 추가되거나, 존재하는 요소나 속성이 제거되기도 한다.

온톨로지의 구성을 개념, 관계, 인스턴스 등으로 이루어진 엔티티 요소와 공리로 나누어 볼 때, 엔티티는 실제적으로 온톨로지의 클래스, 속성, 인스턴스, 데이터타입 등을 포함한다고 할 수 있다. 이전 버전의 온톨로지와 새로운 버전의 온톨로지를 엔티티 관점에서 비교해 보면 다음과 같은 경우로 나누어 볼 수 있다.

- 새로운 온톨로지에 나타난 요소가 이전 온톨로지에는 없는 경우
- 이전 온톨로지에 있는 요소가 새로운 온톨로지에는 없는 경우
- 동일한 이름의 요소가 두 개의 온톨로지에 동시에 존재하는 경우
- 실제로는 동일한 요소가 두 개의 온톨로지서 서로 다른 이름으로 존재하는 경우가 발생할 수 있다.
- 이전 온톨로지의 요소가 병합이나 분리된 경우

온톨로지의 데이터 변화과 관련된 연구는 변화된 내용을 인지하는 방향의 연구와 변화된 서로 다른 버전의 온톨로지간의 변화 항목들을 표현하고 저장하여 문제점을 해결하는 방향의 연구로 나눌 수 있다.

먼저 변화된 내용을 인지하는 연구는 새로운 정보를 인지하고 이를 온톨로지에 추가하는 방법으로서 일반 텍스트데이터, 데이터베이스, 다른 온톨로지 등의 정보소스에 포함되어 있는 정보를 대조하여 이를 적용하는 것이다. 텍스트데이터는 비구조적인 데이터를 포함하고 있기 때문에 정보추출의 과정이 필요하다. 외부의 온톨로지나 데이터베이스는 구조적인 정보소스이며 개념과 관계 인스턴스 등을 표현하고 있다. 그러나 기본 온톨로지와의 호환성 때문에 변환과정이 필요하다. 데이터베이스의 경우에는 데이터베이스 스키마와 엔티티의 개념을 온톨로지로 표현하고 나타낼 수 있도록 변환과정이 필요하다. 새로운 정보를 평가하기 위해서는 경험적인 여

러 가지 규칙들을 적용할 수 있다. 평가된 스키마와 인스턴스 요소들은 기존의 지식과의 관계를 형성하기 위한 관계 처리 루틴으로 넘어간다. 기존의 온톨로지 요소와의 일치 여부를 체크하고, 이를 적용하는 방식으로 진행한다.

버전시스템의 실제 예로 CVS가 있다.[6] CVS는 버전제어시스템을 다루기 위한 스크립트의 집합개념이며, 파일 단위로 버전 제어를 한다. 즉, 두 파일간의 구문적인 차이를 찾아내는 능력을 지닌다. 그러나 CVS는 개념이 아닌 구문수준에서 작동하며, 이는 온톨로지 엔티티의 복잡한 구조와 관계를 버전에 따라 제어하는 데는 한계가 있다.

서브버전(Subversion) 시스템[7]의 경우에는 인터넷상의 소스코드와 기타 자원들의 변화에 대한 관리를 수행한다. 그러나 기본적으로 파일간의 텍스트의 변화에 대한 제어 시스템이므로 온톨로지 스키마와 같은 구조적인 변화에 대한 대응을 할 수는 없다.

변화된 서로 다른 버전의 온톨로지를 구분하기 위하여 변화된 내용을 표현하고 저장하는 연구로서 [8]의 연구는 버전로그를 이용하는 연구이다. 버전로그는 서로 다른 버전들에 대한 기록이다. 버전로그는 온톨로지 개념의 서로 다른 버전을 보여주며, 이를 이용하여 변화의 관점에서 이러한 버전들의 해석이 가능하도록 하며, 시스템에서 온톨로지의 무결성을 보장하고 변화의 정의와, 변화를 감지하기 위해서 사용한다. 버전로그는 각 버전의 온톨로지의 개념을 지닌다. 각 버전은 어느 순간의 개념의 정의를 표현하도록 하였다. 온톨로지에서 생성된 각 클래스, 속성, 인스턴스에 대하여 버전로그를 생성한다.

온톨로지에서 변화가 생기면 새로운 버전 인스턴스가 EvolutionConcept 클래스의 인스턴스가 만들어지고, 이는 새로운 개념을 표현한다. EvolutionConcept 인스턴스는 생성된 개념에 대한 과거와 현재의 버전에 대한 리스트를 지닌다. 온톨로지에 존재하는 개념에 대한 변화가 생기면, 새로운 버전 인스턴스가 EvolutionConcept의 인스턴스로 생성된다. 버전인스턴스는 트랜잭션타임, 즉 수정이 생긴 순간의 시간을 지닌다. 버전변화가 일어난 온톨로지의 버전을 명시한다. 다음은 버전 온톨로지를 사용한 버전로그의 예이다.

```
<EvolutionConcept rdf:ID="fd42cc20">
  <refersTo rdf:resource="…/university#Student"/>
  <hasVersion>
    <ClassVersion rdf:ID="389a99b0">
      <hasTransactionTime>624</hasTransactionTime>
      <hasState>confirmed</hasState>
      <hasID>Student</hasID>
    </ClassVersion>
  </hasVersion>
  <hasVersion>
    <ClassVersion rdf:ID="389a99b1">
      <hasTransactionTime>628</hasTransactionTime>
      <hasState>confirmed</hasState>
      <hasID>Student</hasID>
      <subtypeOf rdf:resource="#fd42cc22" />
    </ClassVersion>
  </hasVersion>
</EvolutionConcept>
```

온톨로지의 변화를 표현하기 위하여 변화집합(Changeset)과 그래프를 사용한 연구도 있다.[9] 온톨로지간의 변화를 오퍼레이션들의 집합으로 정의하고 이를 변화집합의 개념으로 정리하였다. 변화집합은 그래프 형태로 표현하였고, 그래프의 노드는 새로운 버전의 온톨로지를 의미하며, 에지는 두 온톨로지간의 변화집합을 의미하도록 설계되었다. 시스템은 그래프를 참조하여 새로운 버전의 온톨로지를 적용한다.

[10]에서는 온톨로지가 하나의 버전에서 다른 버전으로 변화하는 경우에 변화된 내용을 상세하게 정리하여 이를 OWL 형태로 표현하여 질의 처리에 사용할 수 있도록 하였다. 그러나, 제시된 그래프나 문서를 통하여 항상 최신의 온톨로지를 적용한다는 보장은 할 수 없다.

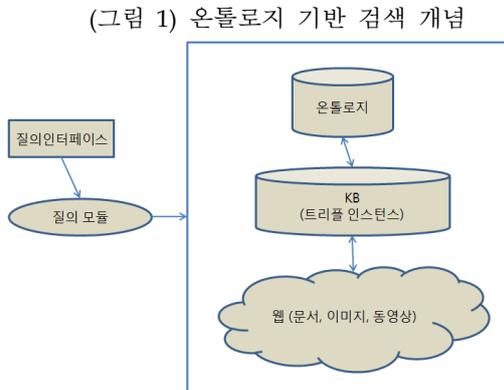
기존의 연구들을 종합해보면, 두 온톨로지간의 변화된 내용을 기술하고, 이를 저장하여, 질의 처리에 사용하는 방법을 사용하였다. 그러나, 세부적인 클래스, 속성, 도메인 등의 변화 시퀀스를 제공하는 방법을 사용하지는 않았거나, 변화된 내용에 대한 대체 방법을 제시하여 최신의 온톨로지에 대하여 질의가 처리되도록 하지는 못하였다. 온톨로지의 호환성을 고려하여 최신의 온톨로지를 이용하여 질의 처리하고자 한다면 변화된 내용이 새로운 온톨로지에서도 적용 가능하도록 대체 방법을 제시하여 질의가 가장 최신

의 정보를 결과로 도출하도록 유도해야 한다. 본 연구가 추구하고자 하는 방향이 이를 해결하는 방법을 제시하는 것이다.

### 3. 온톨로지 변화 제어

#### 3.1 온톨로지 기반 질의와 온톨로지 호환성

온톨로지 기반의 질의는 규칙 추론과 같이 정의된 규칙에 의해 기존에 존재하지 않는 새로운 정보를 생성하는 기능과 관계 추론과 같이 온톨로지 관계를 이용하여 검색의 범위를 넓히거나 좁히는 기능, 그리고 온톨로지의 내용을 검색결과로 출력하는 기능 등을 수행한다. 그리고 대부분 온톨로지와 연결된 인터넷 자원과 이들 간의 관계를 검색결과로 도출하게 된다.

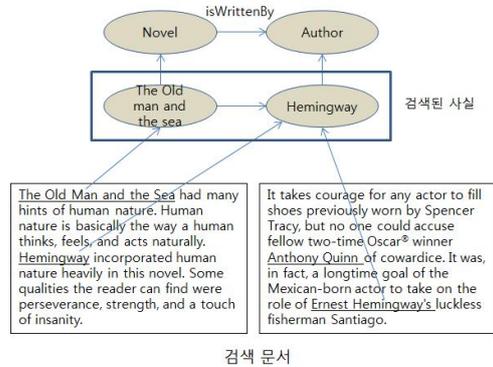


(Figure 1) Ontology based search concept

(그림 1)은 이러한 온톨로지 기반의 질의 처리 개념을 보여주고 있다. 사용자의 질의인터페이스로부터 생성된 질의가 질의 모듈을 거쳐 온톨로지와 지식베이스(KB)에 대하여 트리플(triple) 형태의 사실(fact)들이 검색되고 검색된 사실들을 가지고 웹을 검색하는 개념의 다이어그램이다. 예를 들어, 미국의 작가 “헤밍웨이”와 그의 작품 “노인과 바다”를 주제로 검색을 수행한다고 가정하면, 입력질의는 사용자의 질문으로부터 생성된다. 생성된 질의는 먼저 온톨로지와 지식베이스를 대상으로 실행된다. 온톨로지로부터 개체클래스를 검색하고, 개체클래스의 사실들 중에서 조건을 만족하는 사실들을 찾아낸다. 다

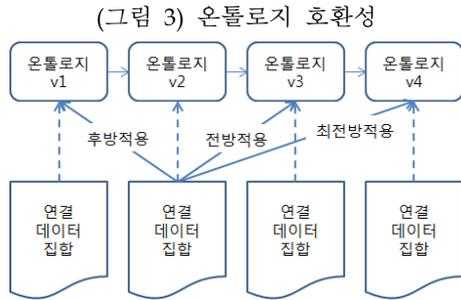
음으로, 지식베이스에서 검색된 사실들과 관계를 맺고 있는 사실들과 문서들을 검색하는 형태로 진행된다. (그림 2)는 검색된 사실과 연관된 문서들을 보여준다.

(그림 2) 온톨로지 기반 검색 결과 문서들



(Figure 2) Ontology based search results

온톨로지 기반의 질의를 수행함에 있어서 시간이 지나 다른 버전의 온톨로지를 가지고 구축된 데이터집합이나 시스템들 간의 질의 처리에서 상호호환적인 소통에 문제가 발생할 수 있다. 이를 온톨로지 호환성이라고 한다. 온톨로지 호환성이란 온톨로지 버전과 이와 연관된 데이터 소스간의 관계를 의미한다. 실제 시스템에서 우리는 동일한 도메인에 대한 동일한 데이터에 대하여 여러 가지 버전의 온톨로지를 적용할 수 있다. (그림 3)에서 보면, 만약 과거 버전의 온톨로지에 맞추어진 데이터를 새로운 버전의 온톨로지에 적용한다면 그러한 경우를 “전방적용(Prospective use)”이라고 한다. 반대로, 새로운 버전의 온톨로지에 맞추어진 데이터를 과거 버전의 온톨로지에 적용한다면 이를 “후방적용(Retrospective use)”이라고 한다.[2] 온톨로지의 변화의 상황에 따라 전방적용이 가능한 경우와 후방적용이 가능한 경우가 다를 수 있다. 본 연구에서는 특히 전방적용중에서 가장 최신의 버전의 온톨로지를 대상으로 수행되는 질의는 “최전방적용”이라고 정의 한다.



(Figure 3) Ontology compatibility

<표 1>에는 온톨로지 버전의 변화에 따른 카테고리별 호환성의 문제를 정리해 놓았다. 예를 들어, 온톨로지에 추가적인 사항이 발생하여 버전이 새로 생성된 경우에는 전방적용은 가능하지만 후방적용은 허용되지 않는다.

<표 1> 온톨로지 호환성 적용

항목	전방적용	후방적용
요소생성	가능	불가능
요소삭제	불가능	가능
요소수정	불가능	불가능
경미한 수정	가능	가능

<Table 1> Ontology compatibility adapt

<표 1>에서의 가능, 불가능의 기준은 온톨로지 전체를 하나의 단위로 보아 판단하는 것이다. 그러나, 온톨로지와 데이터간의 호환성을 논할 때에도 전체적인 온톨로지보다는 실제적인 세부적인 요소의 호환성을 측정하여 적용하는 것이 보다 정확한 버전 적용이 가능하다. 즉, 클래스나 속성 각각에 대한 세부적인 히스토리를 유지하고 적용할 수 있다면 보다 정교한 버전 적용이 가능하다는 것이다.

더 나아가 가장 최신의 온톨로지에 대하여 질의 처리가 가능하도록 하기 위해서는 온톨로지의 변화 과정에서 미리 최전방 적용이 가능하도록 설계하는 것이 최신의 질의 결과를 보장할 수 있다.

### 3.2 최신 온톨로지 버전 질의 처리

온톨로지의 변화에 따른 버전간의 온톨로지

호환성 문제를 해결하기 위해서는 온톨로지 관리자가 두 버전간의 차이를 인지하고 적용 가능해야 하며, 온톨로지의 언어에 관계없이 버전을 관리할 수 있어야 한다. 결과적으로 기존의 온톨로지 시스템에도 적용 가능하도록 준비하여야 한다. 기존의 연구들에서는 호환성을 고려하여 질의를 처리할 때 최신의 버전을 택하여 처리하도록 하였으나 본 연구에서는 항상 가장 최신의 버전에 대한 질의 결과를 생성하도록 온톨로지를 설계하도록 한다.

온톨로지 기반의 질의 형태는 질의의 중심이 무엇이나에 따라 스키마에 관한 질의, 데이터에 관한 질의와 복합질의의 세 가지 유형으로 분류될 수 있다. 스키마에 관한 질의는 클래스 계층 구조에 관한 질의, 속성 계층 구조에 관한 질의, 속성의 도메인 클래스에 관한 질의, 속성의 레인지 클래스에 관한 질의 등이 이에 해당한다. 데이터에 관한 질의는 실제 데이터를 대상으로 특정 조건에 해당하는 데이터를 요구하는 질의이다. 복합질의는 두 가지 유형이 복합적으로 포함된 질의 유형이다. 이러한 유형에 따라 생성된 질의는 질의가 생성될 당시의 온톨로지 버전에 따라 생성된 것이다. 만약, 온톨로지의 버전이 달라지고 이에 따라 새로운 데이터들이 형성되어 있다면, 예전에 생성된 질의는 전방적용이 가능할 수도 있고 가능하지 않을 수도 있다. 온톨로지의 변화된 구성요소에 따라 질의의 결과에 영향을 미칠 수 있다. 온톨로지의 변화에 따라 질의에 영향을 미치는 구성요소는 다음과 같이 분류할 수 있다. 클래스에 있어서는 클래스의 생성, 삭제, 상하 관계의 변화 및 이동, 재명명, 병합, 분리 등이 있으며, 속성에 대해서는 생성, 삭제, 상하 관계의 변화 및 이동, 재명명, 병합, 분리, 도메인 변화, 레인지 변화 등을 예로 들 수 있다. 이러한 여러 요소를 파악하여 오리지널버전에 적용된 질의를 새로운 버전의 온톨로지에 대하여 적용 가능한지를 판단하는 것이 중요하다. 새로운 버전의 온톨로지에서는 기존의 온톨로지에 비하여 원래 있던 클래스가 삭제되는 경우에 기존의 온톨로지를 대상으로 수행하였던 질의는 수행할 수 없다.

예를 들어, 다음과 같은 RQL 질의가 온톨로지에 대하여 수행되었다고 가정하자.

```
select X
from Artist{X}          (질의 1)
```

질의는 Artist 클래스에 소속되어 있는 인스턴스를 구하는 질의이다. 온톨로지 버전1이 온톨로지 버전2로 변화하였을 경우에, Artist 클래스가 계속 유지되고 있다면, 위의 질의는 새로운 버전에서도 전방적용에 의한 질의 처리가 가능하다. 그러나, 온톨로지가 변화하는 과정에서 Artist 클래스가 Painter와 Sculptor 클래스로 분리되었다면, 새로운 온톨로지 버전에서는 전방적용에 의한 질의 처리가 불가능하다.

두 번째 (질의 2)를 생각해 보자. 버전 1에서 속성 creates의 도메인과 레인지 클래스를 검색하는 RQL 질의가 다음과 같이 수행되었다고 가정하자.

```
select $C1, $C2
from { $C1 } creates { $C2 } (질의 2)
```

질의를 구성하는 구성 요소중에서 변수에 해당하는 클래스 C1, C2는 변화가 없고, creates 속성의 히스토리를 검색하여 최신의 온톨로지를 적용할 수 있다. 그러나, creates 속성이 makes 속성으로 변경되었을 경우에는 전방적용에 의한 최신의 온톨로지 적용은 불가능하다.

위의 예와 같이 온톨로지 데이터의 변화가 일어난 경우에 기존의 질의를 새로운 버전의 온톨로지에 적용이 가능하지 않다면 항상 최신의 풍부한 정보를 검색하고 추론해야 하는 인터넷 검색의 관점에서 엄청난 손실이다.

본 연구에서는 이러한 문제를 해결하기 위하여 온톨로지 데이터의 변화에 따른 버전업을 수행할 때 구버전의 질의가 최신 버전에서도 실행이 가능하도록 즉 최전방 적용이 가능하도록 설계하고 구축하는 방법에 대하여 논한다.

### 3.3 온톨로지 버전 변화 정보

질의1은 별도의 정보가 없다면 새로운 버전에서 실행이 불가능하지만 실제로는 다음과 같이 실행 될 수 있다.

```
select X
from Painter{X}
```

```
+
select Y
from Sculptor{Y}
```

즉, Painter와 Sculptor 클래스 각각에 대하여 질의를 수행하고 이 결과를 합하면 되는 것이다.

질의2에 있어서도 동일한 클래스에 대하여 creates를 대체 할 수 있는 유사한 개념의 속성이 존재하도록 설계를 한다면 다음과 같이 질의를 수행할 수 있다.

```
select $C1, $C2
from { $C1 } makes { $C2 }
```

이와 같이, 클래스와 속성에 대한 변화내용을 분석하여 버전 변화에 따른 별도의 적용이 가능한 구조로 표현한다면, 이전 버전에서 실행되는 질의가 최전방적용 가능한 질의로 변형 할 수 있다. 질의가 최전방적용이 가능해지면 새로운 온톨로지에 연결되어 있는 데이터들을 검색할 수 있으며, 따라서 좀 더 많은 내용에 대한 검색이 가능해진다. 나아가, 여러 사이트에 걸쳐 다른 버전의 온톨로지를 바탕으로 데이터시스템이 구축되어 있다고 가정한다면, 그 효과는 더욱 커진다고 예상할 수 있다.

온톨로지 버전의 변화를 감지하고 표현하는데 있어서 가장 중요한 요소는, 오리지널 온톨로지와 새로운 온톨로지 두 개 간의 변화를 어떻게 표현하고 저장하느냐의 문제이다. 전체적인 온톨로지 관점이 아니라, 실제로 질의가 수행되는 단위인 클래스와 속성의 변화 히스토리를 각각 표현하고 저장하는 방법이 요구된다. 가장 중요한 것은 이 모든 것이 가장 최신의 온톨로지에 대하여 기존 질의가 수행 가능하도록 구성되어야 한다. 본 연구에서는 이를 구체적으로 명시해주는 그래프 구조를 제안 한다. 그래프에는 클래스와 속성의 변화 내용이 그대로 표현되며, 나아가 변화된 사실에 대하여 질의 처리시에 필요한 내용들을 명시하여 최신의 온톨로지에 대하여 실행이 가능하도록 한다.

구체적으로 언급하면, 클래스가 삭제된 경우에 이를 대체할 수 있는 클래스를 표현해 준다. 클래스가 재명명된 경우에는 기존의 클래스가 새로운 이름의 클래스에 대하여 실행되도록 지시

해 줄 수 있다.

표는 클래스와 속성의 변화를 그래프 형태로 각각 표현한 것이다. 버전이 변하면서 온톨로지의 클래스와 속성중에 변화가 생기고, 이를 그래프 형태로 나타낸 것이며, 이를 보면 가장 최신의 온톨로지 버전에 대한 대체 가능 상태를 알 수 있다.

<표 2> 온톨로지변화항목

항목	요소	그래프
클래스 변화	생성	
	무변화	
	삭제	
	재명명	
	병합	
	분리	
속성 변화	생성	
	무변화	
	삭제	
	재명명	
	병합	
	분리	

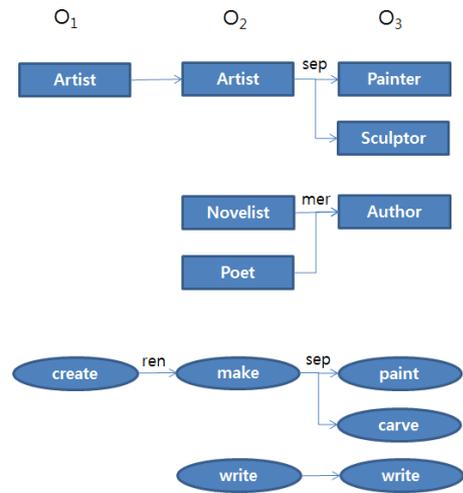
<Table 2> Ontology change items

클래스 변화를 표현하기 위하여 클래스를 사각형으로 표현하고 클래스에 변화된 내용이 있는 경우 변화의 종류를 간선에 표시하도록 하였다. 클래스 변화 내용은 삭제, 재명명, 병합, 분리로 분류하였다. 질의변형 방법은 그래프에서

클래스가 삭제된 경우에는 그래프에 명시된 대체클래스에 대하여 질의를 수행하도록 질의를 변형한다. 재명명의 경우에는 새로운 이름의 클래스에 대하여 질의를 변형하도록 진행된다. 병합의 경우에는 병합된 클래스에 대하여 질의를 수행하며, 클래스가 분리된 경우에는 분리된 두 개의 클래스에 대하여 각각 질의를 수행하고 질의 결과를 병합한다.

속성의 변화를 표현하기 위해서 속성을 타원으로 표현하였다. 속성에 변화된 내용이 있는 경우에 변화의 종류를 간선에 표시하도록 하였다. 변화의 내용은 삭제, 재명명, 병합, 분리로 분류하였다. 질의변형 방법은 그래프에서 속성이 삭제된 경우에는 그래프에 명시된 대체속성에 대하여 질의를 수행하도록 질의를 변형한다. 재명명의 경우에는 새로운 이름의 속성에 대하여 질의를 변형하도록 진행된다. 병합의 경우에는 병합된 속성에 대하여 질의를 수행하며, 속성이 분리된 경우에는 분리된 두 개의 속성에 대하여 각각 질의를 수행하고 질의 결과를 병합한다.

(그림 4) 온톨로지 항목 변화 예



(Figure 4) Ontology item change example

(그림 4)는 클래스와 속성의 변화 요소를 가지고 온톨로지의 데이터가 변화될 때 생성된 그래프의 예이다. O<sub>1</sub>, O<sub>2</sub>, O<sub>3</sub>는 각각 온톨로지의 변화된 예를 보여준다. Artist 클래스는 O<sub>2</sub>에서는 그대로 변화 없이 유지되지만, O<sub>3</sub>에서는

Painter 클래스와 Sculptor 클래스로 나누어진다. 한편, Novelist 클래스와 Poet 클래스는  $O_3$ 에서 Author 클래스로 병합된다. 따라서, 세 개의 버전을 거치면서 Artist 클래스와 연관된 모든 질의는 Painter 클래스와 Sculptor 클래스로 대체됨을 알 수 있다. Novelist 클래스와 Poet 클래스는 Author 클래스를 사용하여 질의를 처리하도록 한다. create 속성에 대하여는 make 속성을 거쳐 paint와 carve로 나누어 졌으므로, 질의에 대하여도 두 가지를 고려하여 처리하도록 하면 된다.

이와 같은 변화를 감지하고 최종적으로 적용할 클래스와 속성을 결정해야 하는데 이를 위한 알고리즘이 다음과 같다.

```

Algorithm QueryTransformation
입력 : Query q, OntologyChangeGraph G
결과 : Modified Query q'
for all element ei of q (Classes and Properties)
    search ei from G
        search graph of ei element
        node=last node of ei
        edge=last edge of ei
        case(edge) {
            del : revise q according to
                    alternative node name
            ren : revise q according to
                    renamed node name
            mer : revise q according to
                    merged node name
            sep : revise q according to
                    two separated node name
            else : no change
        }
    q'=q
return q'
endfor
    
```

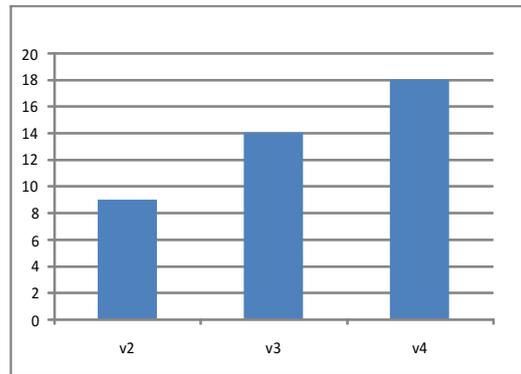
온톨로지변화그래프(OntologyChangeGraph)는 온톨로지의 버전이 생성될 때마다 업데이트되며, 온톨로지의 모든 요소에 대하여 변화된 내용을 기록하게 된다. 클래스와 속성의 변화에 대하여 그래프 형태로 저장하고 검색이 가능하도록 하며, 각 요소의 변화중 기존의 질의가 수행이 불가능하도록 변화하는 경우에는 이를 해결하여 질의가 대체 요소에 의하여 수행 가능하도록 그

래프를 생성한다. 이에 대한 결정은 설계자가 결정하도록 한다.

### 4. 실험

본 논문에서 제안한 온톨로지의 데이터변화에 따른 질의 처리 알고리즘을 시뮬레이션 형태로 검증하기 위하여 온톨로지 스키마는 MovieOntology[11]를 참조하여 영화와 관련된 개념의 클래스 76개와 속성 42개를 정의하고 각 클래스간의 의미적인 관계를 고려하여 계층관계를 정립하였다. 정의된 온톨로지와 연결된 트리플 형태의 KB를 구성하고 이와 관계가 있는 영화와 관련된 문서 500개 이상의 문서 페이지를 수집하여 연결하였다. 온톨로지 스키마를 임의로 3번의 변화를 주어 4개의 버전을 생성하였으며, 오리지널 버전을 기준으로 30개의 질의를 생성하여 버전이 변할 때마다 질의의 변화를 측정하였다. 알고리즘은 C언어로 구현되었고, 윈도우 7 운영체제가 설치된 PC에서 비주얼스튜디오 환경으로 처리하였다.

(그림 5) 요소변화에 영향을 받은 질의의 개수



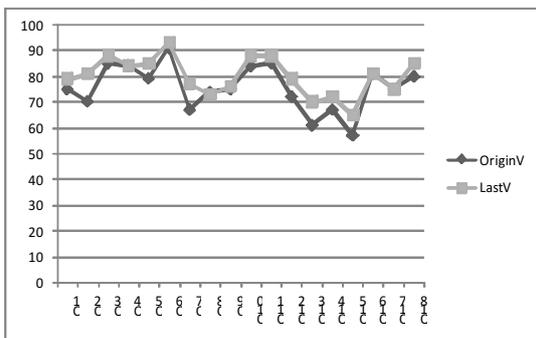
(Figure 5) Number of queries affected by change

오리지널 버전  $V_1$ 에 대하여 버전이 변할 때마다 영향을 받는 질의의 개수는 (그림 5)와 같이 설정 되었다. 개수가 점점 증가하는 추세로 설정되었으며, 이는 온톨로지가 변화되는 내용이 계속적으로 추가되기 때문이다. 반면에 영향을 받지 않는 질의도 존재하도록 하였다.

각 버전이 변화되는 내용을 각 클래스와 속성

에 대하여 온톨로지 변화항목의 세부내용을 기준으로 변화그래프를 설정하고, 질의변형알고리즘에 따라 변형된 질의를 제시하도록 프로그래밍하였다. 오리지널 버전에 대한 질의는 온톨로지 변화그래프를 참조하여 항상 마지막 버전에 대하여 실행되도록 변형하여 결과를 측정하였다. 실험은 최초 버전에 대한 질의 결과의 재현율(Recall)과 마지막 버전에 대하여 변형된 질의를 수행한 결과의 재현율을 비교하는 형식으로 진행되었다. 재현율은 적합 문서 집합을 현재 결과와 비교하여 누락이 없는 정도를 퍼센트로 표시하도록 하였다. 온톨로지의 버전이 바뀌에 따라 연결되는 문서를 변형 혹은 추가하는 방식으로 진행하였으며, 온톨로지의 변경된 내용이 없는 요소에 있어서도 오리지널 버전과 마지막 버전에 연결된 사실과 문서의 내용이 상이하도록 설계하였다.

(그림 6) 질의 재현율 비교



(Figure 6) Query recall results

(그림 6)은 변형된 질의에 대한 재현율을 표시한 것이며, 제시된 재현율을 분석하면, 대체적으로 오리지널 버전에 대한 질의 결과보다 5% 이상의 재현율 상승을 보이는 것으로 조사되었다. 이는 온톨로지 버전의 변화가 발생하는 경우에 가장 최근의 버전에 대한 질의 처리가 가능하도록 방법을 제시하는 것이 질의 결과 만족도를 높이는 것을 의미한다. 결과 그래프의 상승률이 알고리즘의 효율성과 정확성에 대한 증명을 위하여 좀 더 많은 데이터에 대한 검증을 통하여 결과를 검토해볼 필요가 있다.

## 5. 결론

인터넷상의 데이터의 양이 기하급수적으로 증가함에 따라 온톨로지를 이용한 정보 검색의 중요성이 증대되고 있다. 온톨로지를 이용한 검색은 단순한 키워드 검색의 범위를 벗어나 시맨틱 정보를 이용한 검색이 가능하도록 한다. 온톨로지에 대한 연구가 활발히 진행되고 있지만 온톨로지의 변화에 대응하기 위한 연구는 미비한 상태이다. 본 연구에서는 항상 변화할 가능성을 지니고 있는 온톨로지를 변화 내용에 따라 각 버전별로 분류하고 이를 유지하기 위한 질의처리 기법을 제안하였다. 온톨로지의 요소별 변화 항목들을 열거하였으며, 이러한 변화들을 일목요연하게 표현하고 적용하기 위한 변화그래프를 제공하였다. 변화그래프는 이전 버전의 온톨로지를 기반으로 작성된 질의가 최신의 온톨로지를 기반으로 처리되기 위하여 필요한 정보를 제공하여 사용자가 가장 최신의 풍부한 정보를 검색 가능하도록 하는 것을 목표로 하였다. 샘플 온톨로지에 대한 질의 설정을 통하여 제안된 방법의 성능의 효용성을 보였다. 이 연구는 하나의 온톨로지의 변화에 대한 것에 한정되었으며 이질적인 온톨로지 간의 질의 처리 문제해결 또한 중요한 연구 대상이다.

## References

- [1] Berners-Lee, Tim, Helder, J, Lassila, O, "The Semantic Web," Scientific American, pp.28-37, May 2001.
- [2] Klein, M., Fensel, D., "Ontology versioning on the Semantic Web.," Proc. Int. Semantic Web Working Symposium, pp.75-91, 2001.
- [3] Gruber, T.R., "A Translation Approach to Portable Ontology Specifications," Knowl. Acquis., Vol. 5, No. 2., pp.199-220, 1993.
- [4] Resource Description Framework(RDF) Schema Specification 1.0, <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>
- [5] S.K. Oh, B.G. Kim, "Semantic search of web documents using ontology," Journal of Digital Contents Society. v.15, no.5, pp.603-612, Oct. 2014.

[6] <http://www.nongnu.org/cvs/#documentation>

[7] <http://subversion.tigris.org/>

[8] P. Plessers, O. De Troyer. Ontology Change Detection Using a Version Log. In Proceedings of the 4th International Semantic Web Conference (ISWC-05), pp.578-592, 2005.

[9] W3C. SPARQL Query Language for RDF, 2008. W3C Recommendation (15 January 2008). Available at <http://www.w3.org/TR/rdf-sparql-query>

[10] G. Kasneci, F. M. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum. NAGA: Searching and ranking knowledge. In Proc. ICDE-2008, pp.953 - 962. 2008.

[11] <http://www.movieontology.org>

**오 성 균**



1981년 : 홍익대학교 이공대학  
전자계산학과 이학사  
1984년 : 연세대학교 산업대학원  
전자계산학과 공학석사  
1999년 : 홍익대학교 공과대학  
전자계산학과 이학박사  
1987년~현재 : 서일대학교 컴퓨터소프트웨어과 교수  
관심분야 : 능동데이터베이스, XML모델링, 소프트웨어공학

**김 병 곤**



1990년 : 홍익대학교 공과대학  
전자계산학과 이학사  
1992년 : 홍익대학교 공과대학  
전자계산학과 이학석사  
2001년 : 홍익대학교 공과대학  
전자계산학과 이학박사  
1992년~1998년 : 국방과학연구소 연구원  
2001년~현재 : 부천대학교 e-비즈니스과 부교수  
관심분야 : 다차원 데이터 인덱싱, 시맨틱웹, 온톨로지